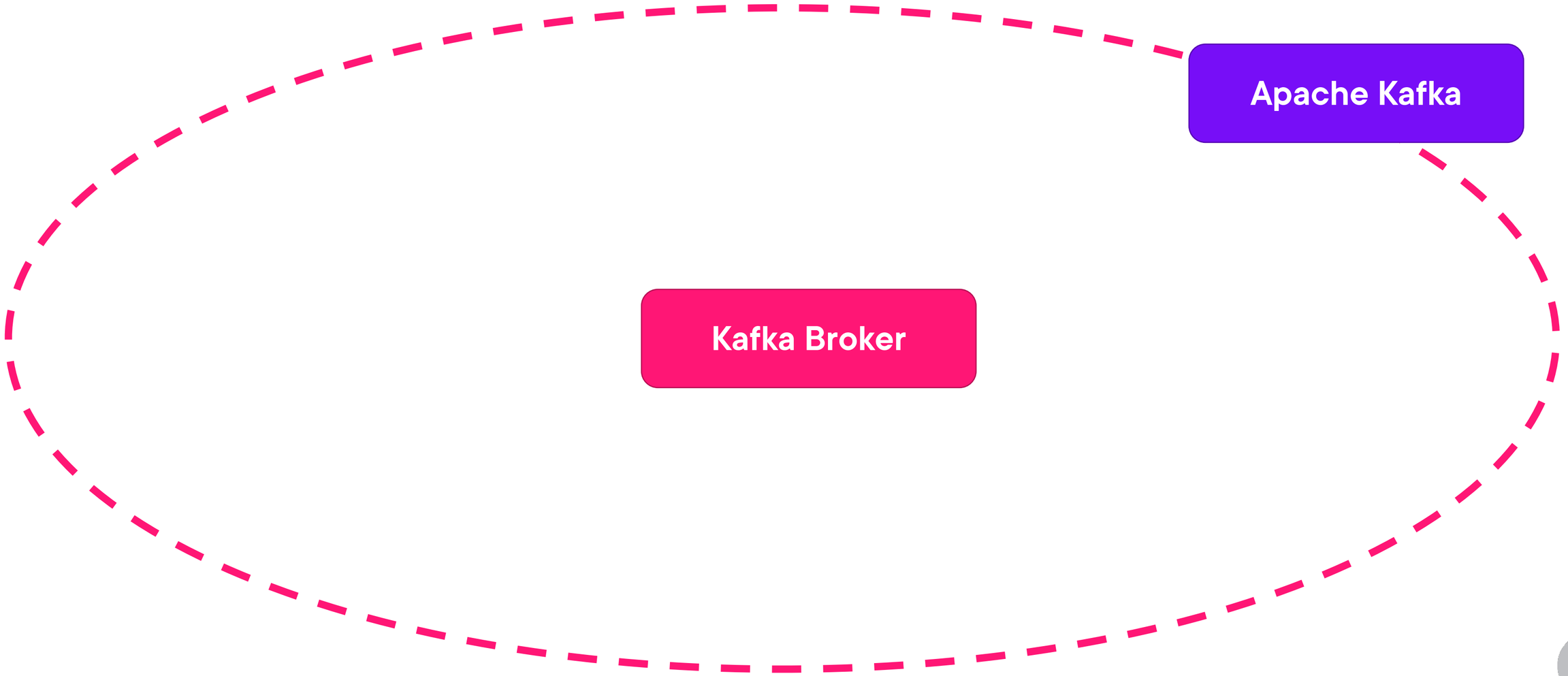
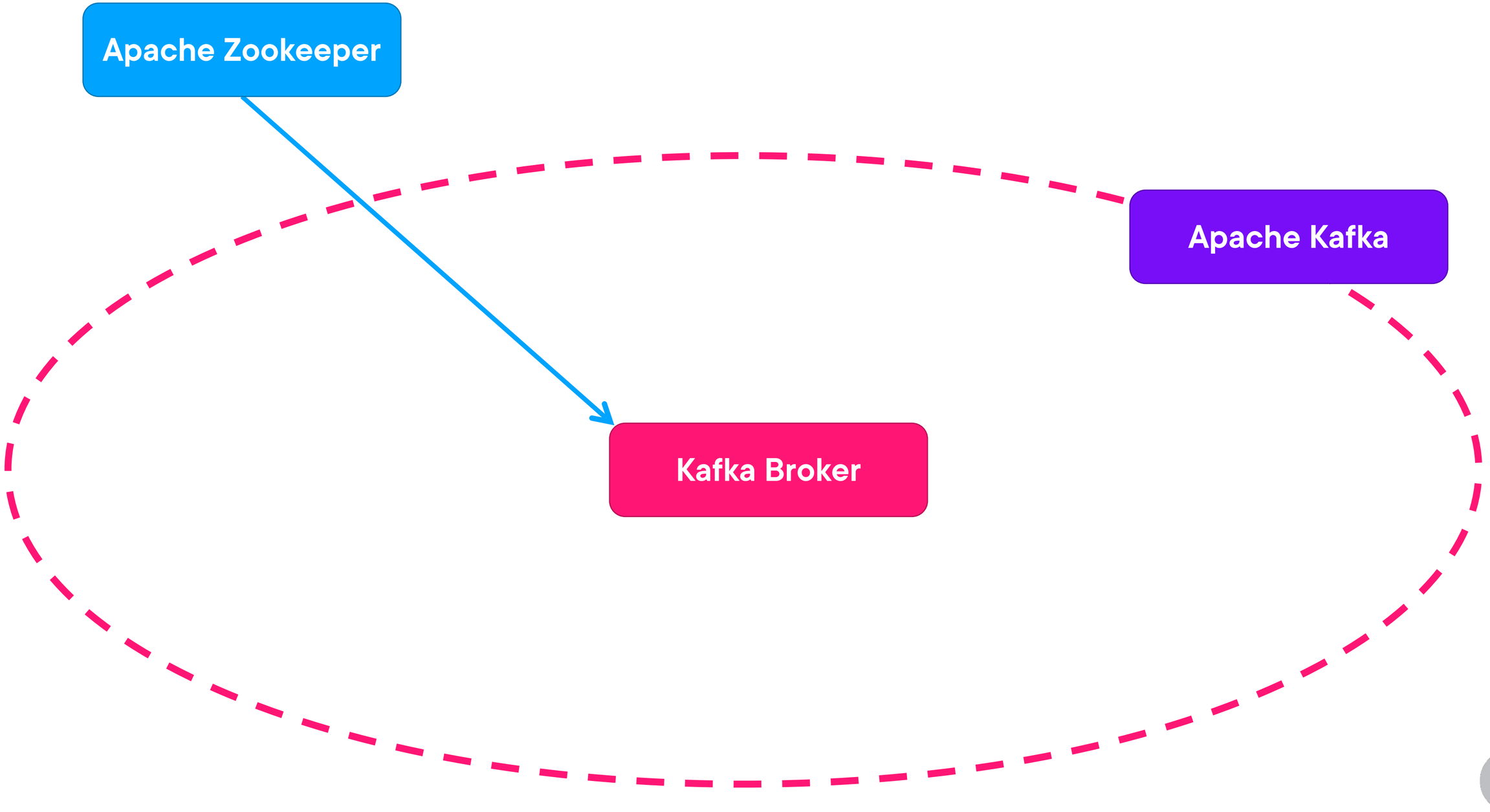
Meet Kafka

Axel Sirota

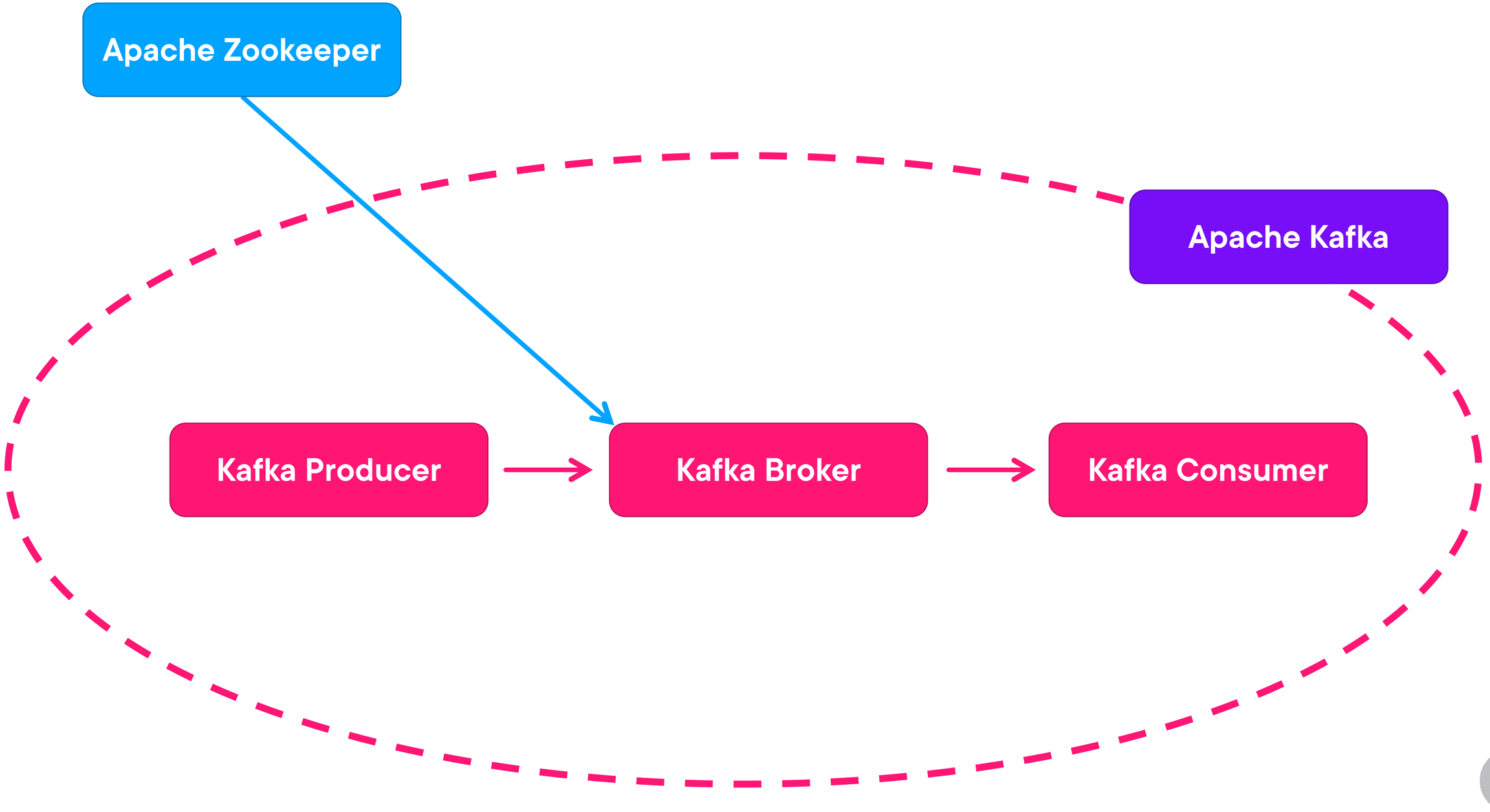
AI and Cloud Consultant

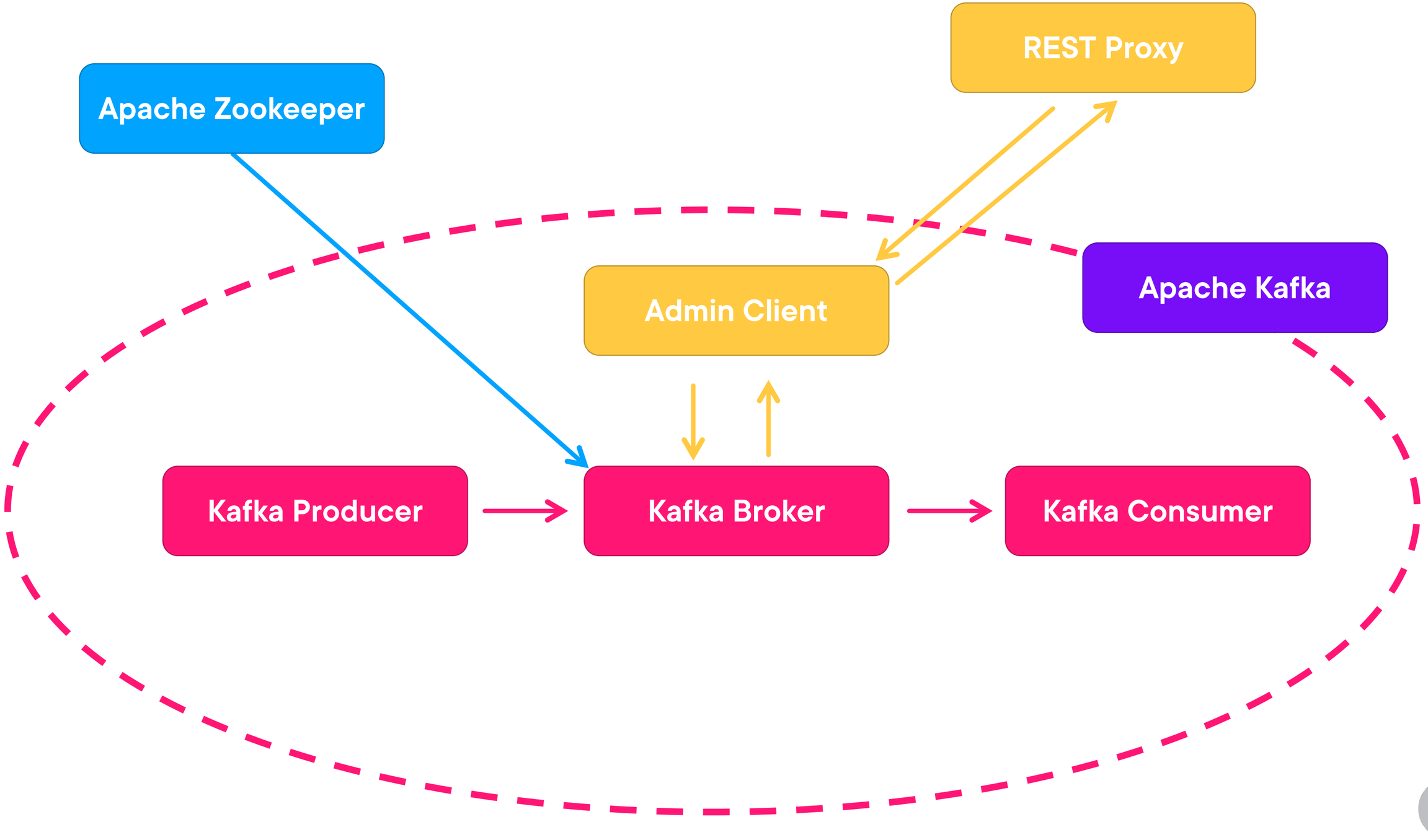
@AxelSirota

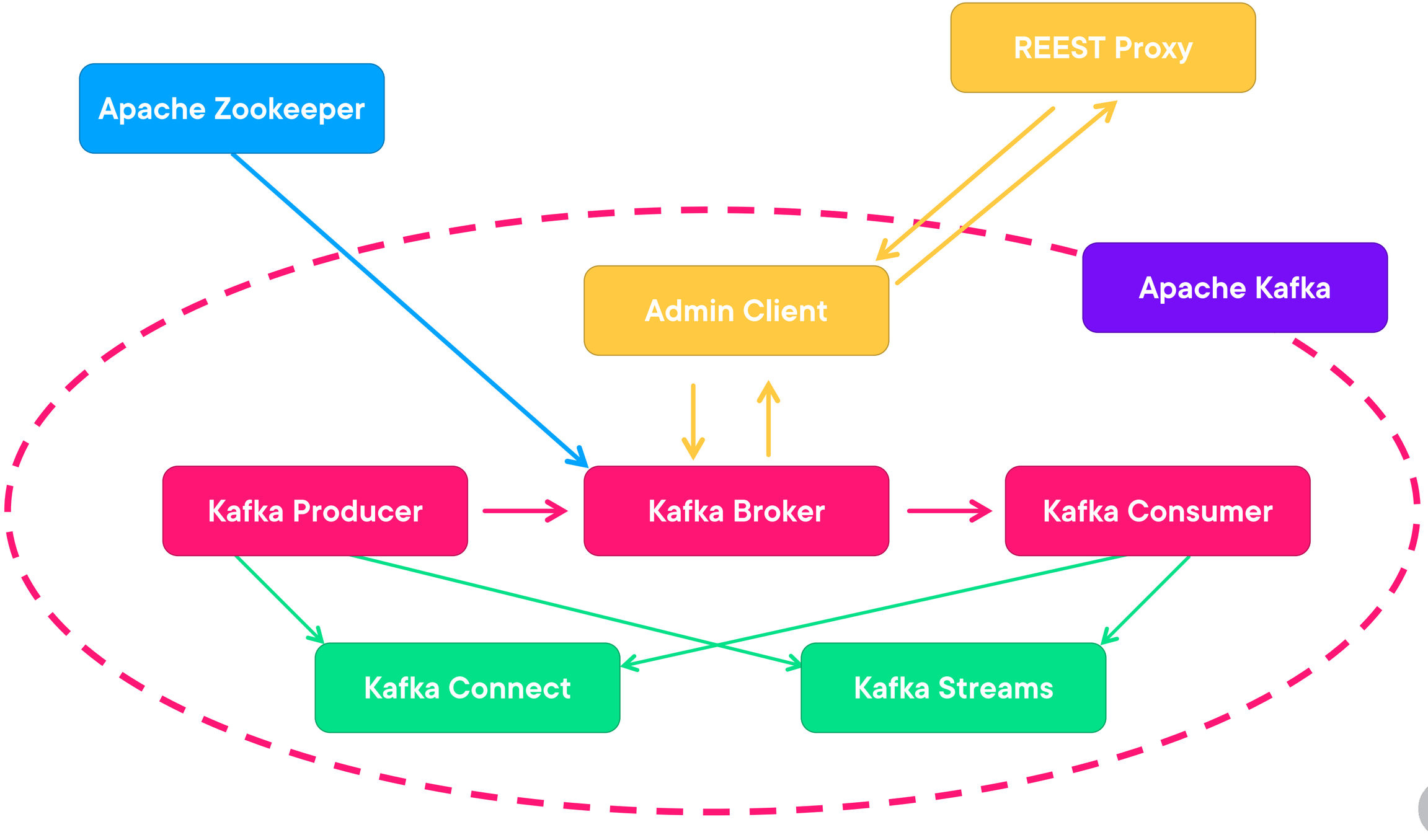

Apache Kafka
Kafka Broker

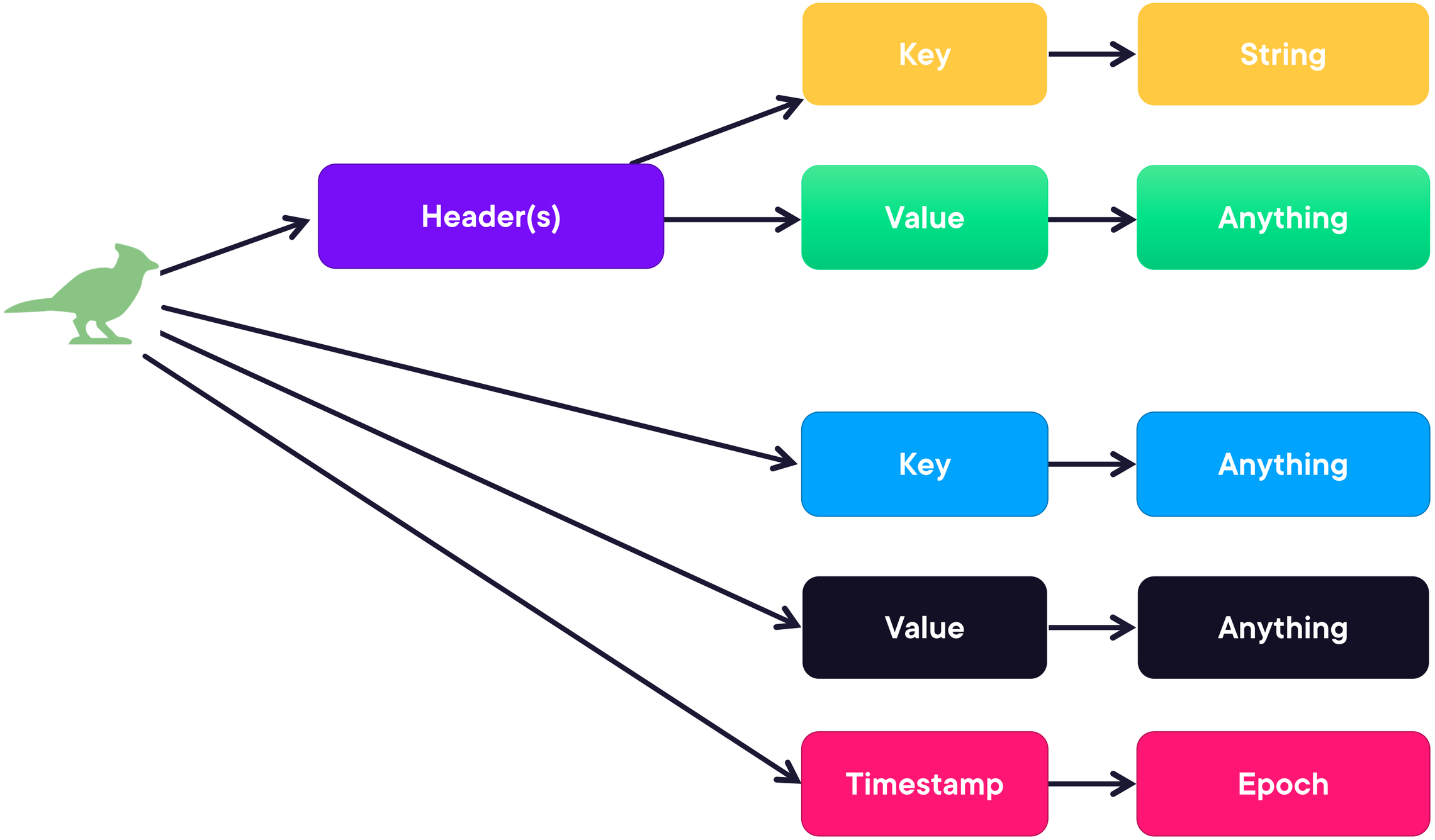

Apache Zookeeper
Apache Kafka
Kafka Broker

- Service Discovery
- Service Address Retrieval
- Management of Elections


Apache Zookeeper
Apache Kafka
Kafka Producer
Kafka Broker
Kafka Consumer


REST Proxy
Apache Zookeeper
Admin Client
Apache Kafka
Kafka Producer
Kafka Broker
Kafka Consumer


REEST Proxy
Apache Zookeeper
Admin Client
Apache Kafka
Kafka Producer
Kafka Broker
Kafka Consumer
Kafka Connect
Kafka Streams


Key
String
Header(s)
Value
Anything
Key
Anything
Value
Anything
Timestamp
Epoch

Deploy Kafka

Kafka Ecosystem

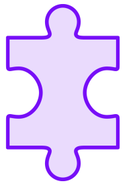
Packaging Versions:

Confluent Cloud

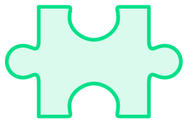
Cloudera

Strimzi offers the Helm charts to deploy Kafka to Kubernetes

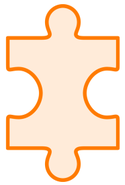
In the Logging Area, for Example, We Have Plugin For:



syslog



fluentd



Logstash

Filebeat

DBs

JDBC connector to cover RDBMS Dos

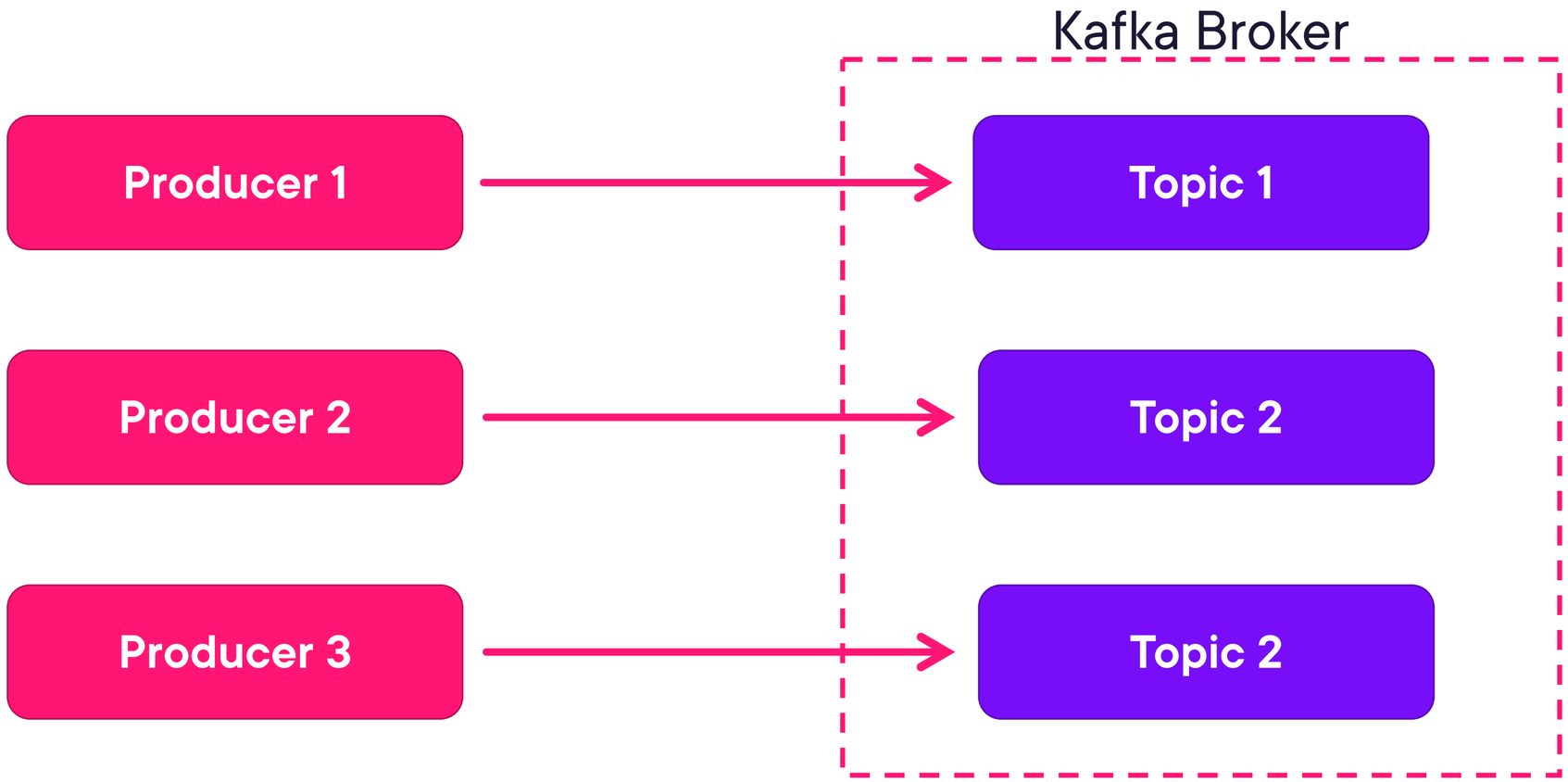
Debezium also offers a connector that include no-SQL Dos

Presto connector that allows us to query Kafka using Presto queries

Interaction with Kafka via the CLI

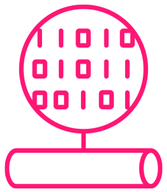
Topics, Partitions, and Rebalance

Topics

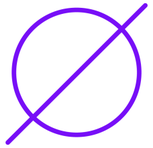

Kafka Broker
Producer 1
Producer 2
Producer 3
Topic 1
Topic 2
Topic 2

Retention Can Be Set To:

Days: Meaning only keeps messages of up to N days and then delete

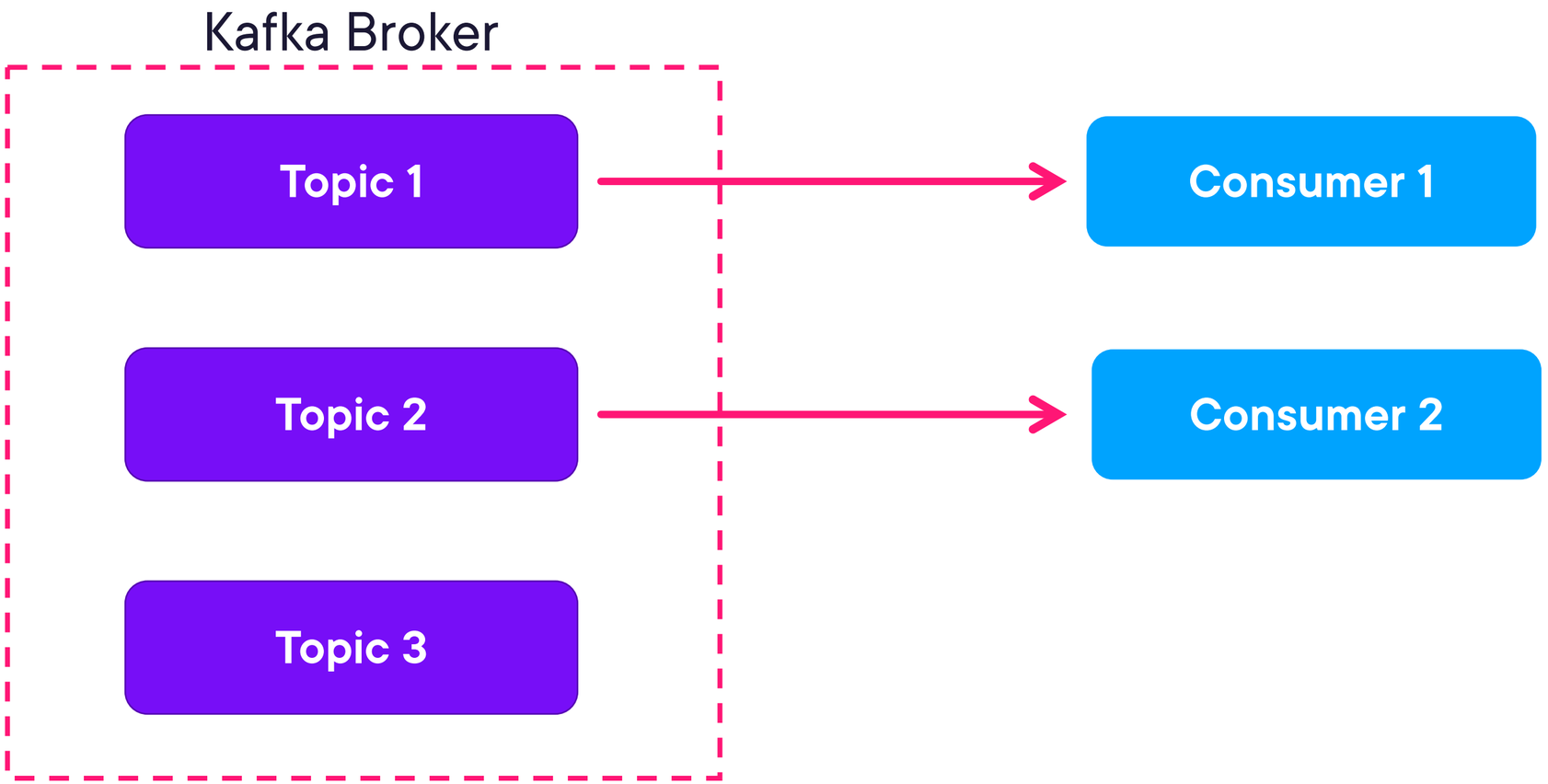


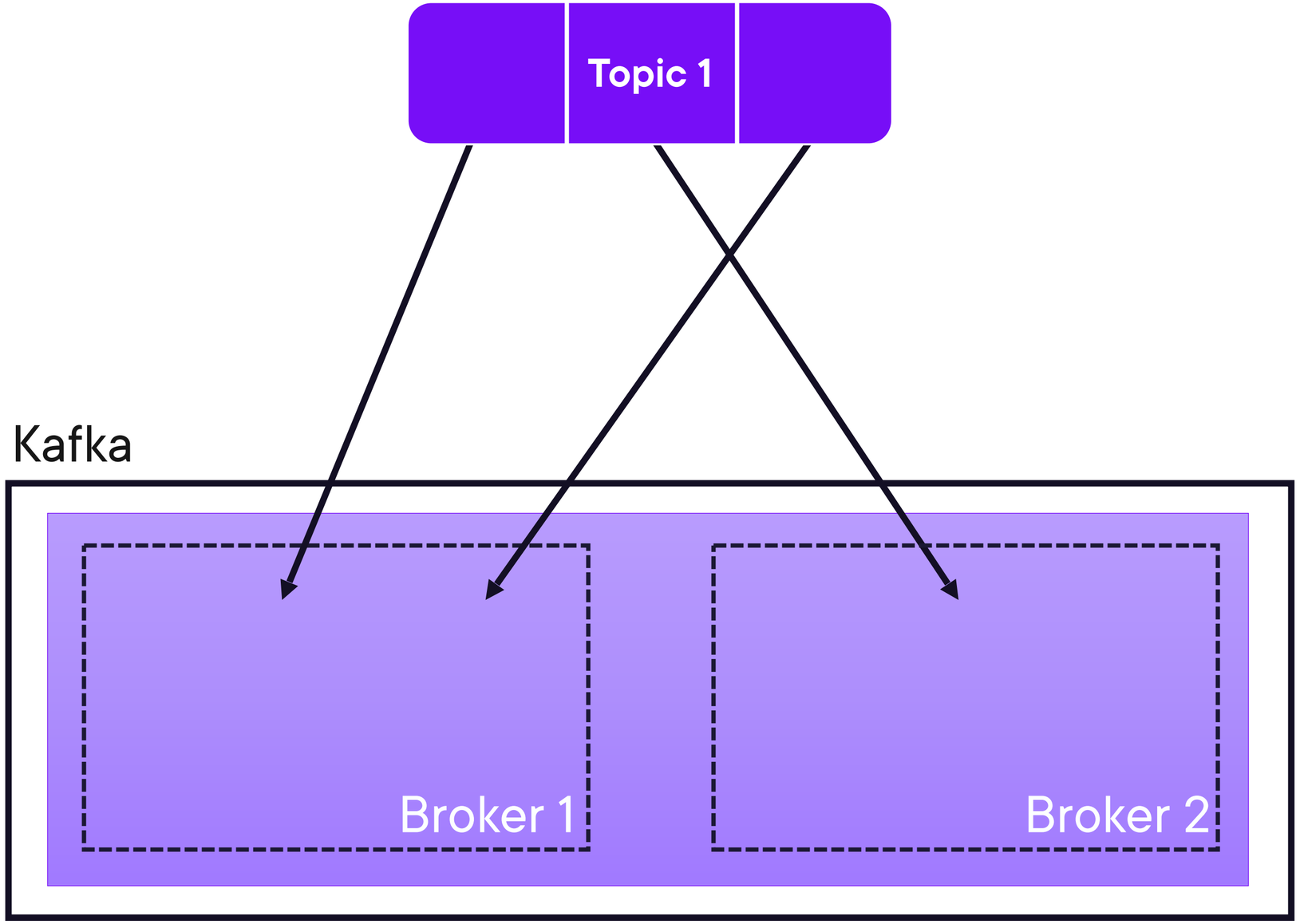
Bytes: Meaning only keep messages until the topic has more than X bytes and then delete



None: Meaning store all messages from the beginning of time

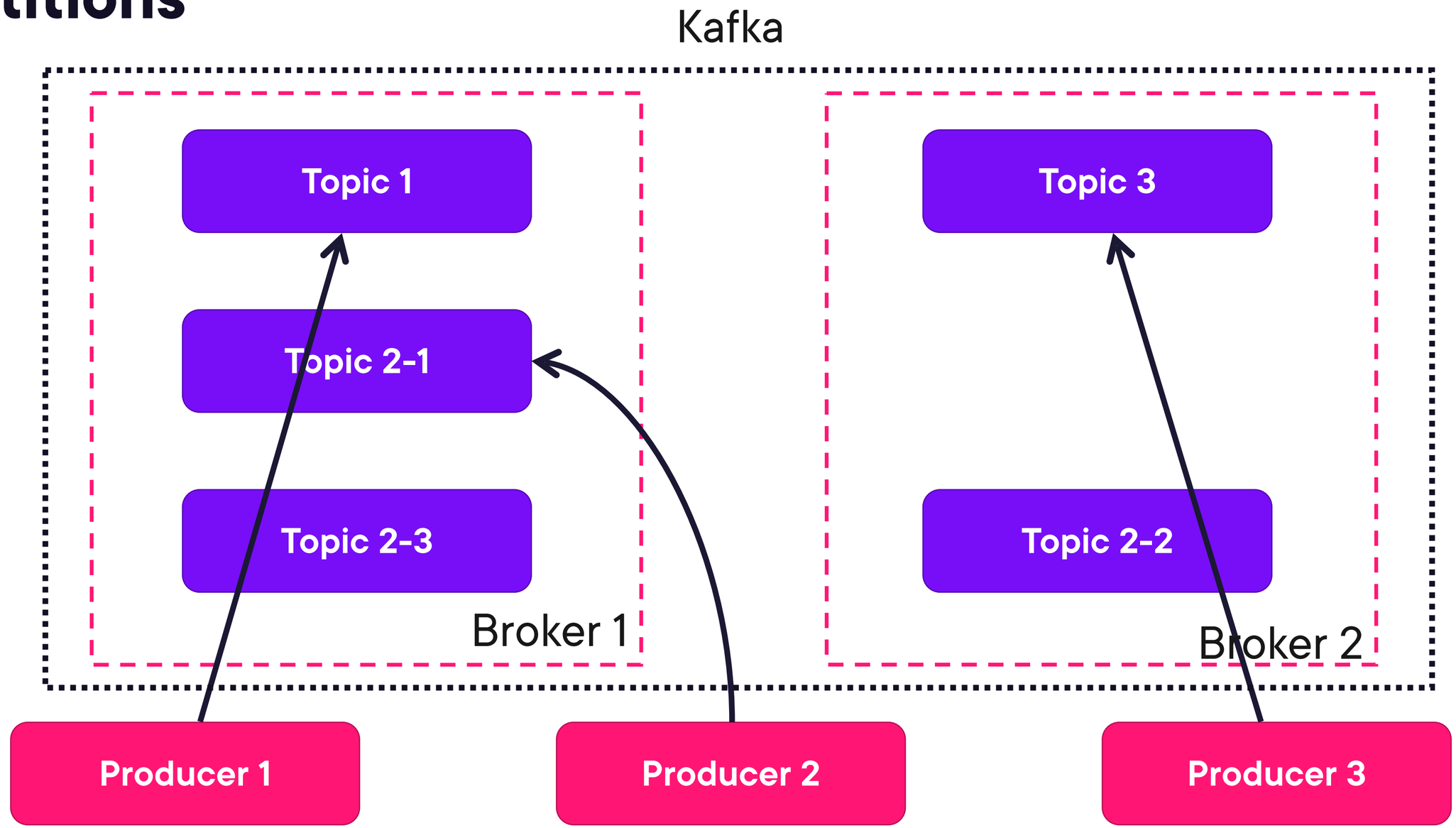
Topics



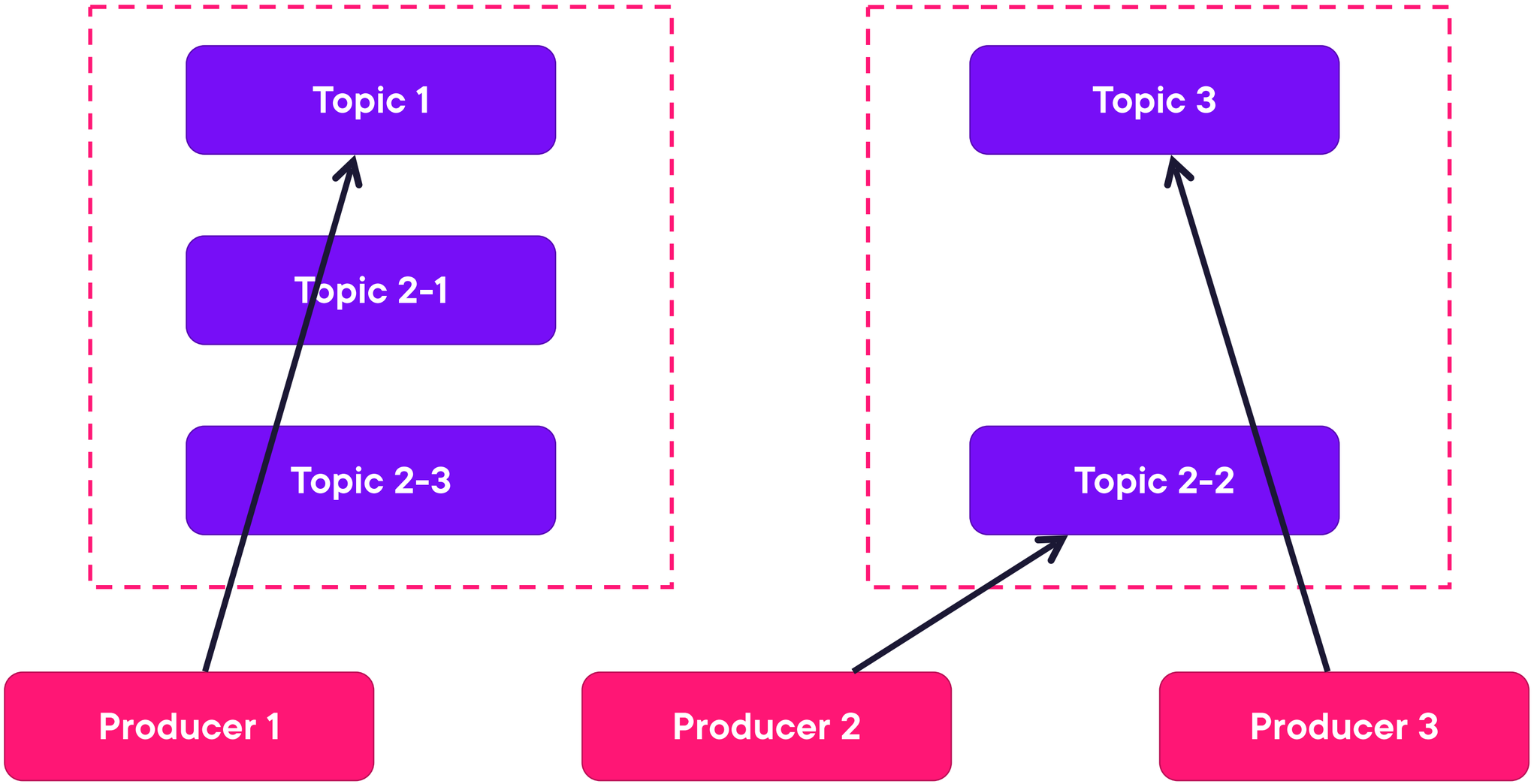

Topic 1
Kafka
Broker 1
Broker 2

The partitioner uses the key to distribute the messages to the correct partition

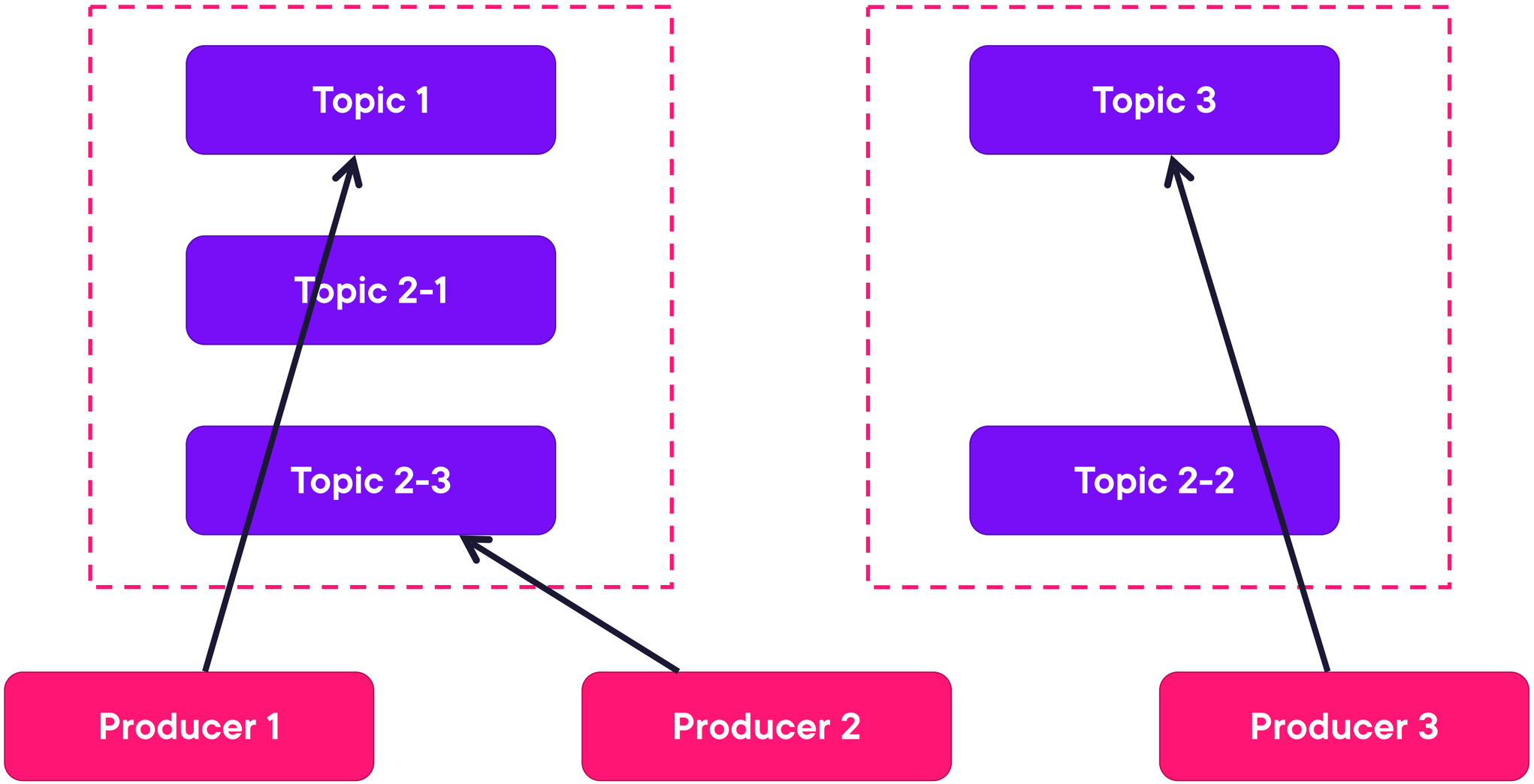
Partitions


Kafka
Topic 1
Topic 2-1
Topic 2-3
Broker 1
Topic 3
Topic 2-2
Broker 2
Producer 1
Producer 2
Producer 3

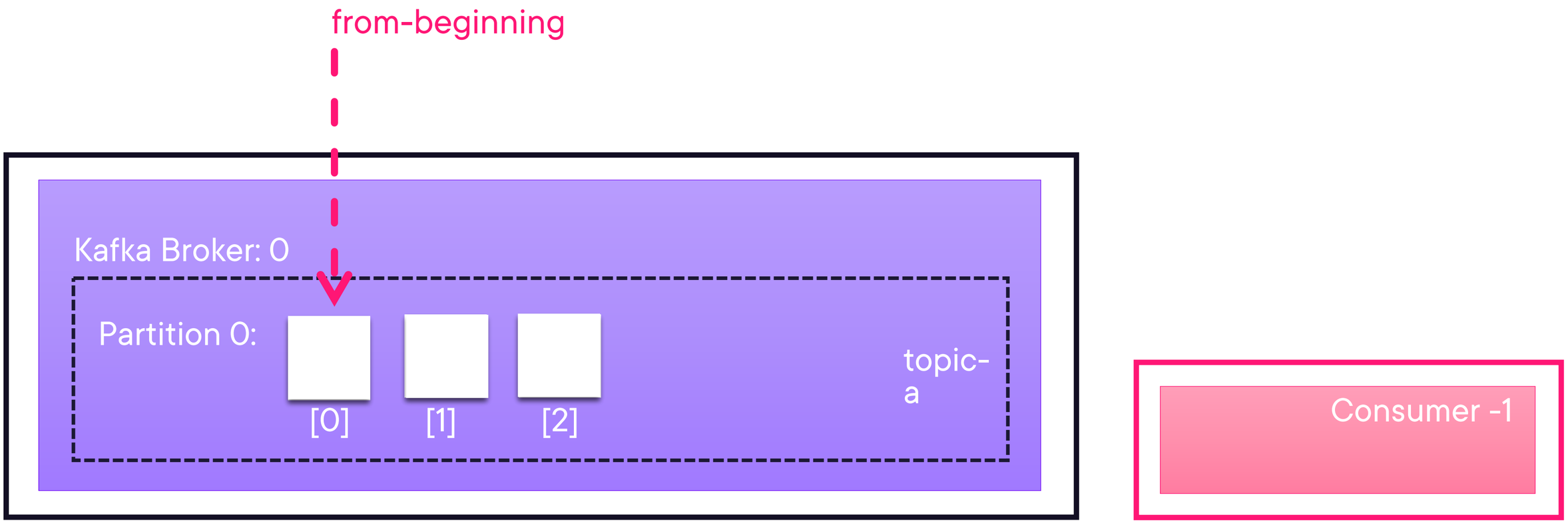
Partitions


Topic 1
Topic 2-1
Topic 2-3
Topic 3
Topic 2-2
Producer 1
Producer 2
Producer 3

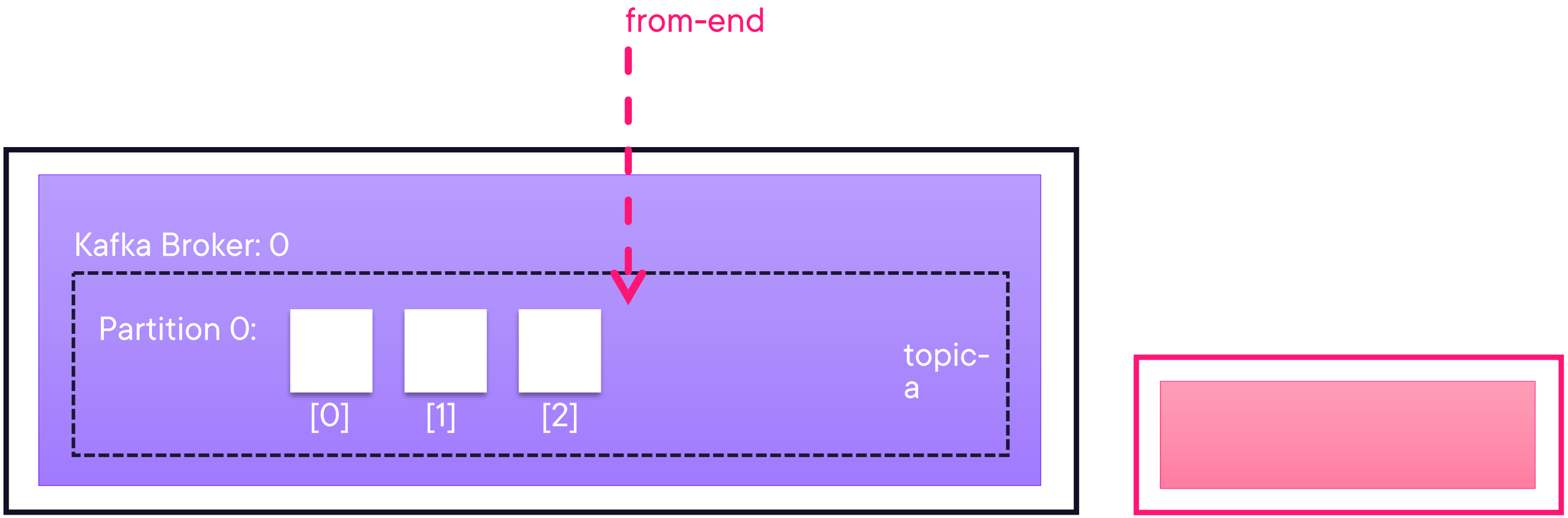
Partitions


Topic 1
Topic 2-1
Topic 2-3
Topic 3
Topic 2-2
Producer 1
Producer 2
Producer 3

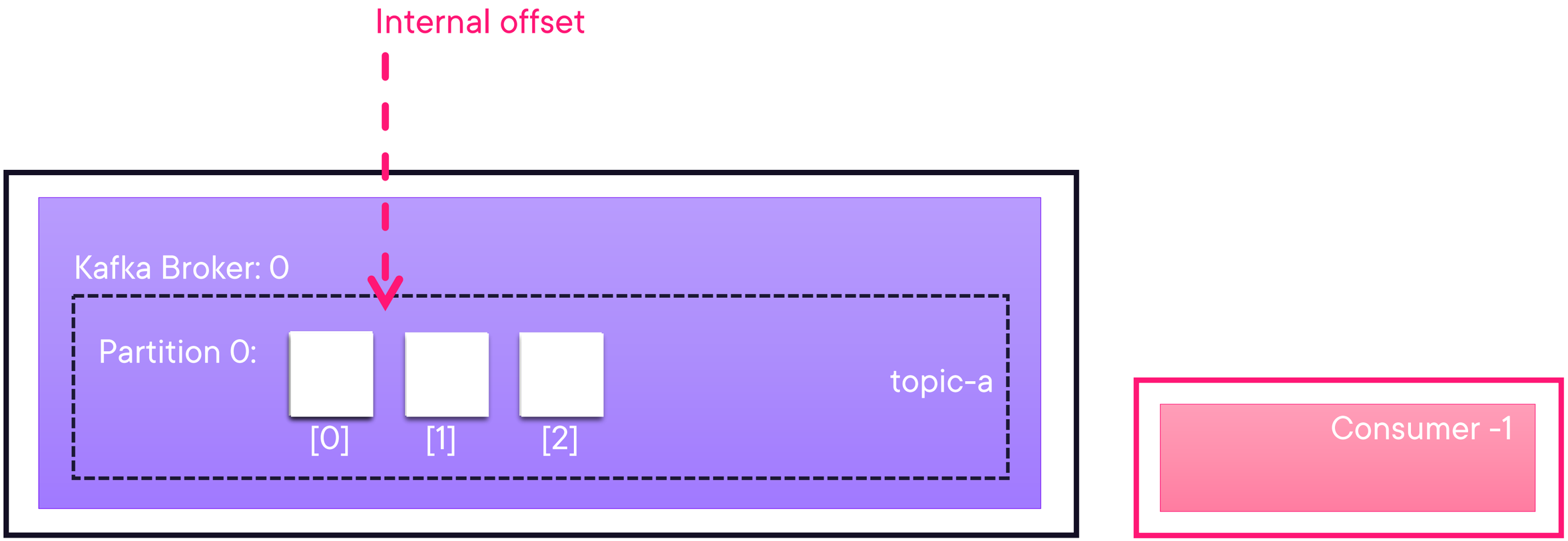
Consumer Offset: From Beginning



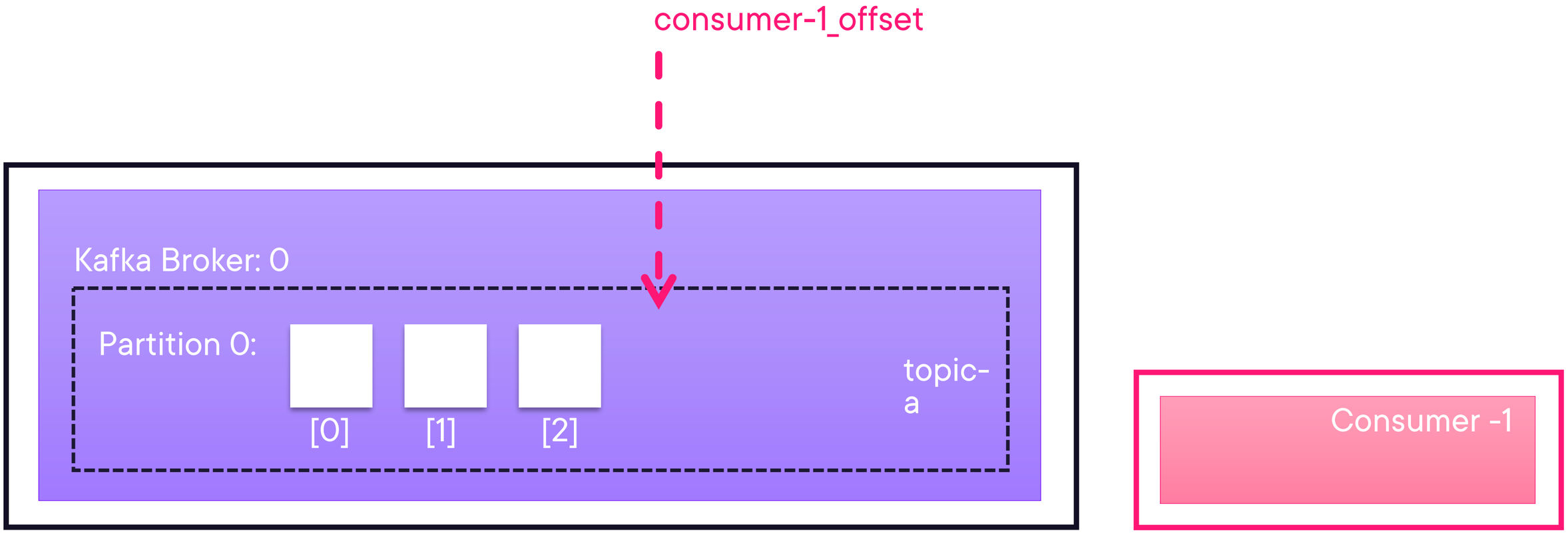
Consumer Offset: From End



Consumer Offset: Already Have Been Here

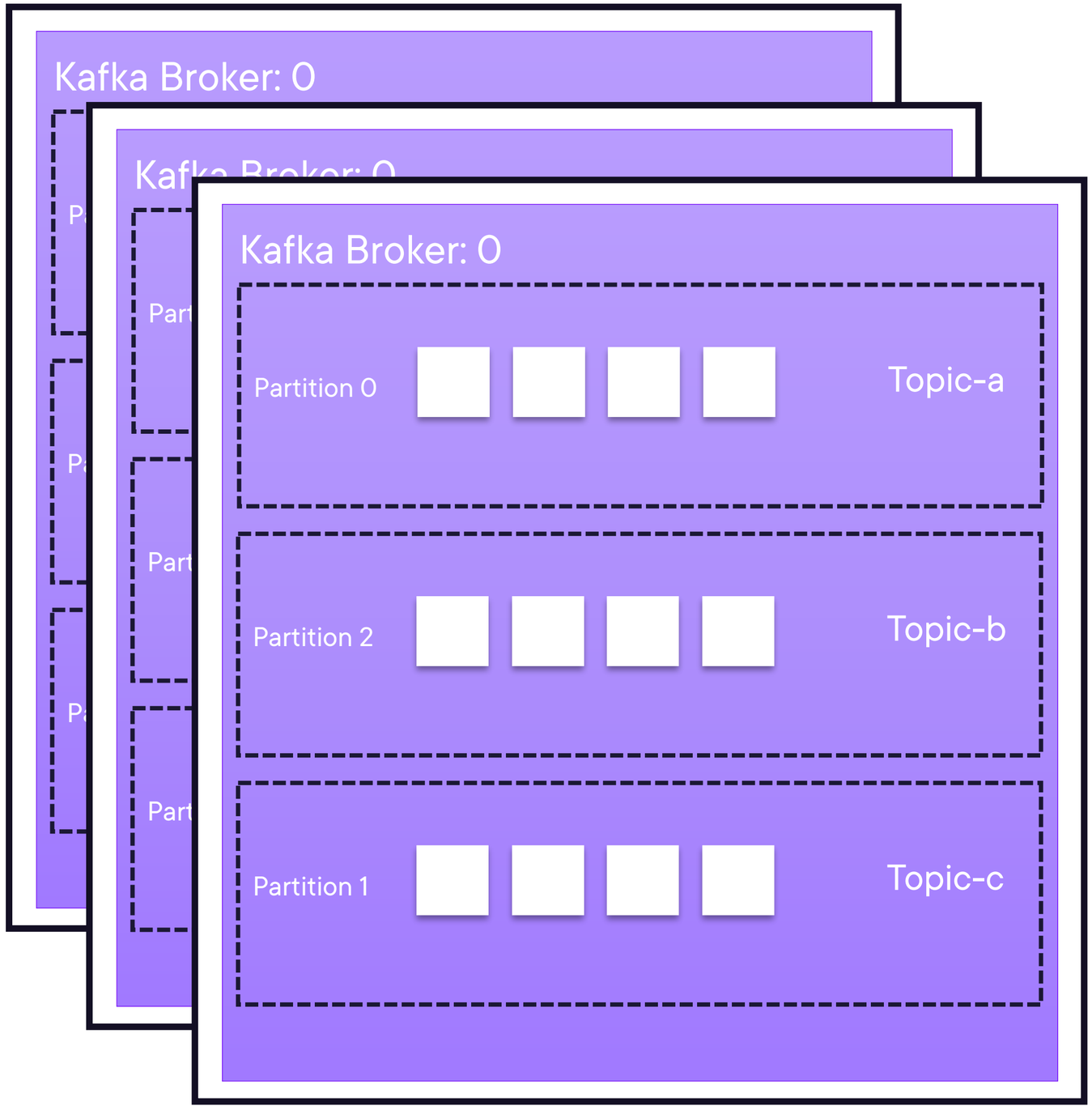


Consumer Offset



We Learned

- Topics can be partitioned
- The partitioner uses the message key to redirect messages to partitions
- Consumers read from partitions and have an offset
- Therefore, more partitions enable more consumers, therefore more scalability



Each partition is on a different broker, therefore a single topic is scaled

Interacting with Kafka via REST Proxy