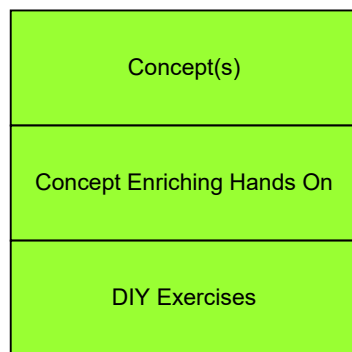
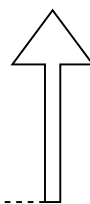
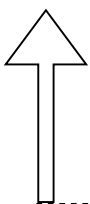
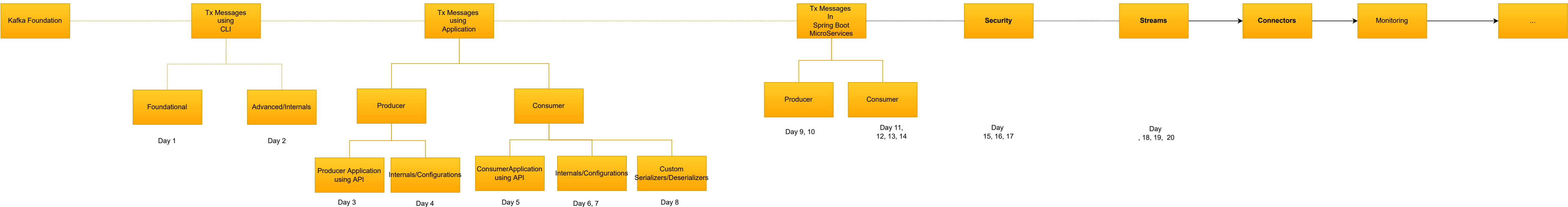
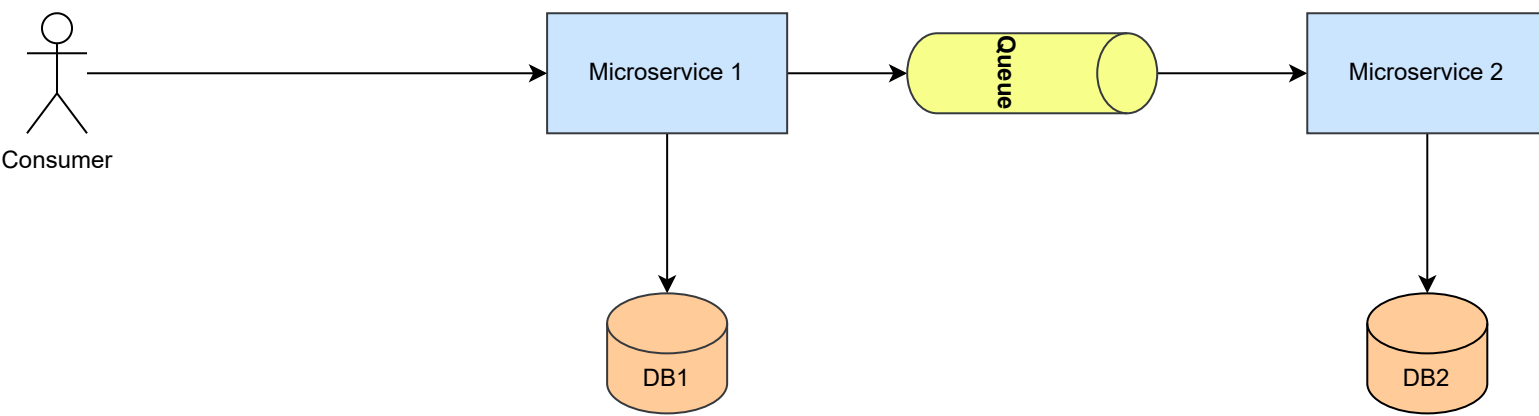
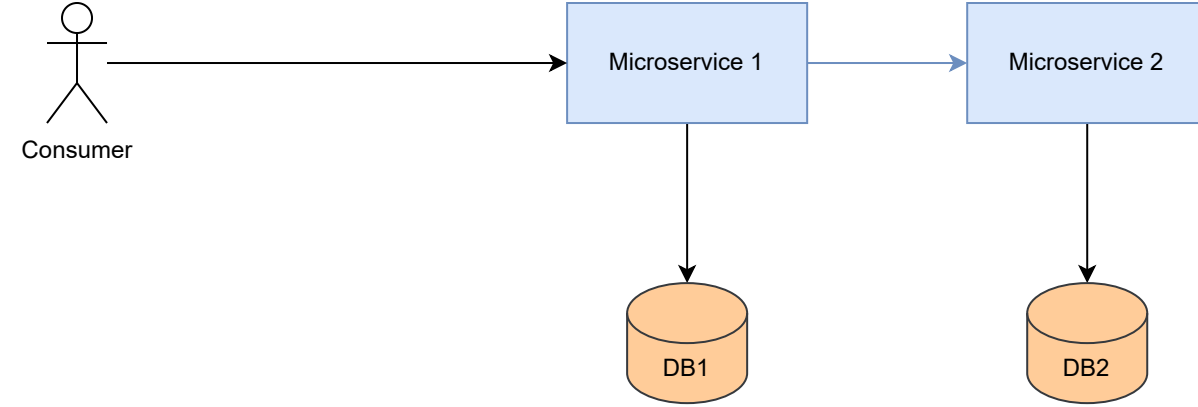


Teaching Principle

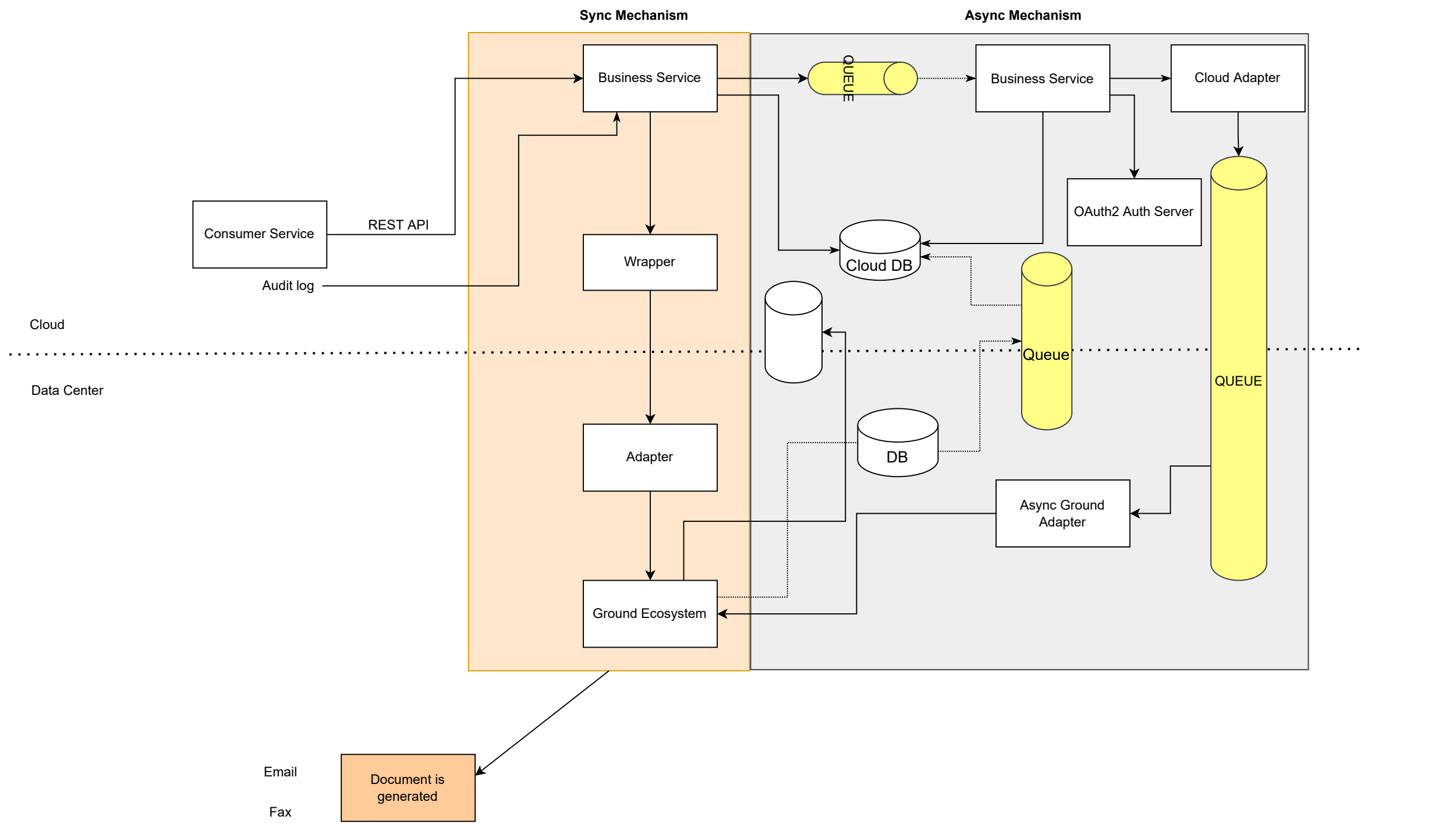


Learning Map

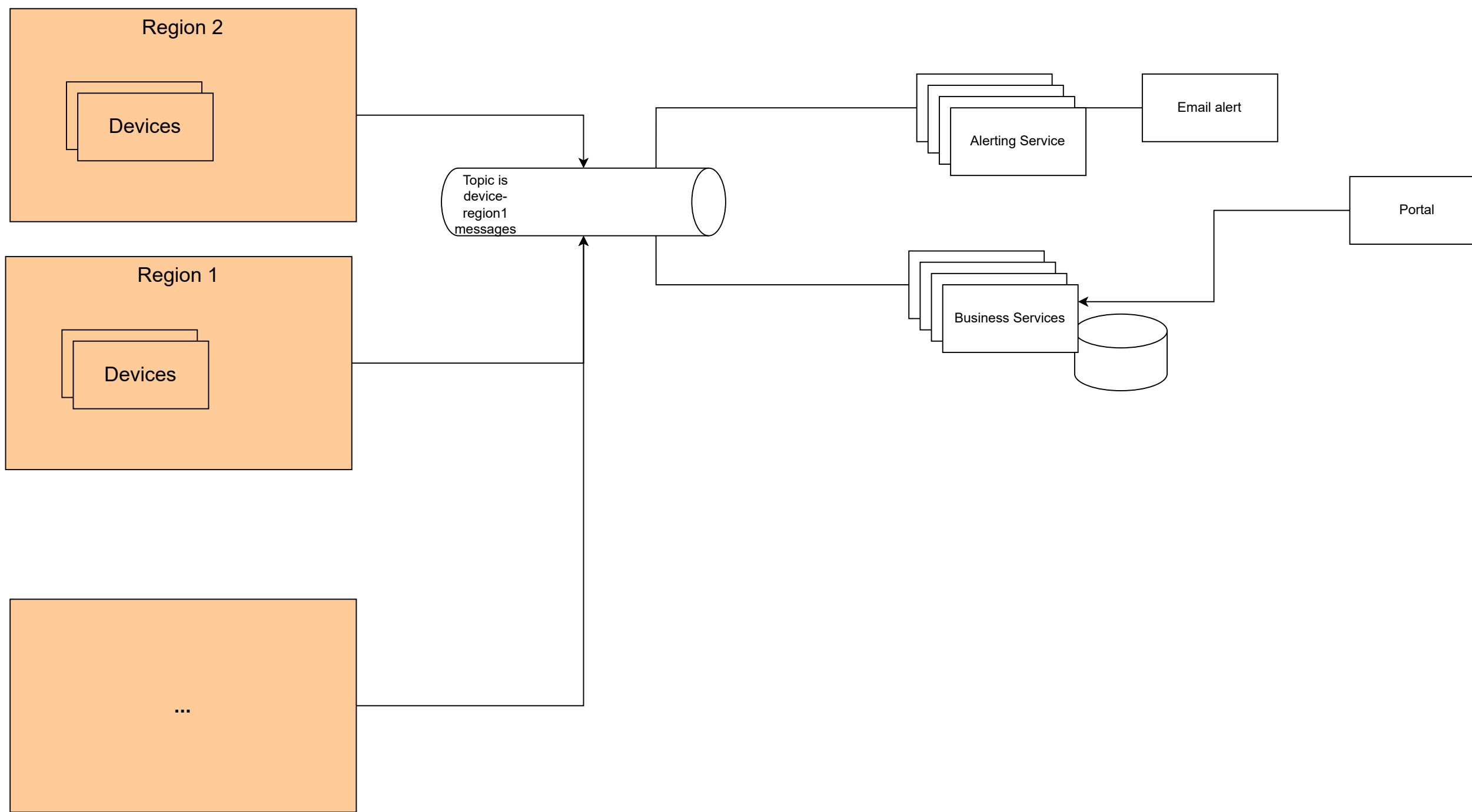




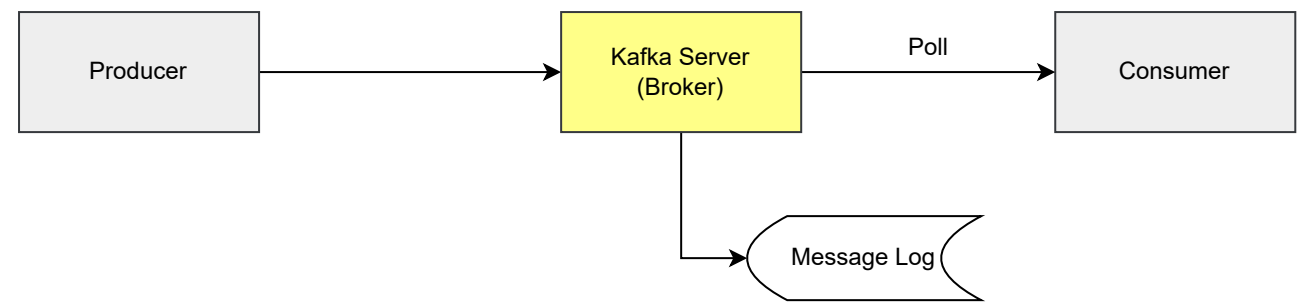
Use case 1



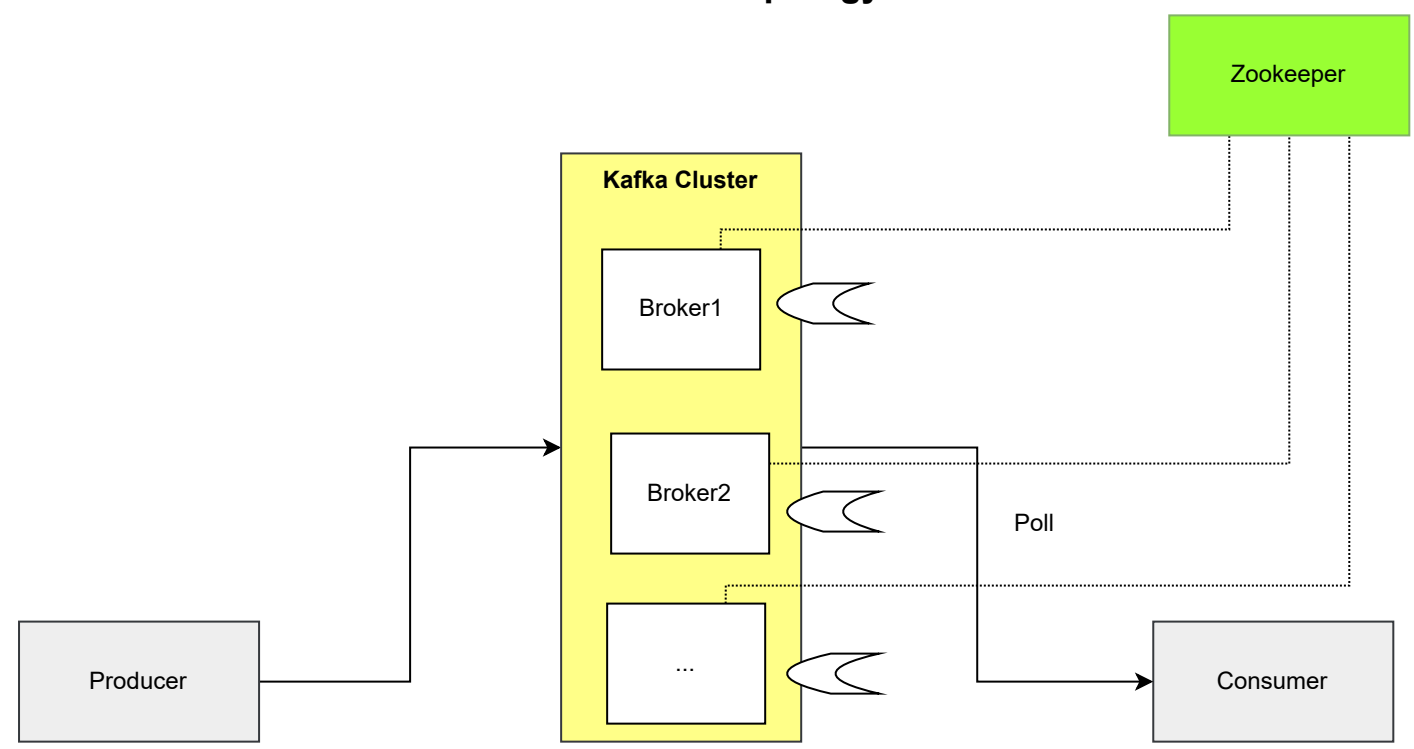
Use case 2



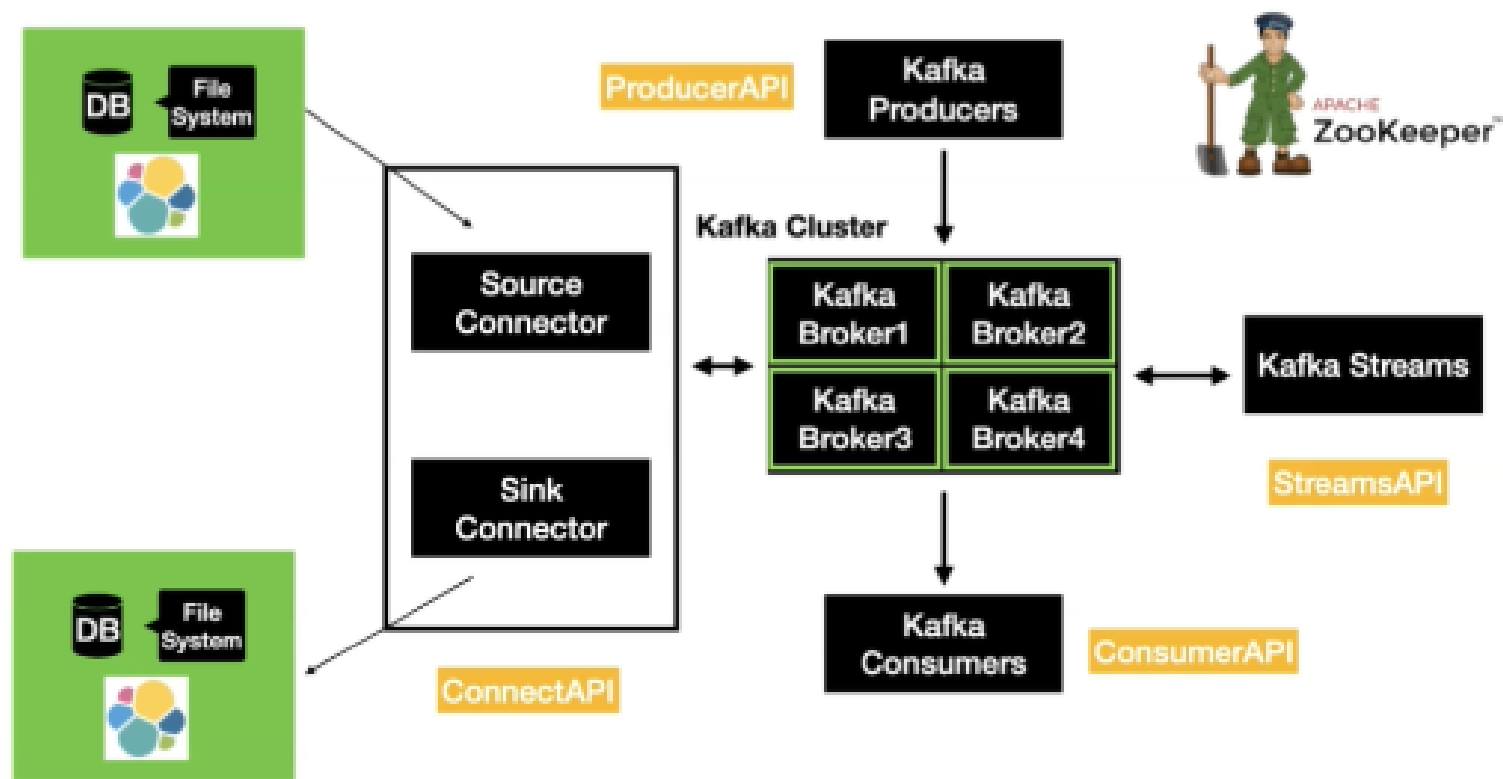
Basic Topology



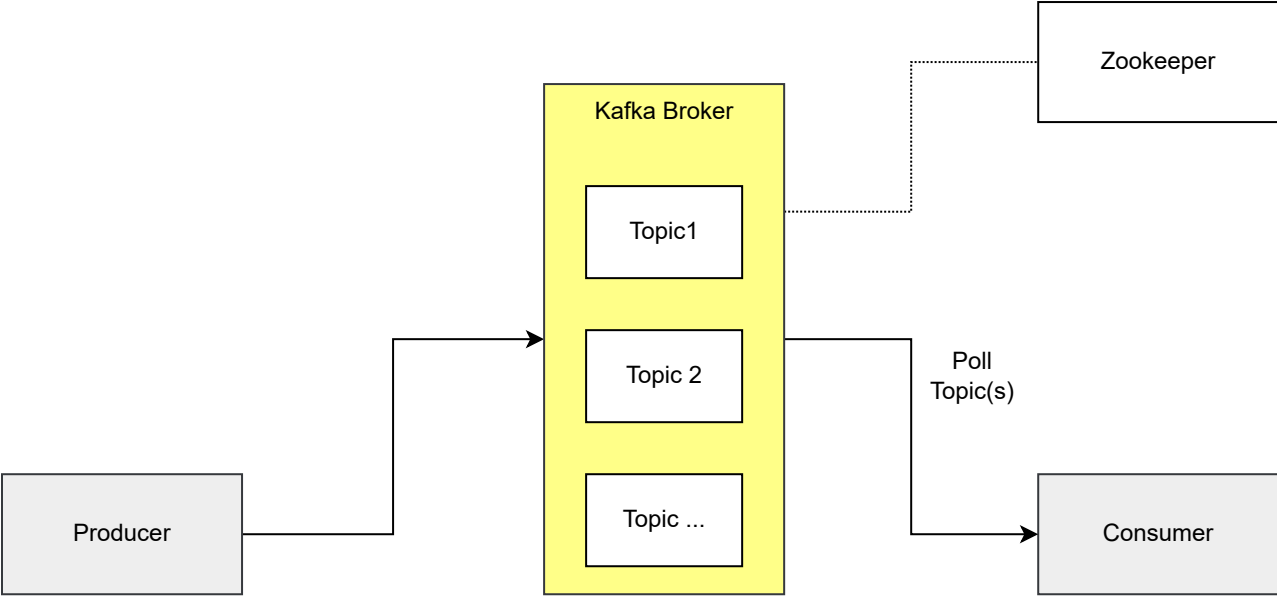
Distributed Topology



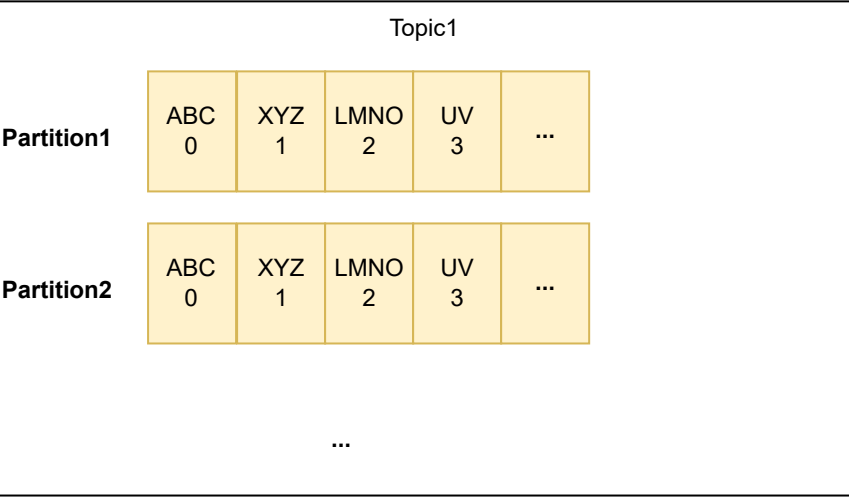
Kafka Landscape



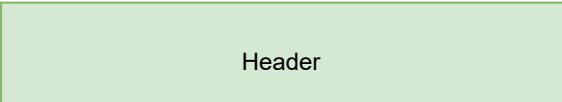
Broker Anatomy



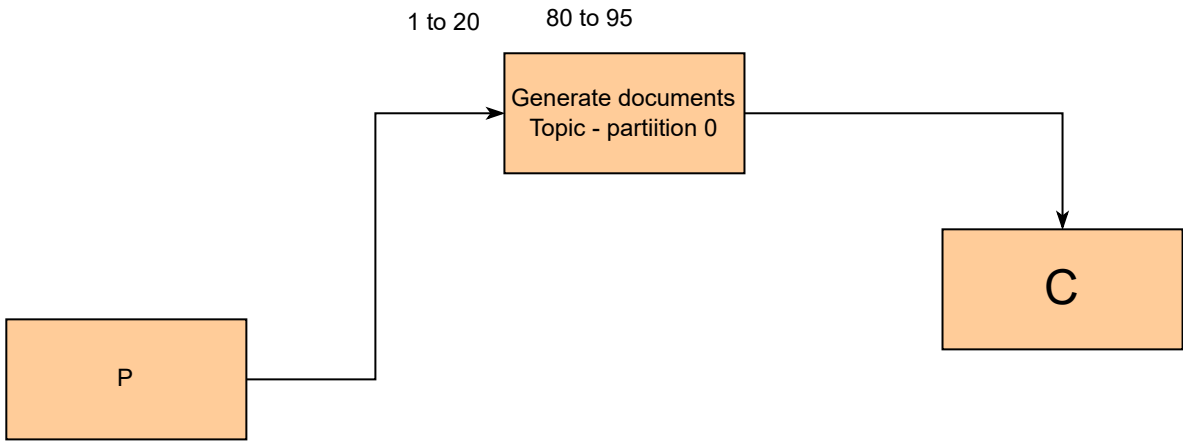
Topic Anatomy



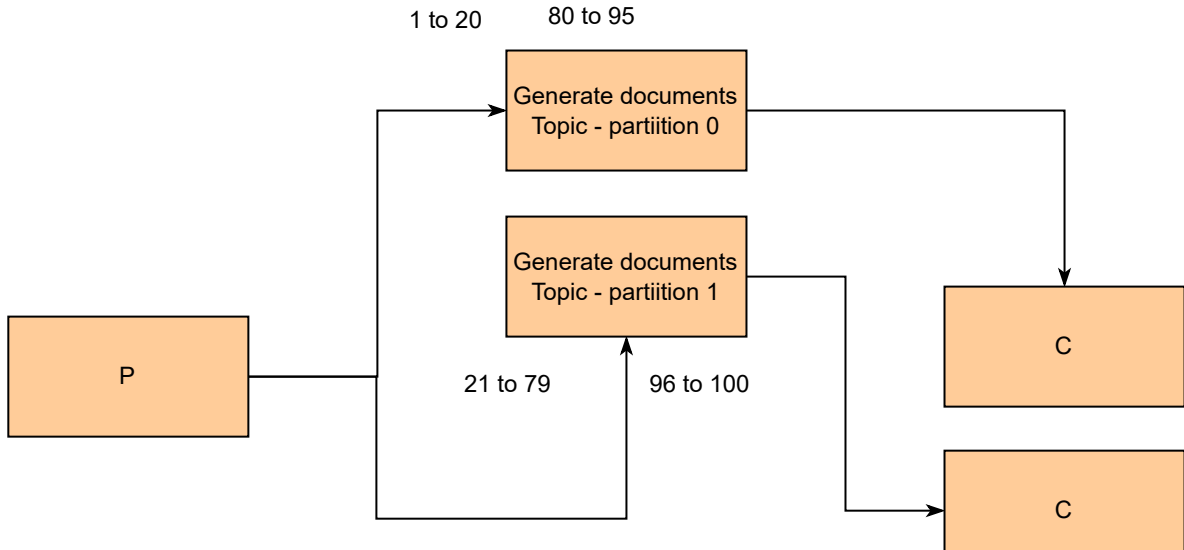
Message Anatomy



Topic-Same



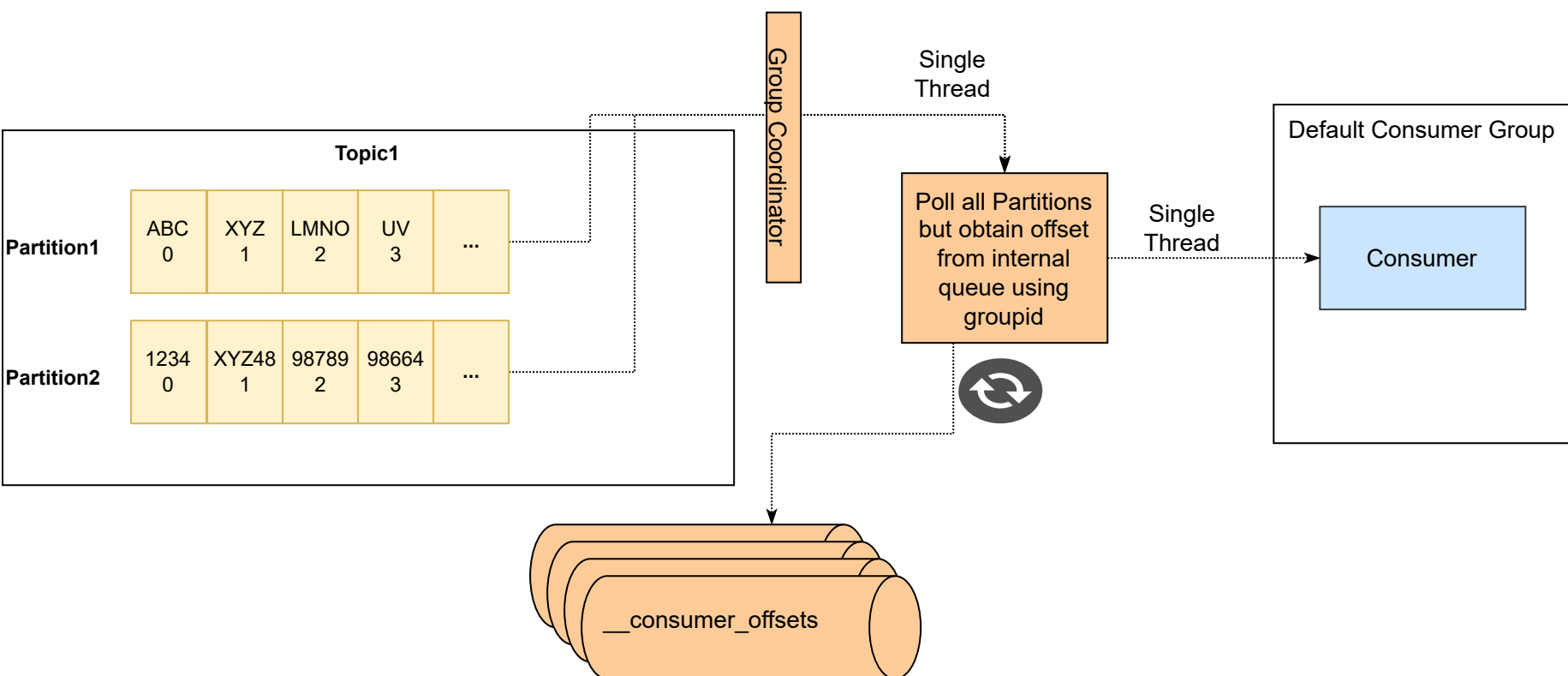
Topic-Same



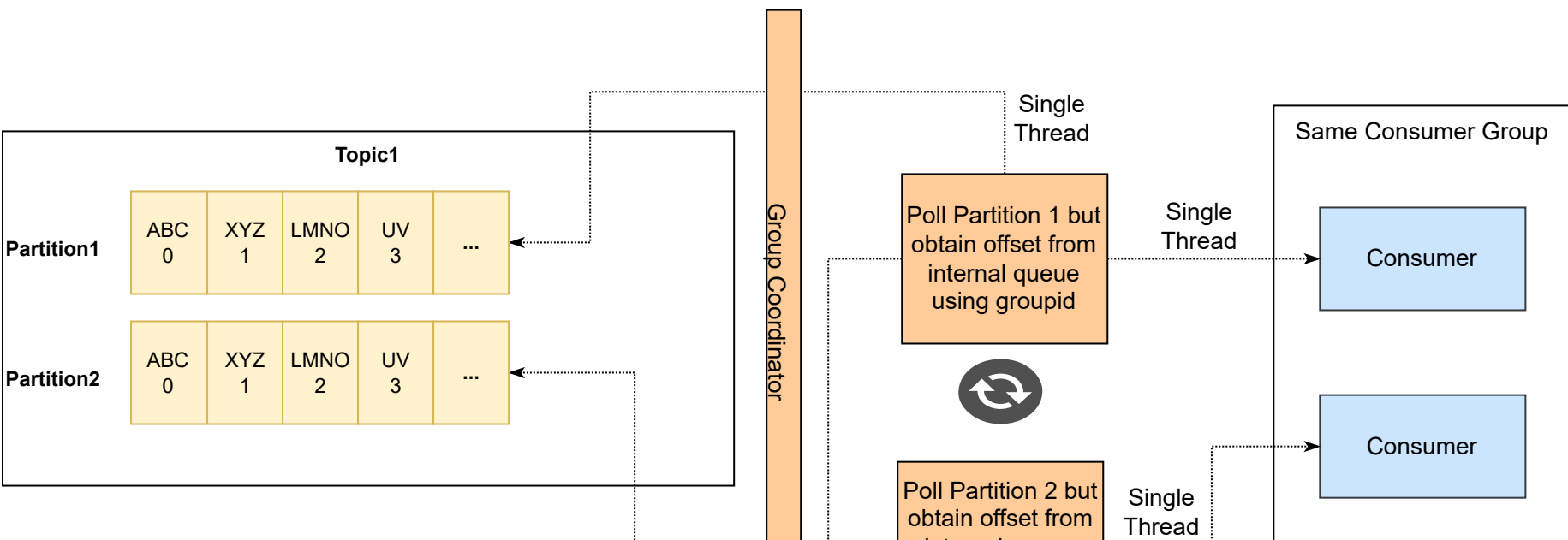
Key(optional)
Value
Timestamp

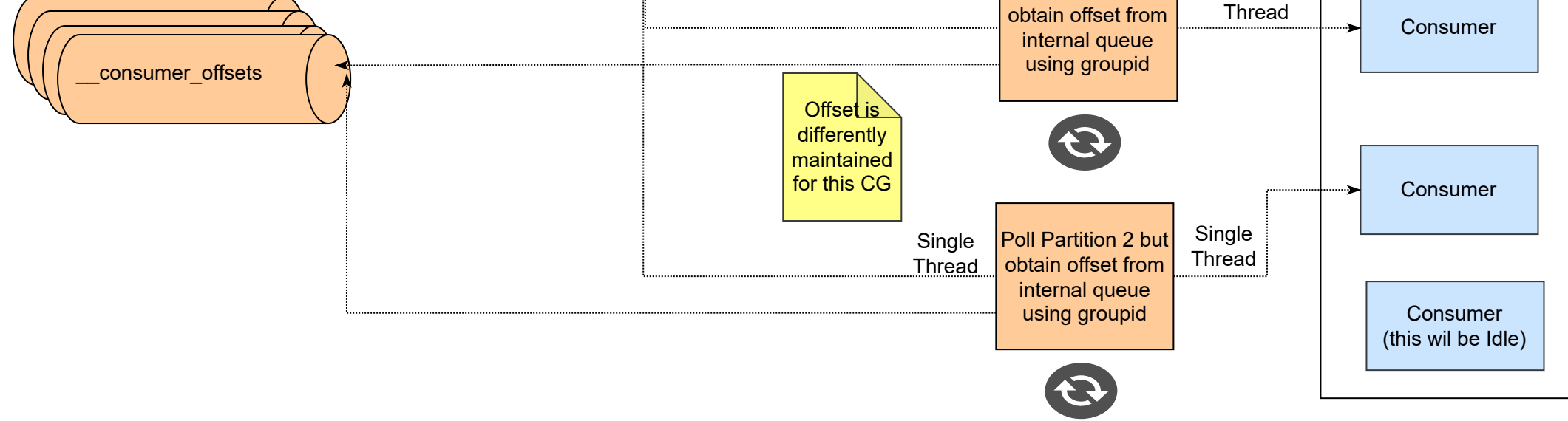


Default Consumer Group

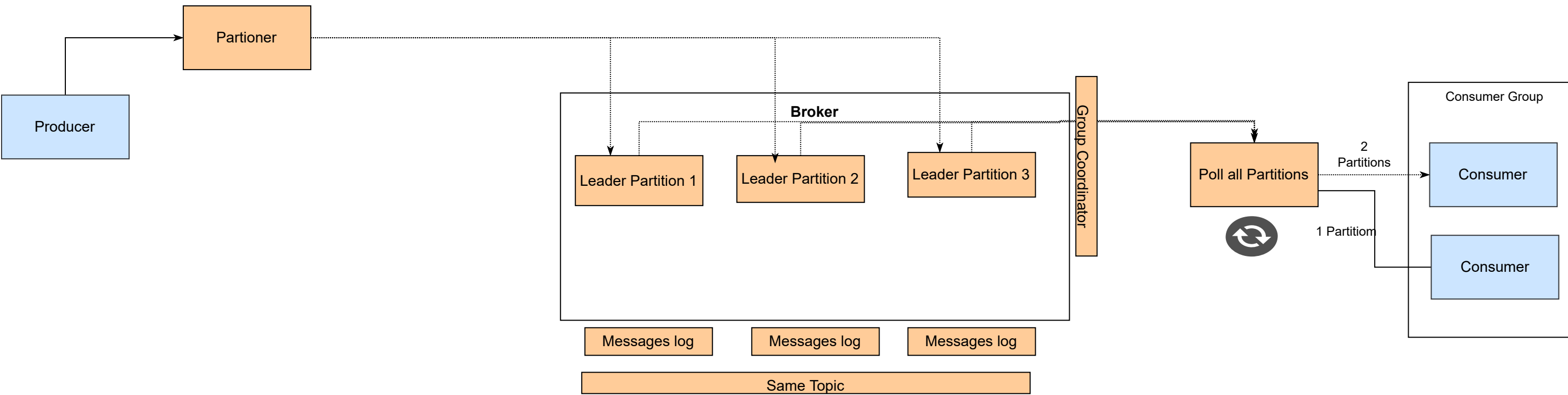


Using same Consumer Group to Read Messages at Scale

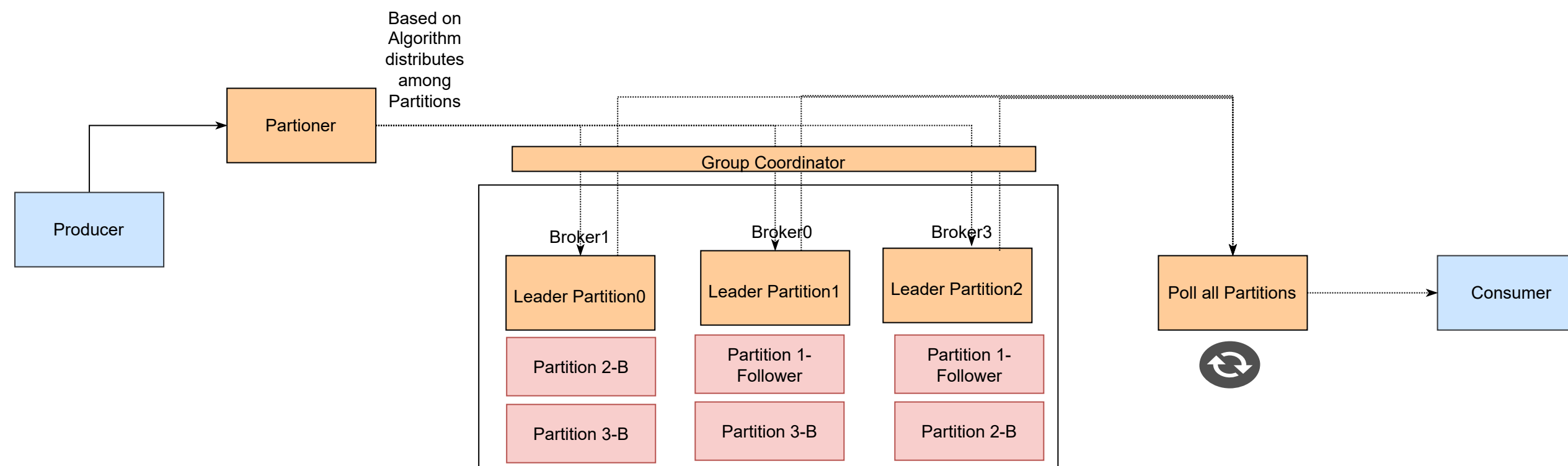




Standalone Topology



Distributed Topology



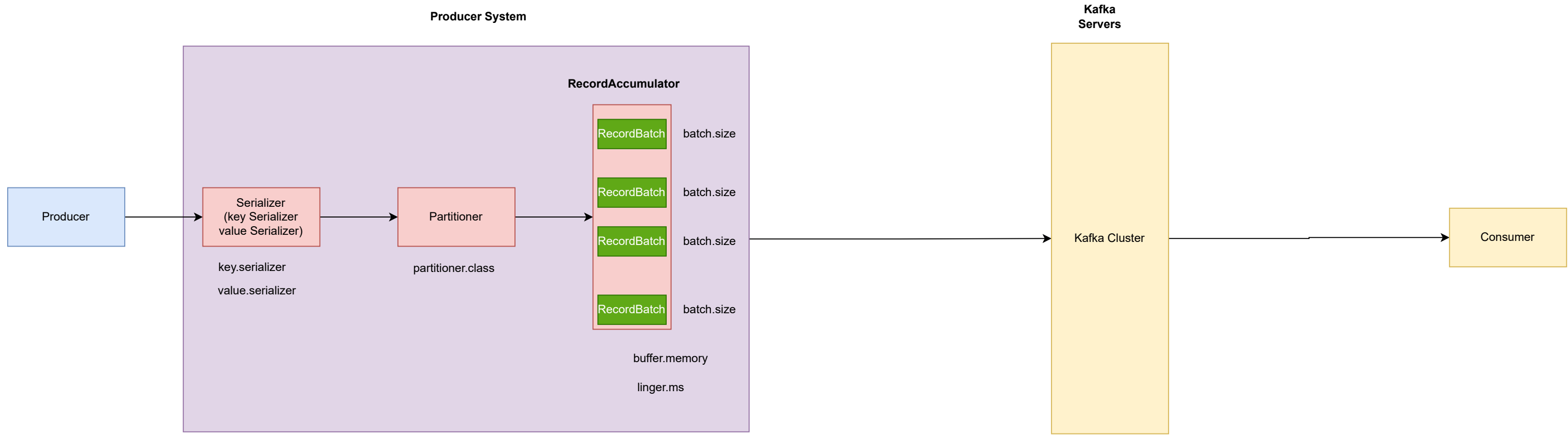
Messages log

Messages log

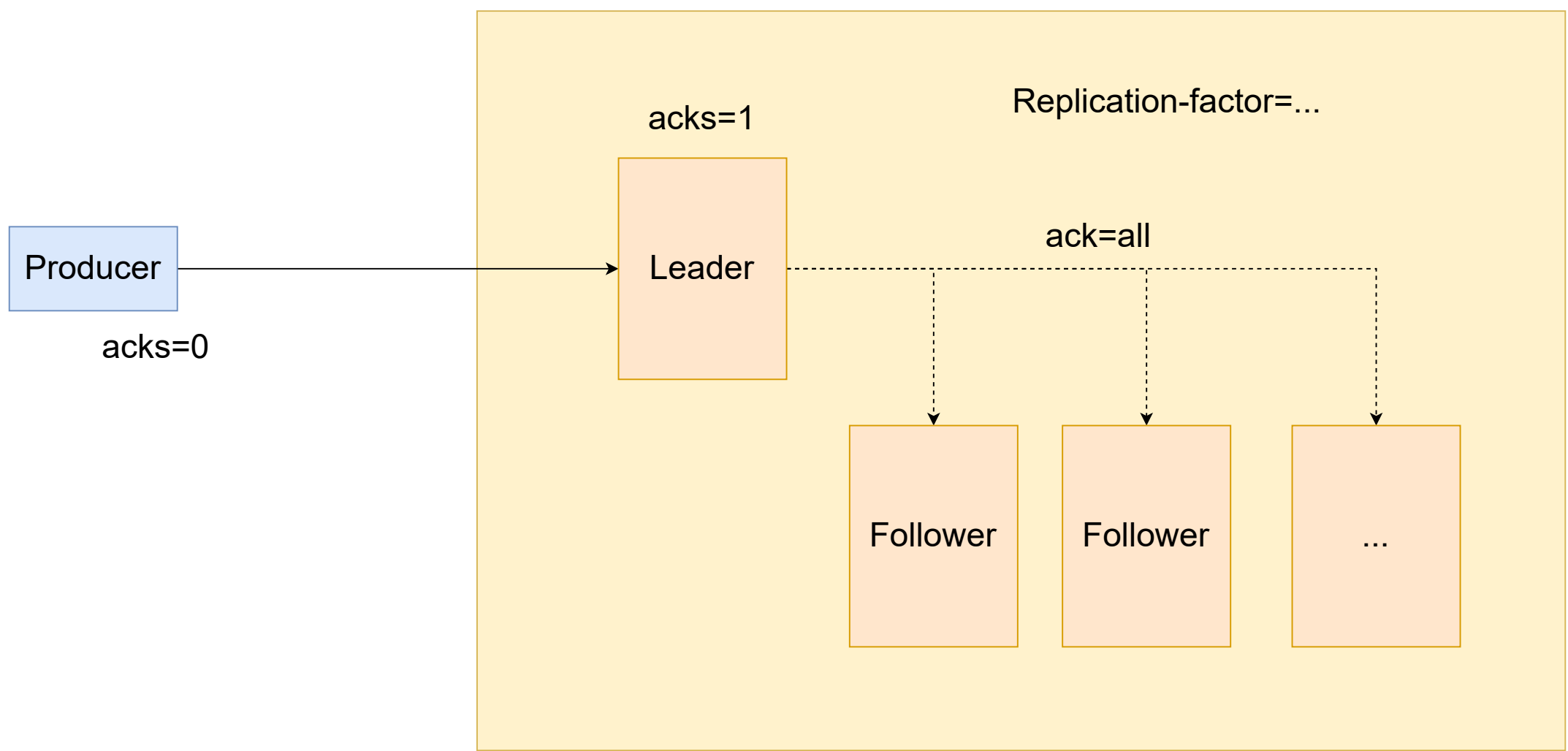
Messages log

Same Topic

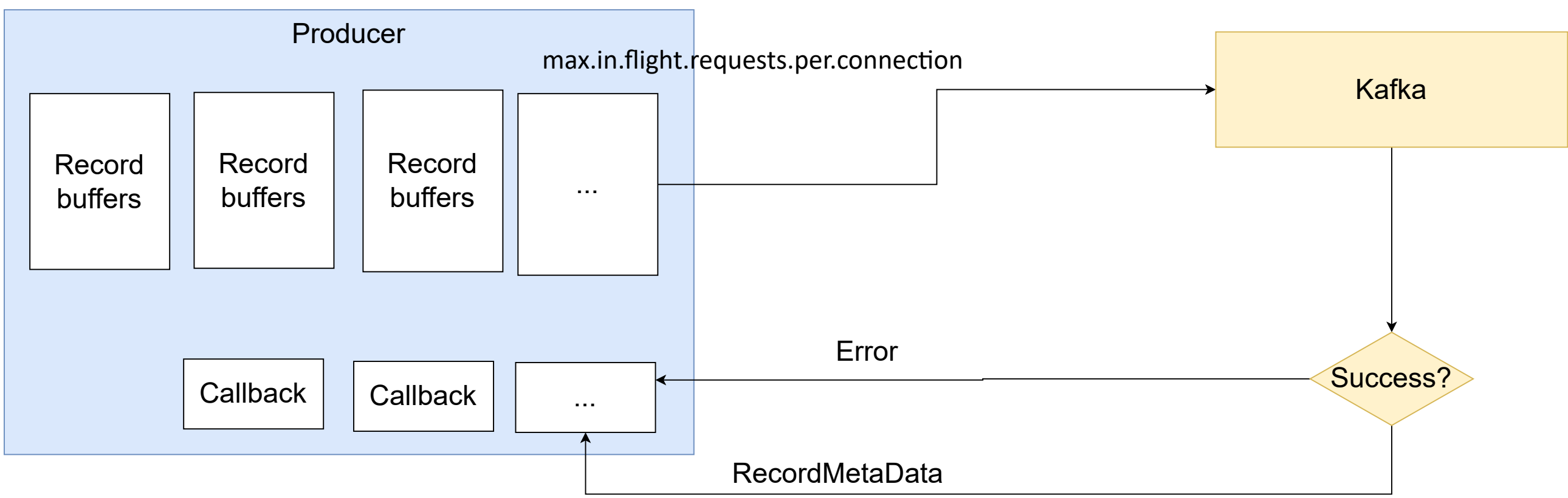
Producer Application Internals & Configurations



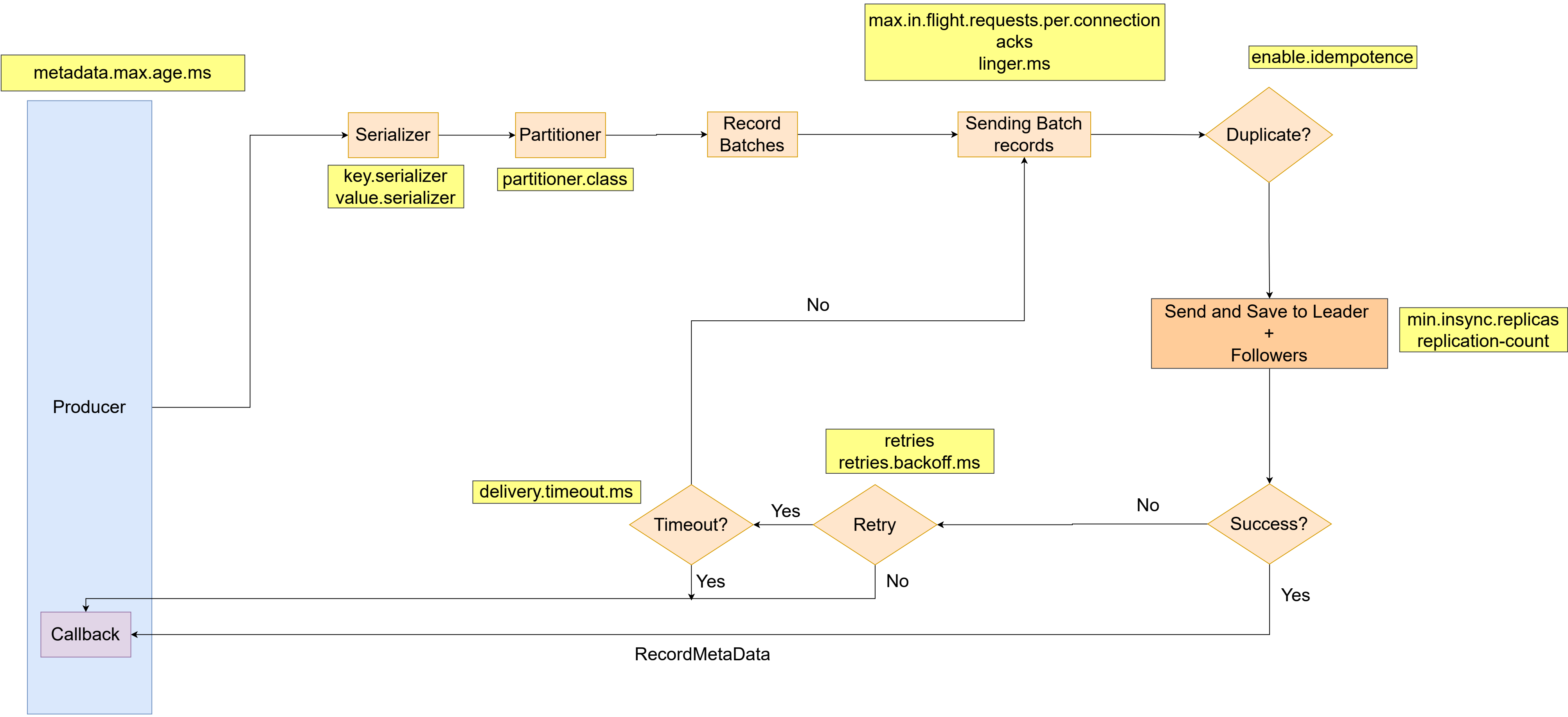
ack configuration



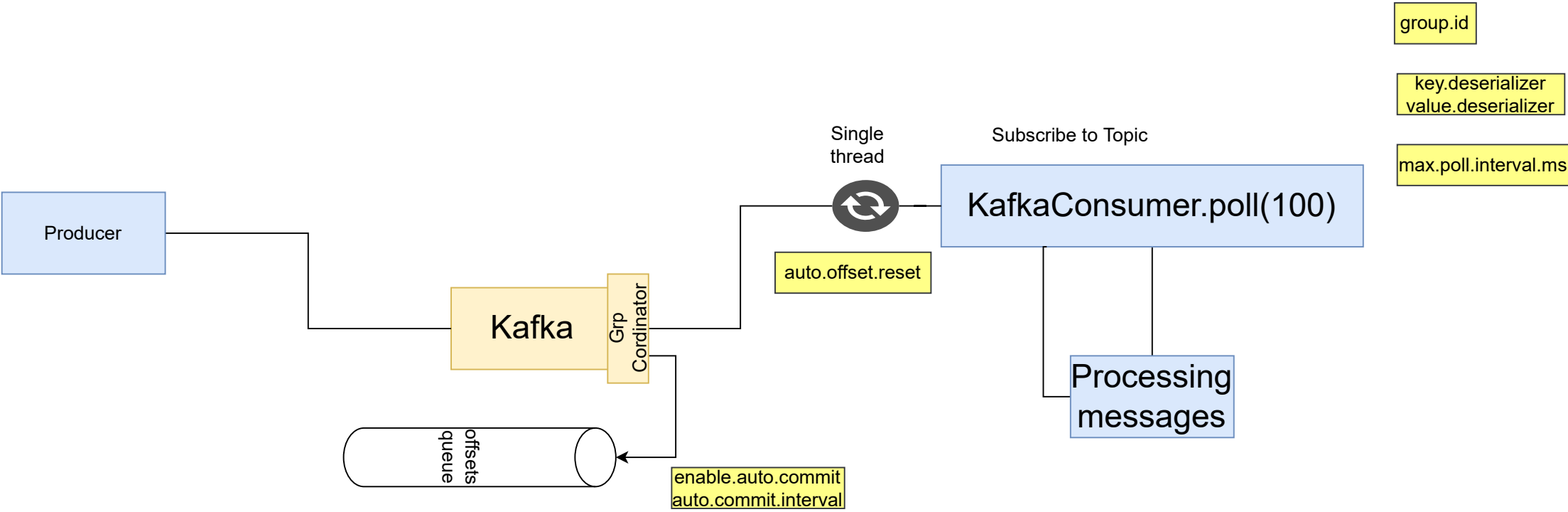
max.in.flight.requests.per.connection Configuration



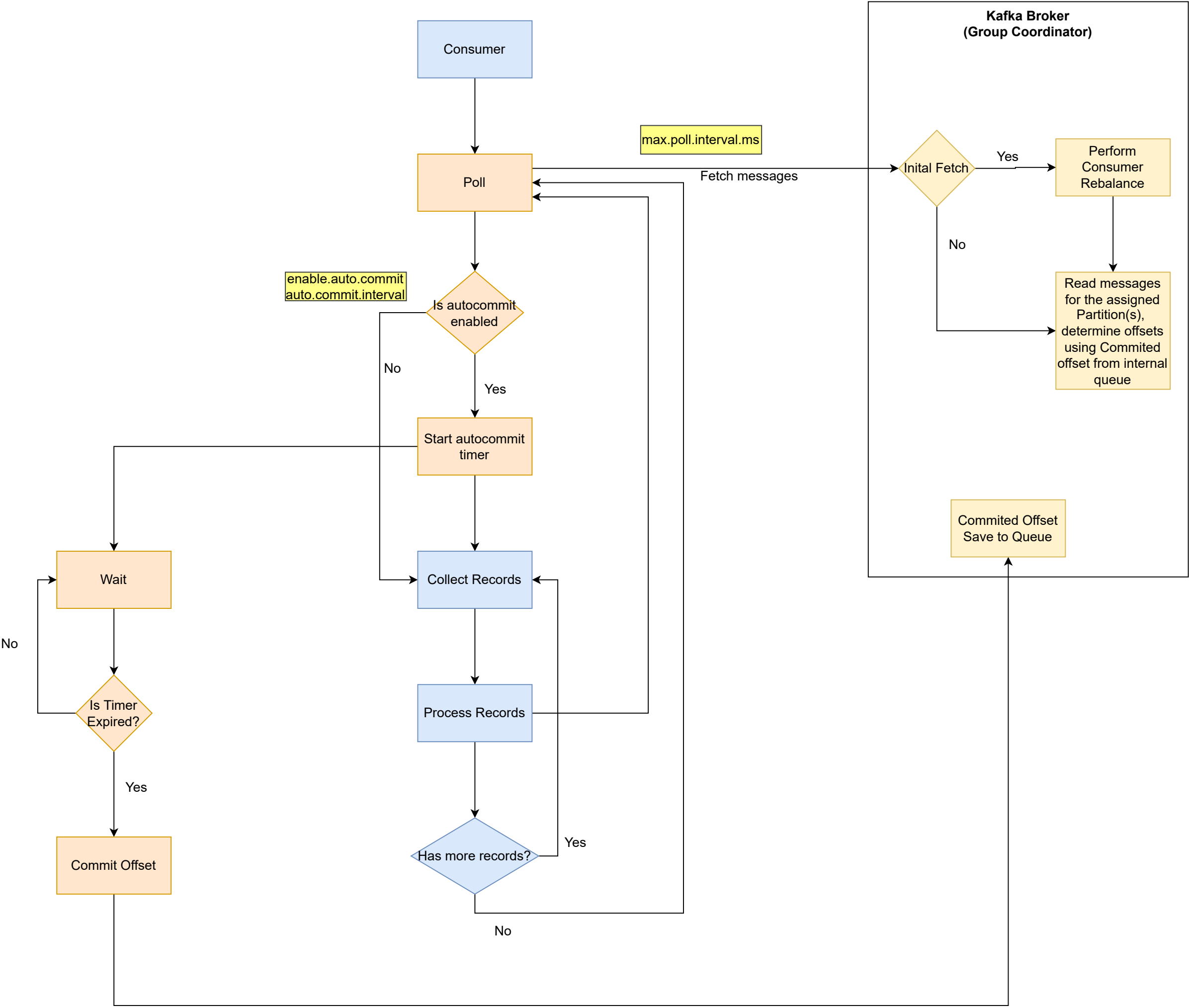
Consolidated Flow



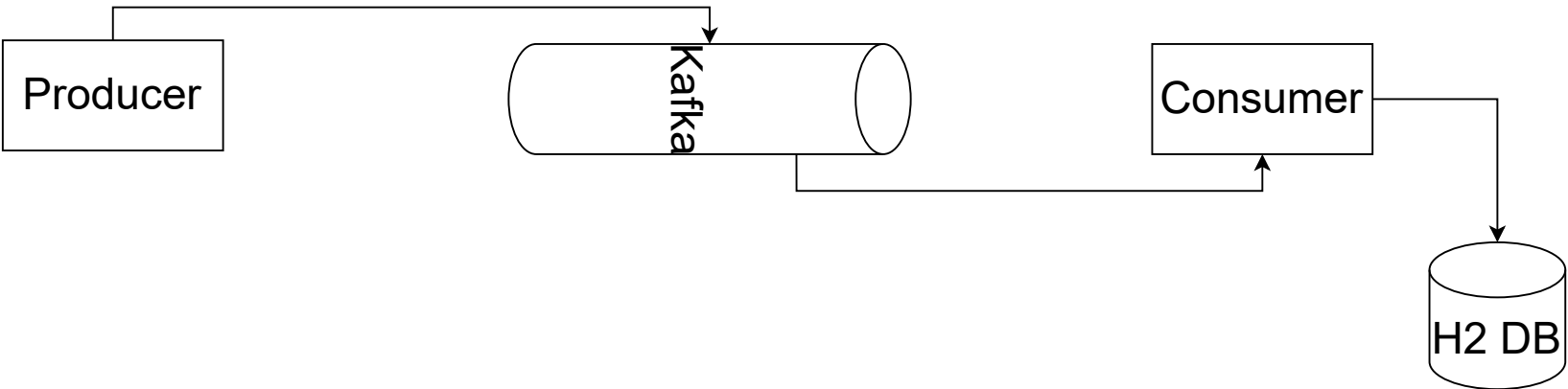
Consumer Pooling

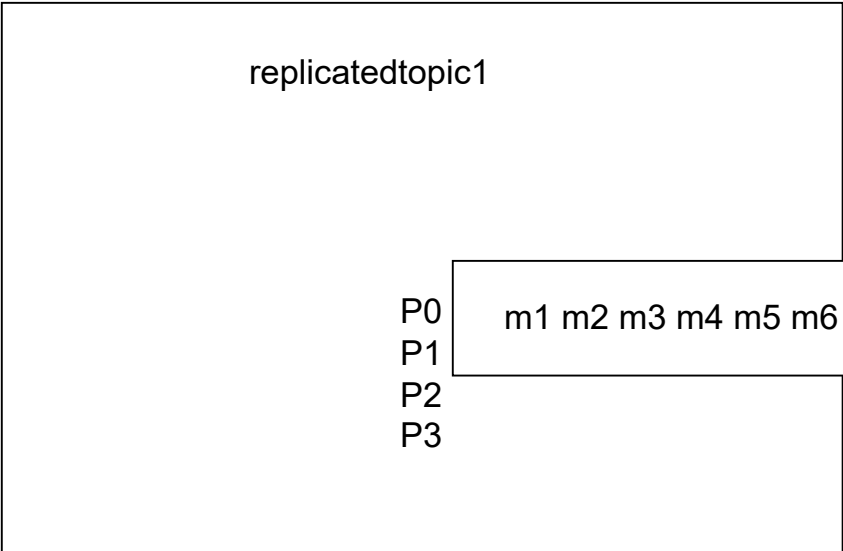
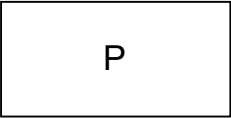


Consumer Internal Working

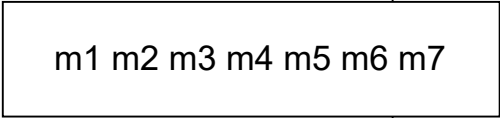


Persistency of read messages





P0
P1
P2
P3

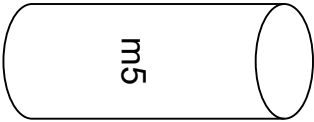


m6 m7

m1 m2 m3 m4 m5

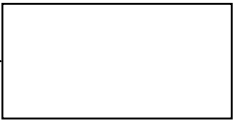
Current offset

c

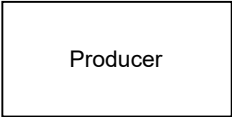


Committed offset

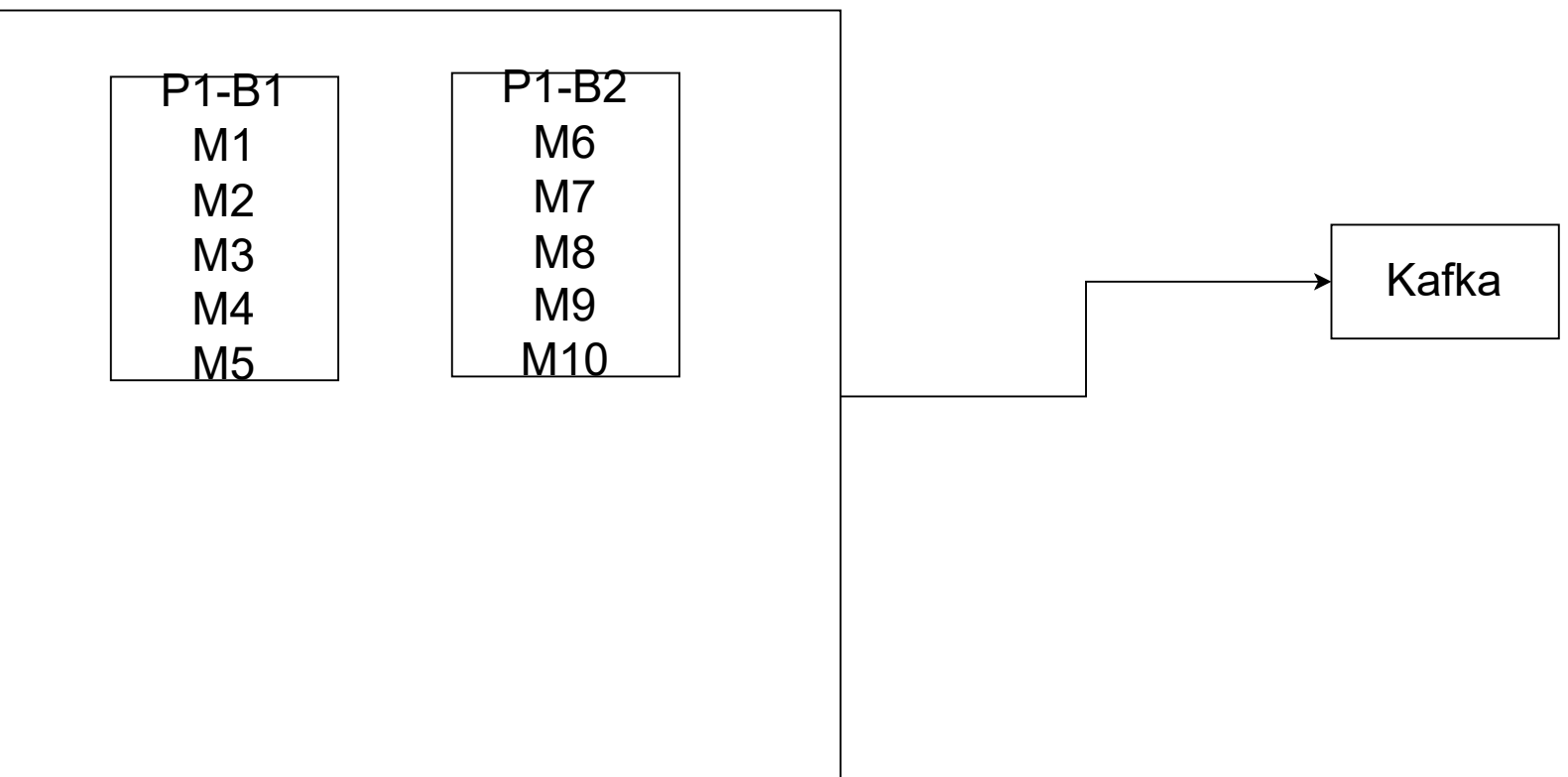
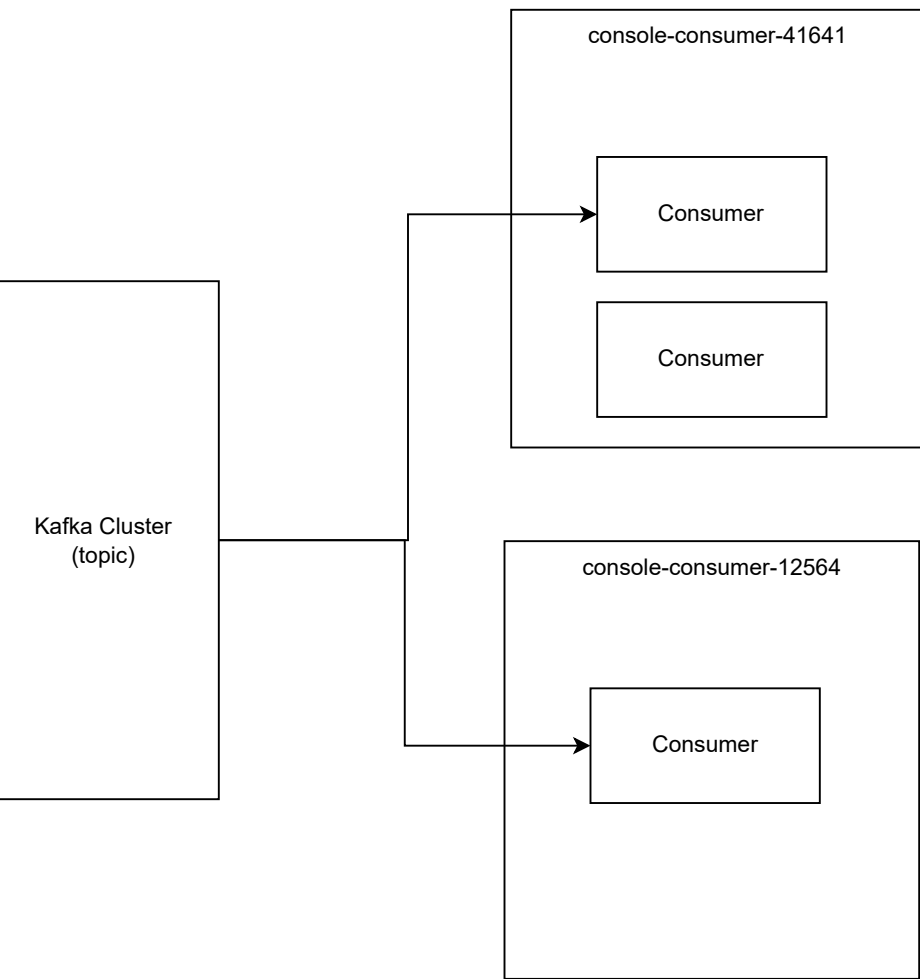
Partitions

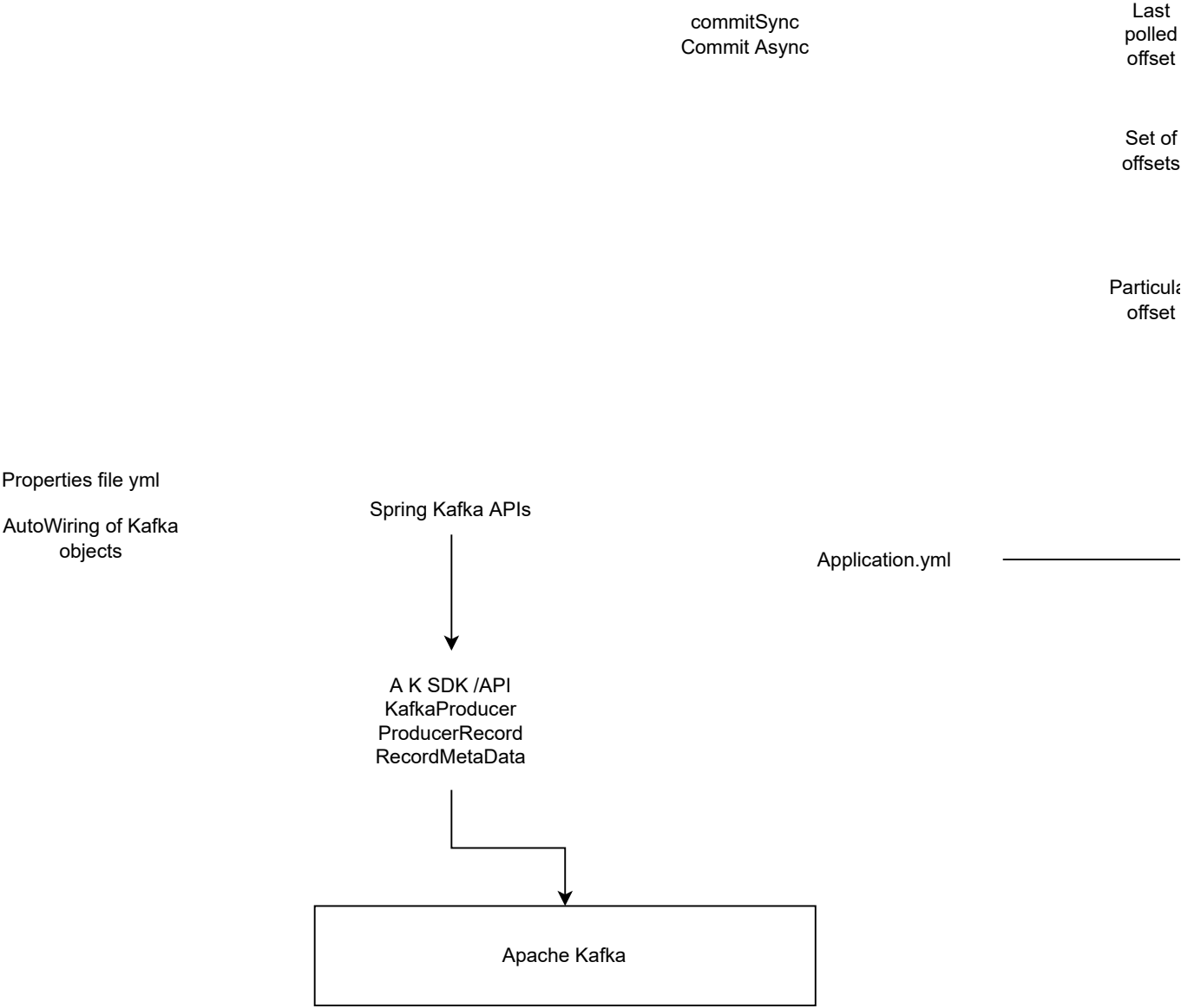


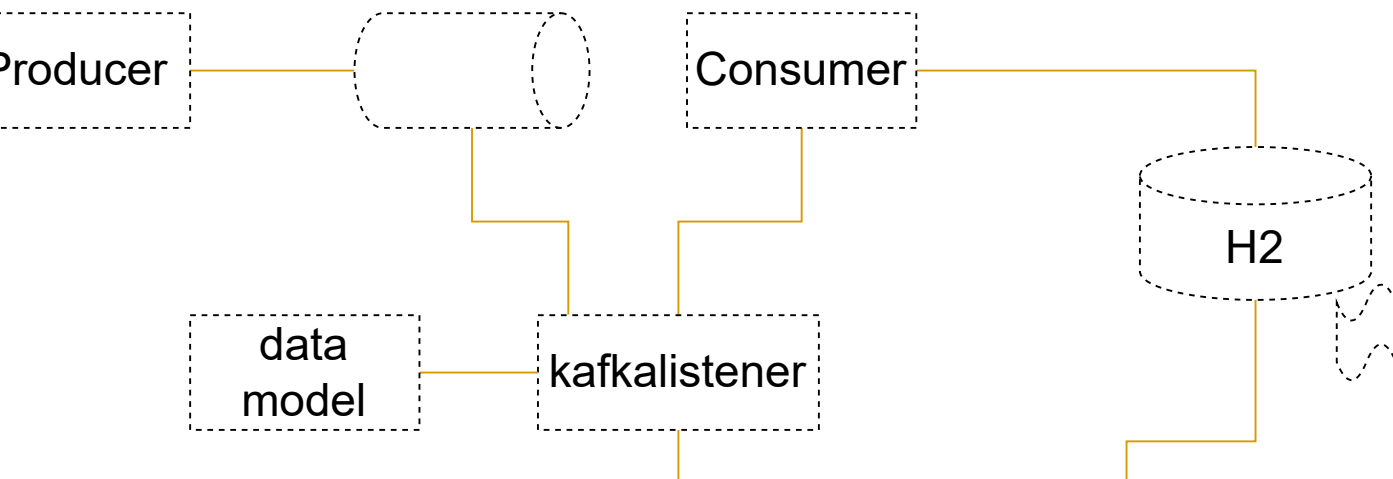
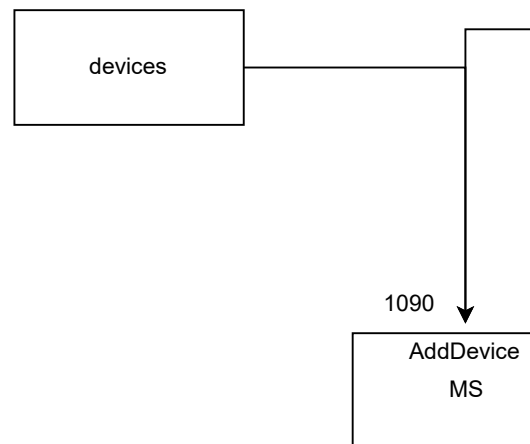
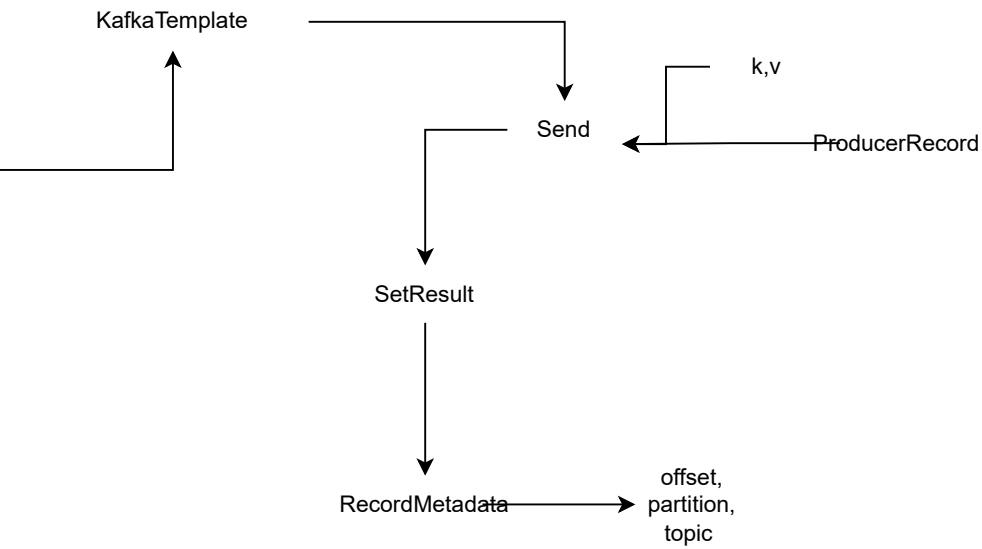
Config Kafkaconsu
Subscrib
Polling
Revocati
Assignme

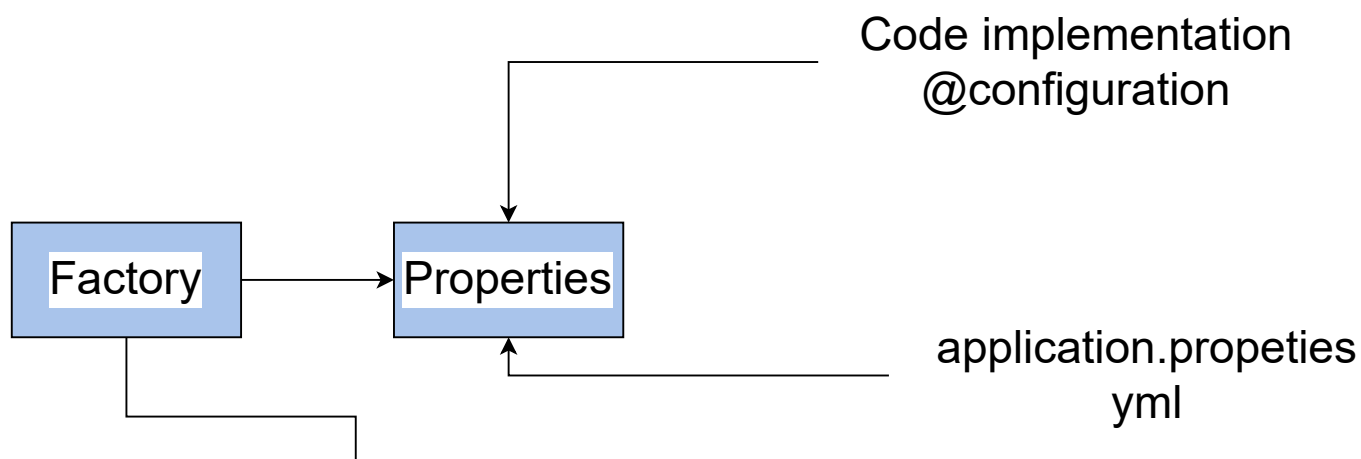
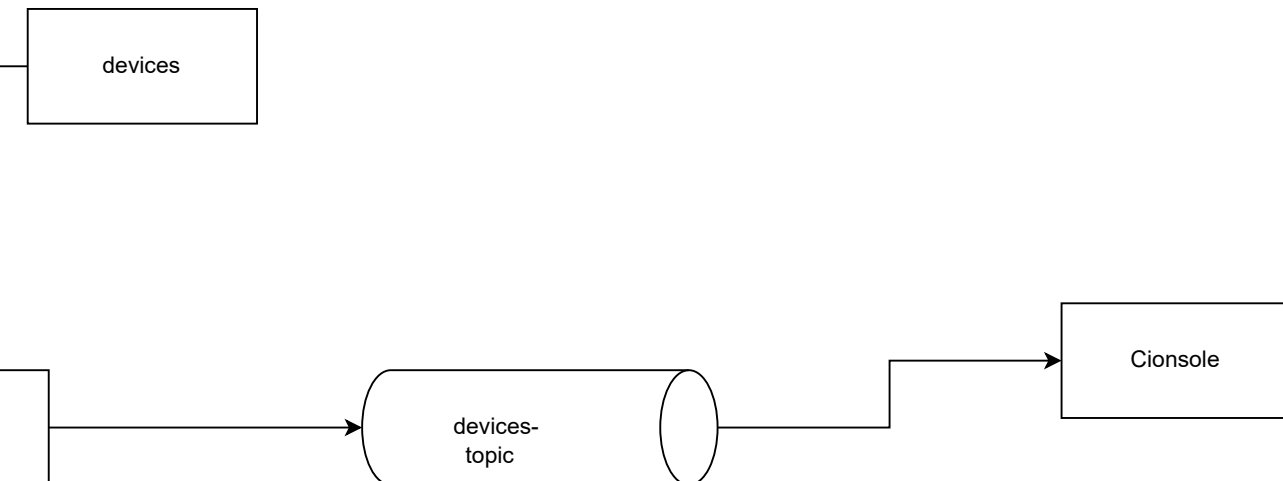


umer
be
on
ent









me

message

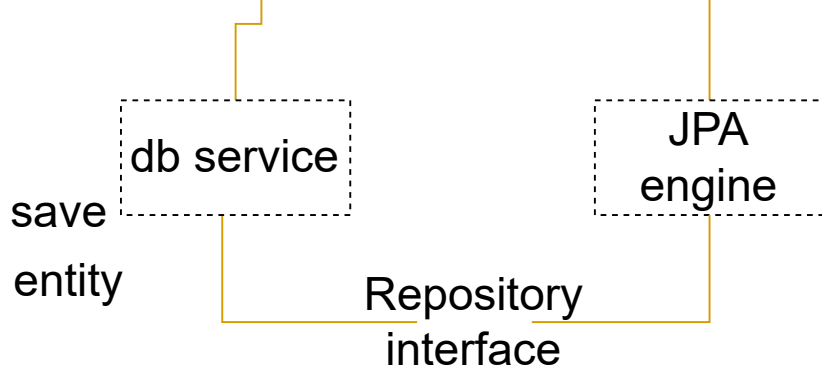
Queue

Partion

Partion

Partion

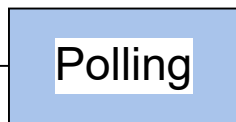




message message

message

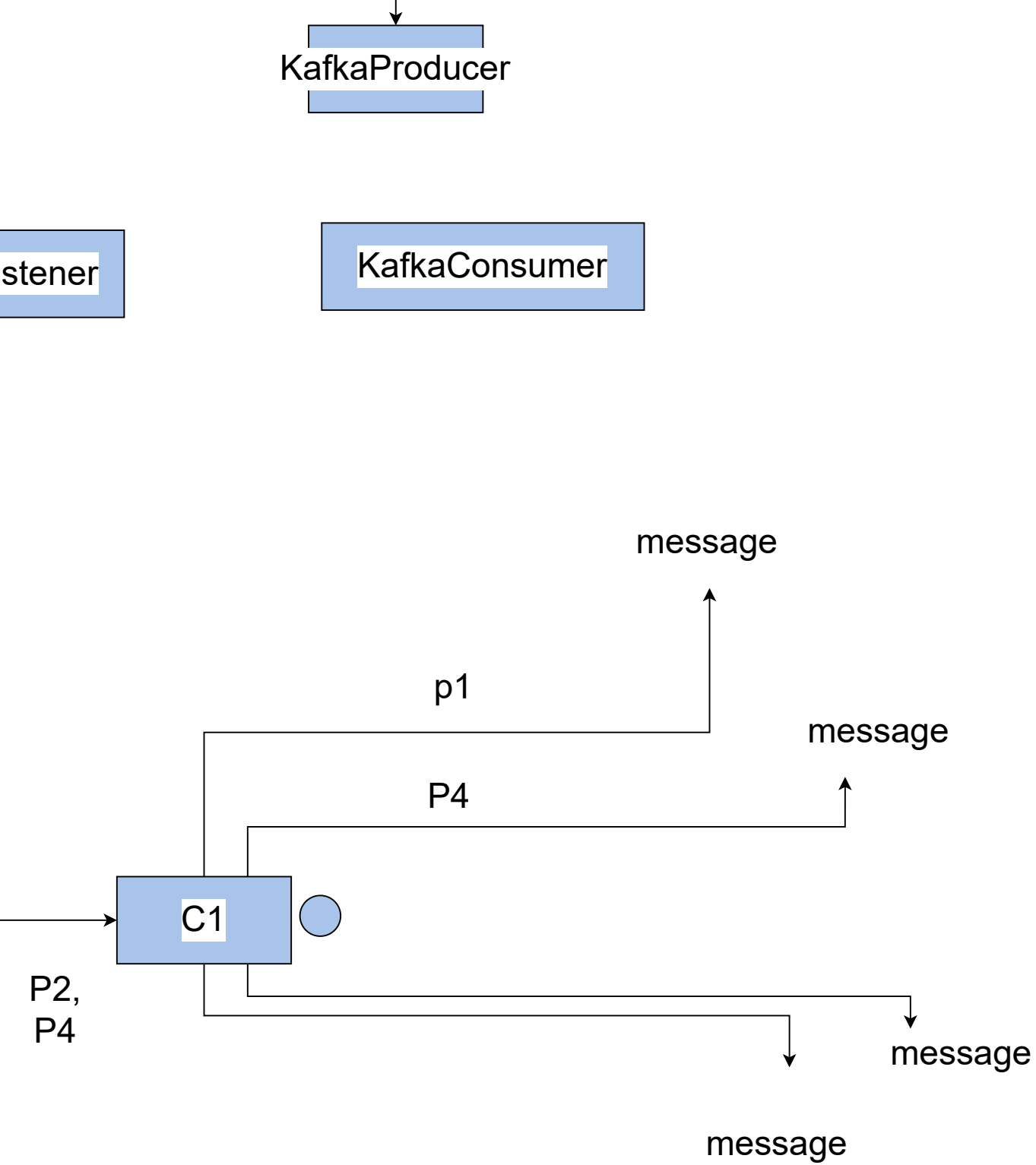
message message



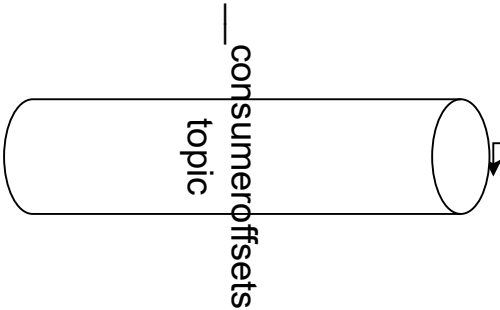
thread 1

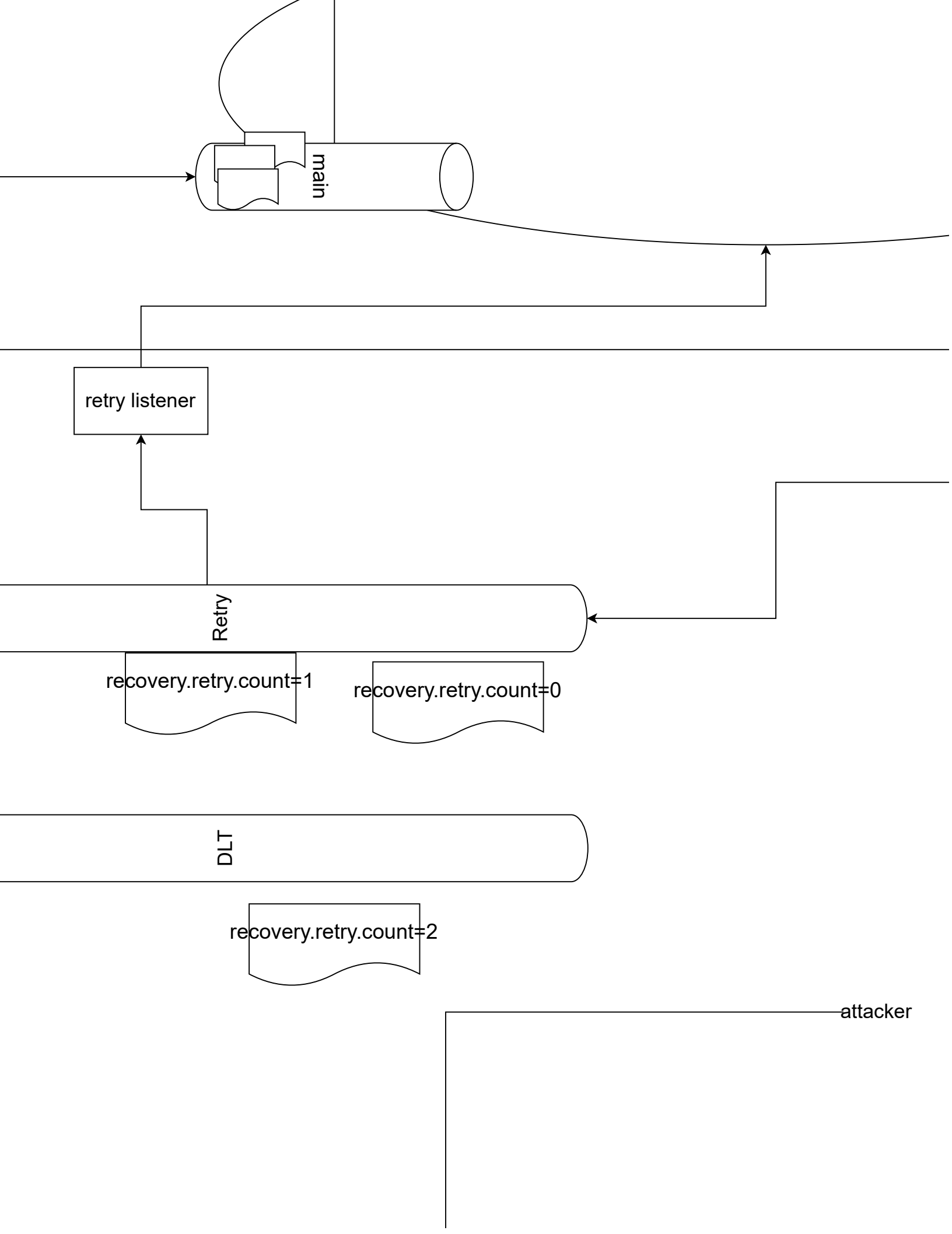
p1,
p2,
p3,
p4

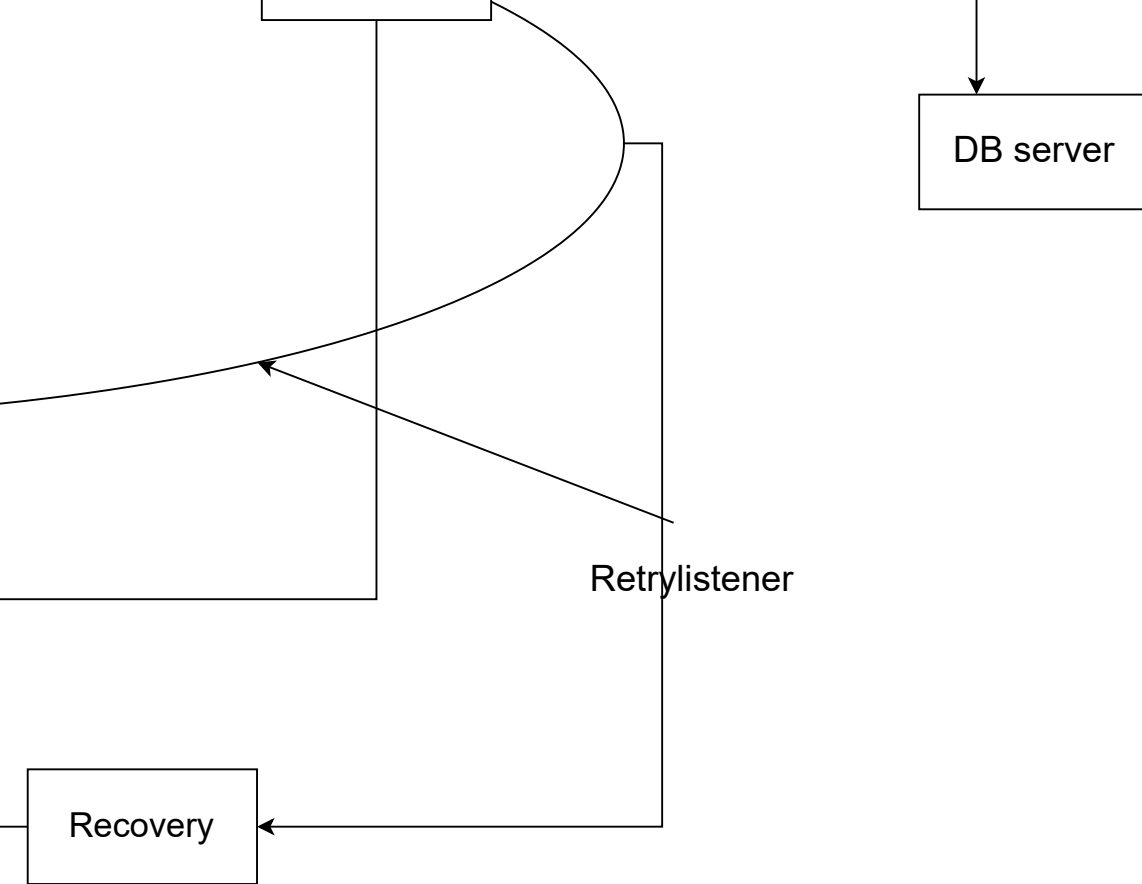
Spring Kafka controller retry



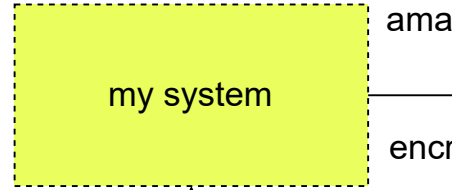
Producer







Trust store



SSL / TLS certificate
public
key

public
key

server

client

root
CA
certs

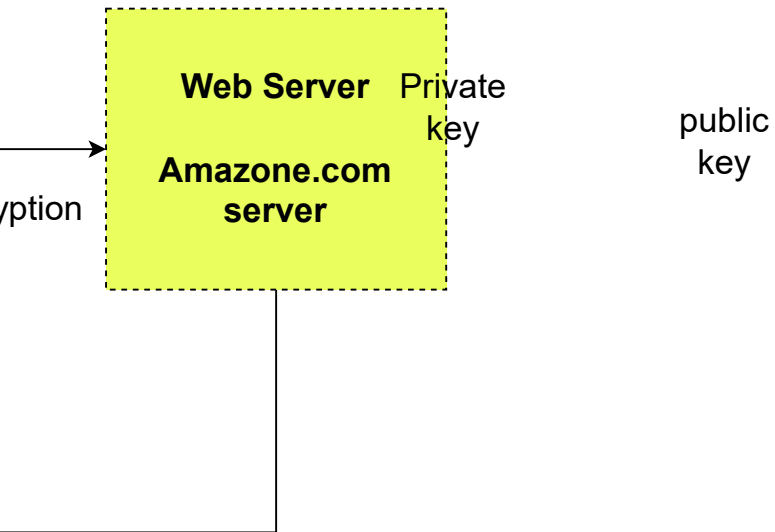
trusted store

signed certificate

keystores



Client



not
CA
certs

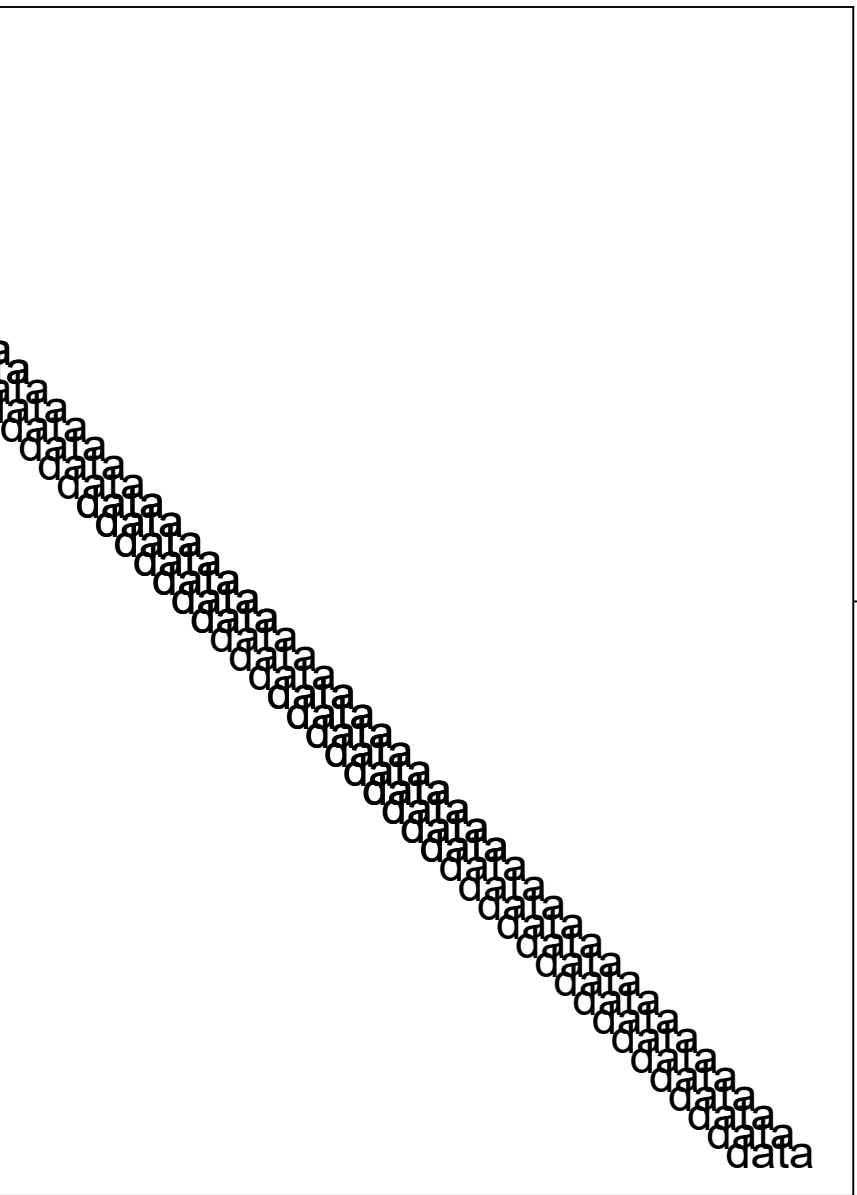
t CA cert

d certificate

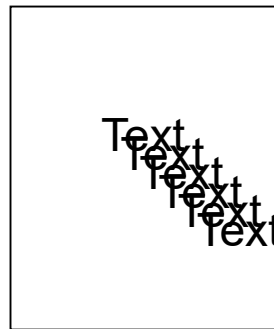
CA

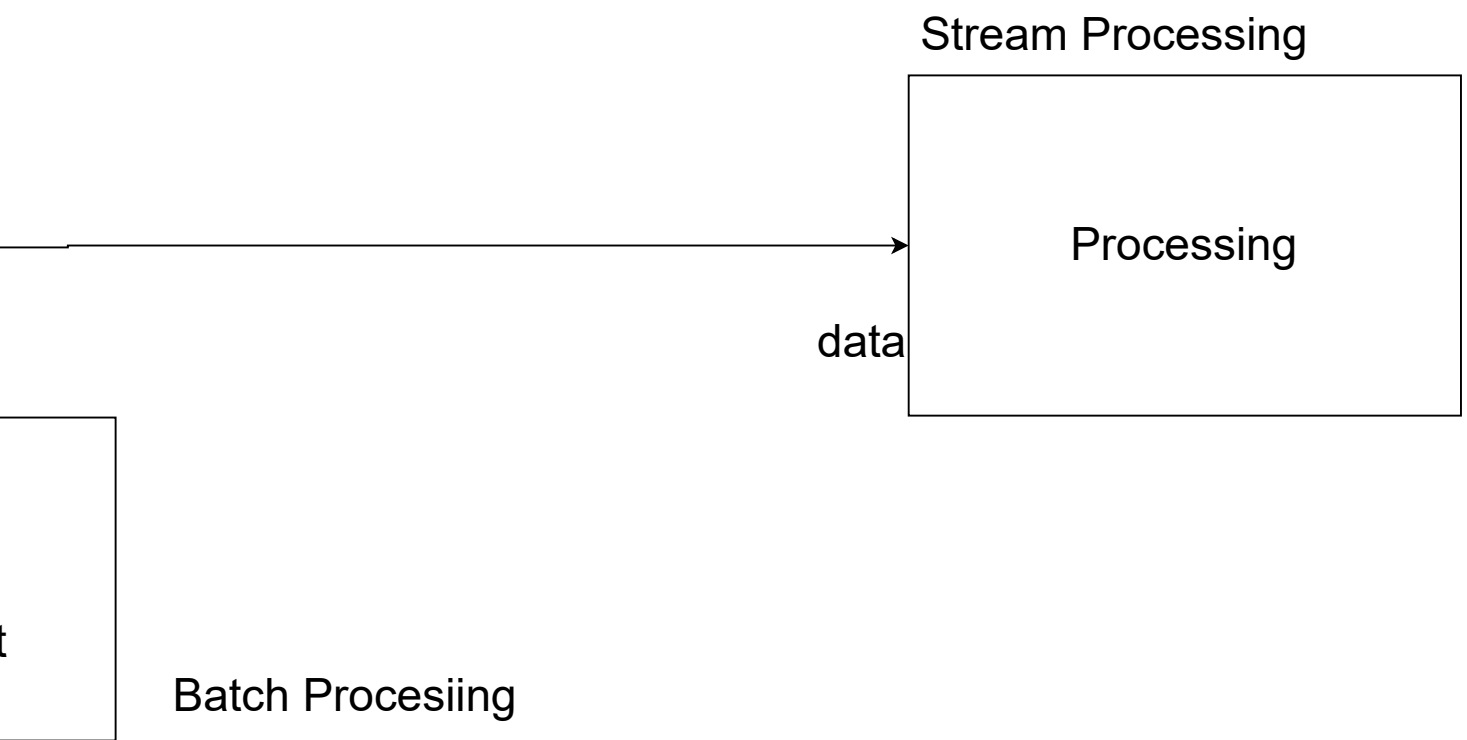
SSL certificates

SSL
- Encryption/ Decryption
- Authentication



StringBuilder







Kafka
configuration



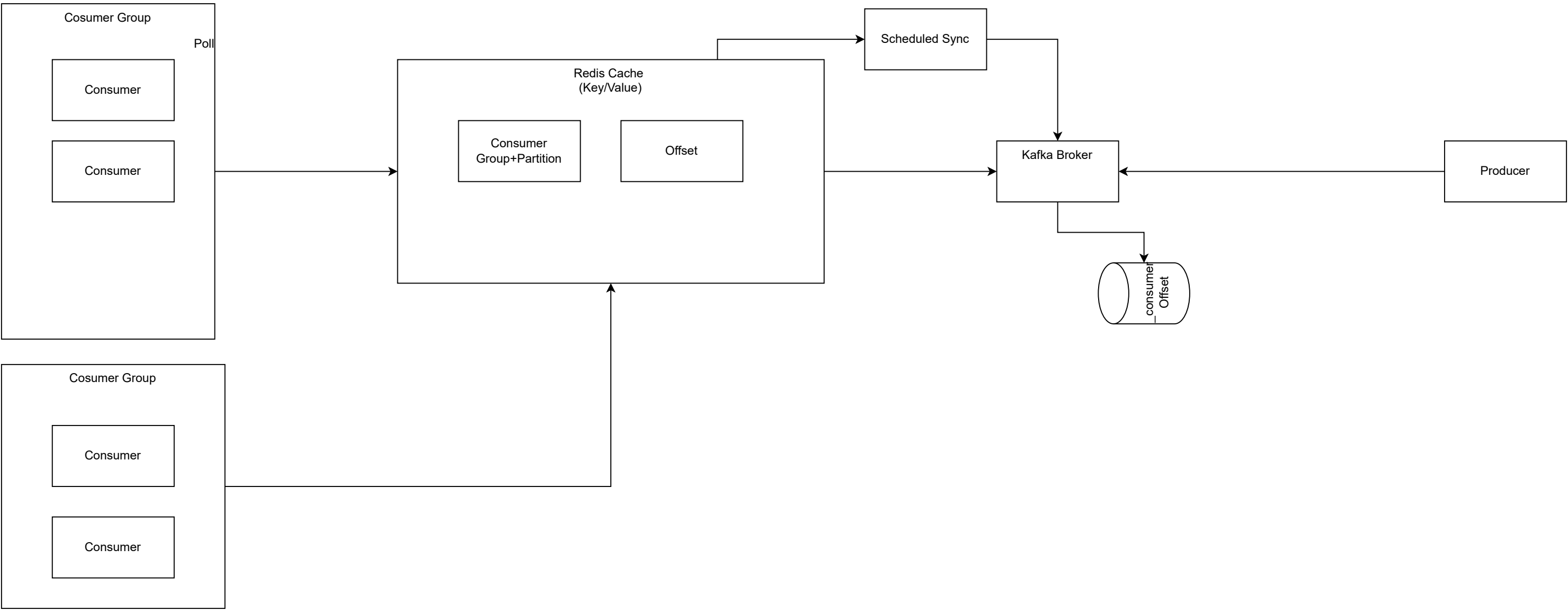
S
fr

Topology

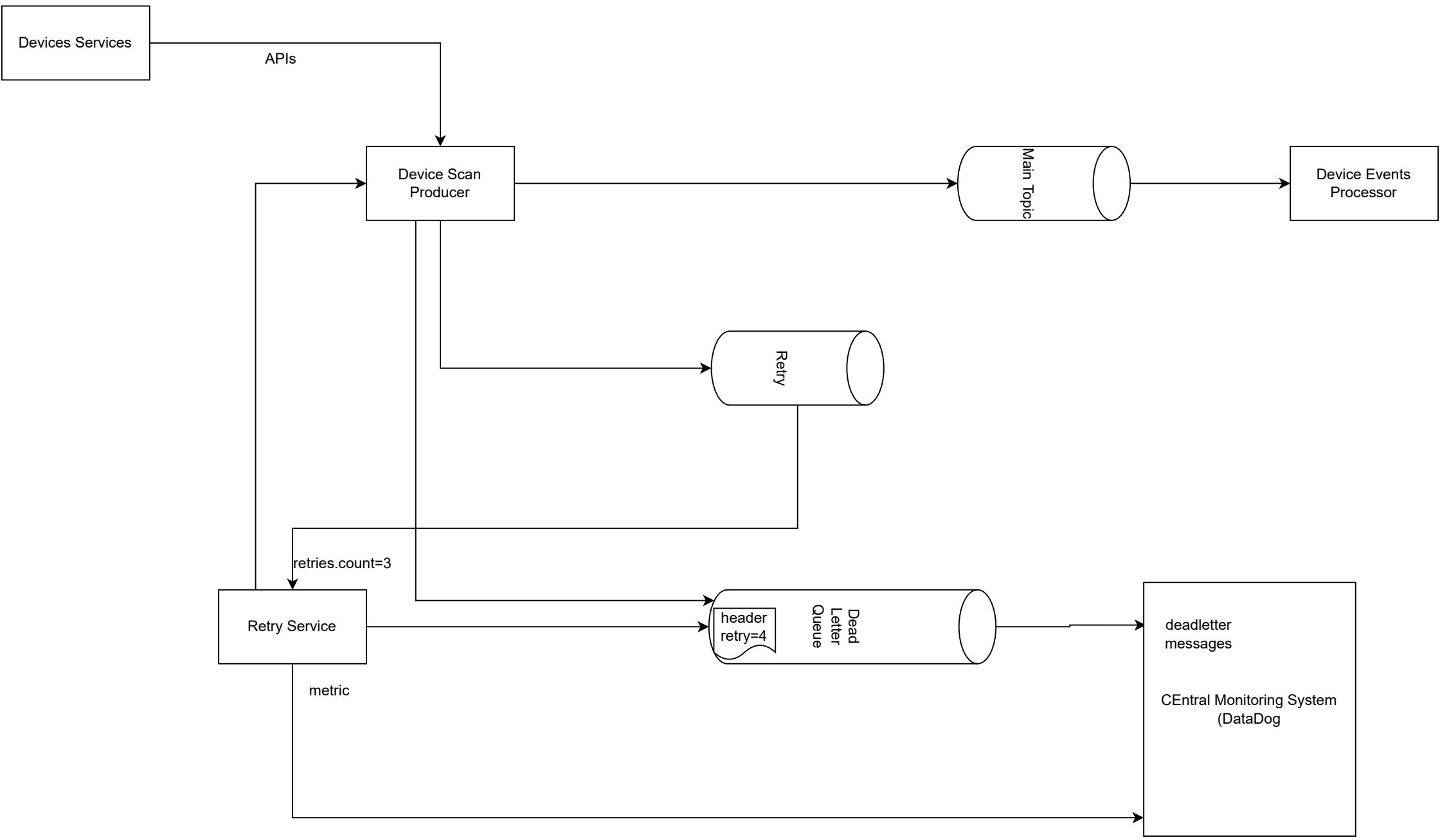
Processors
(Source, Sink, processor)

Streaming
framework

DIY Exercise - day 7

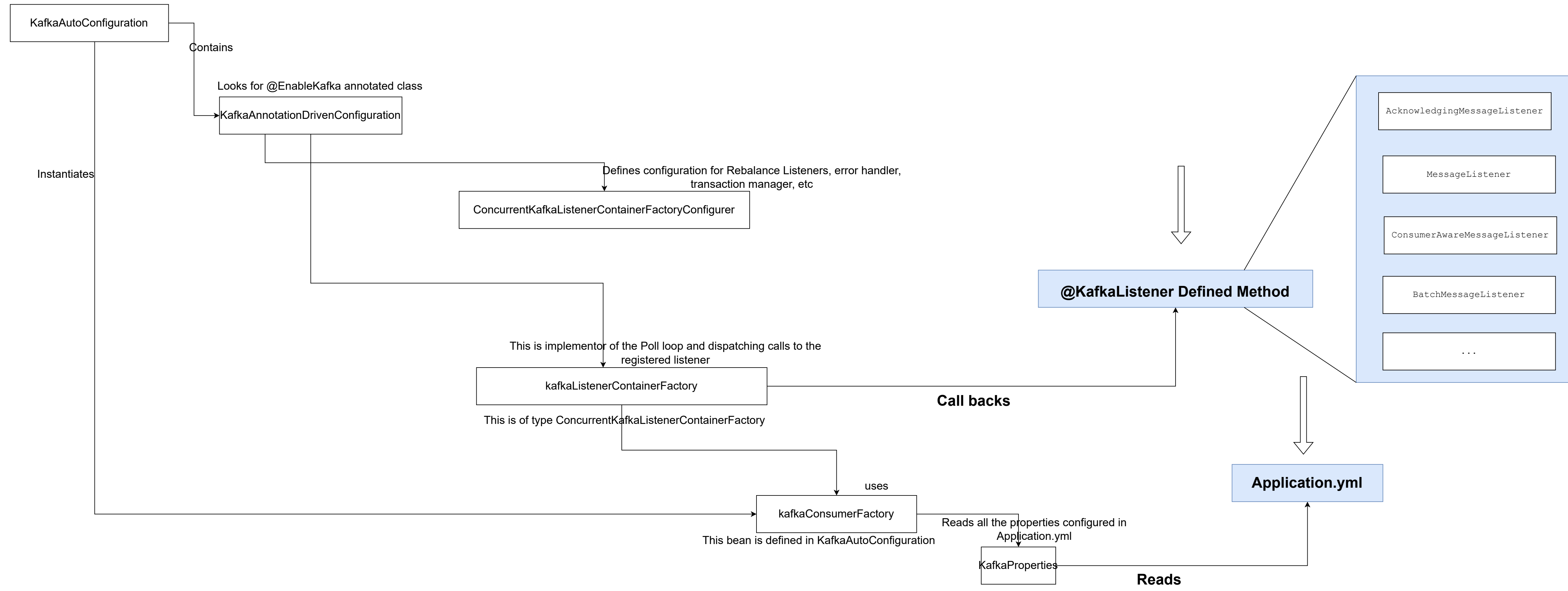


Producer Retry Mechanism

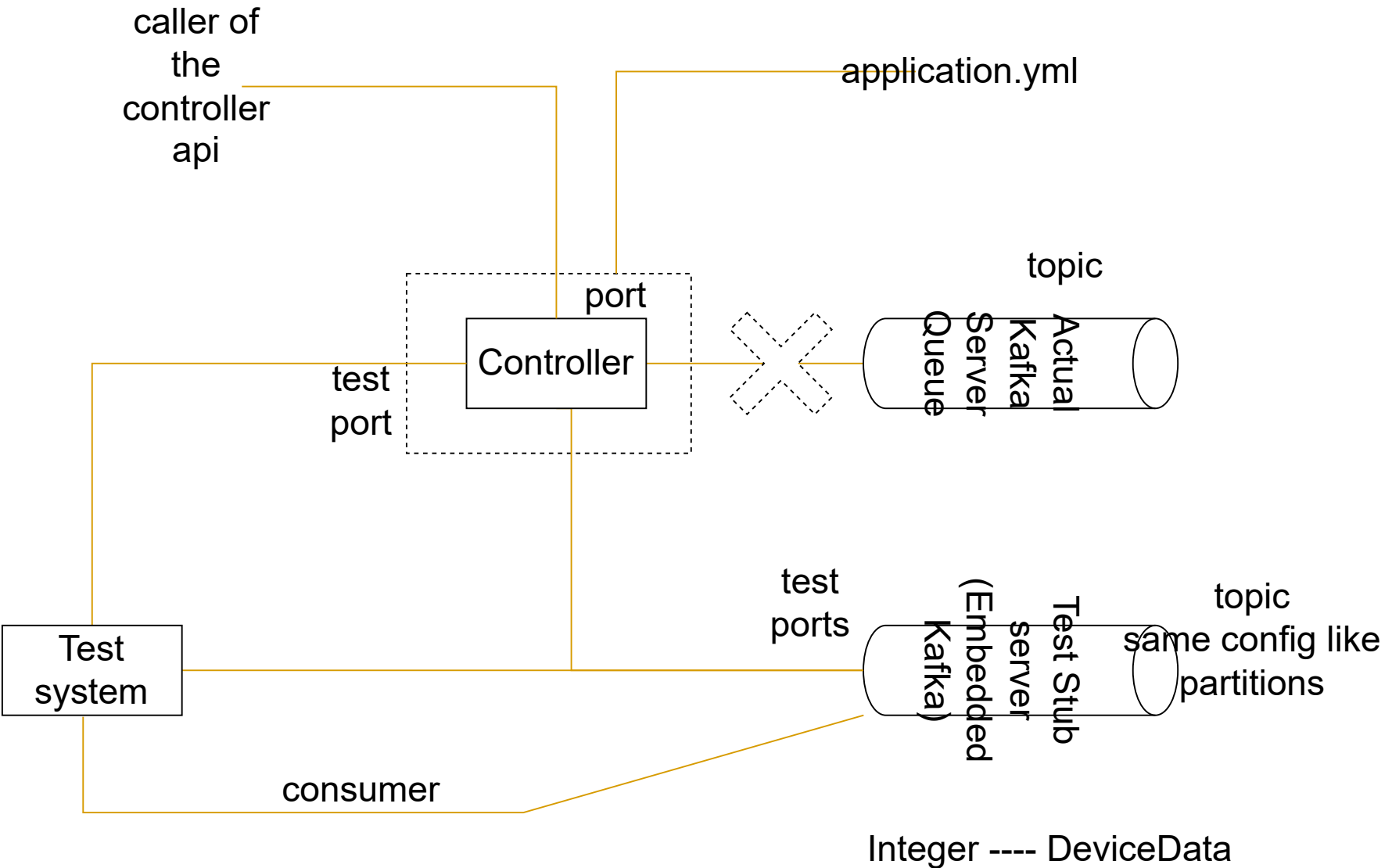


Spring Kafka Consumer Internals

Instantiate needed beans and initiates Other objects

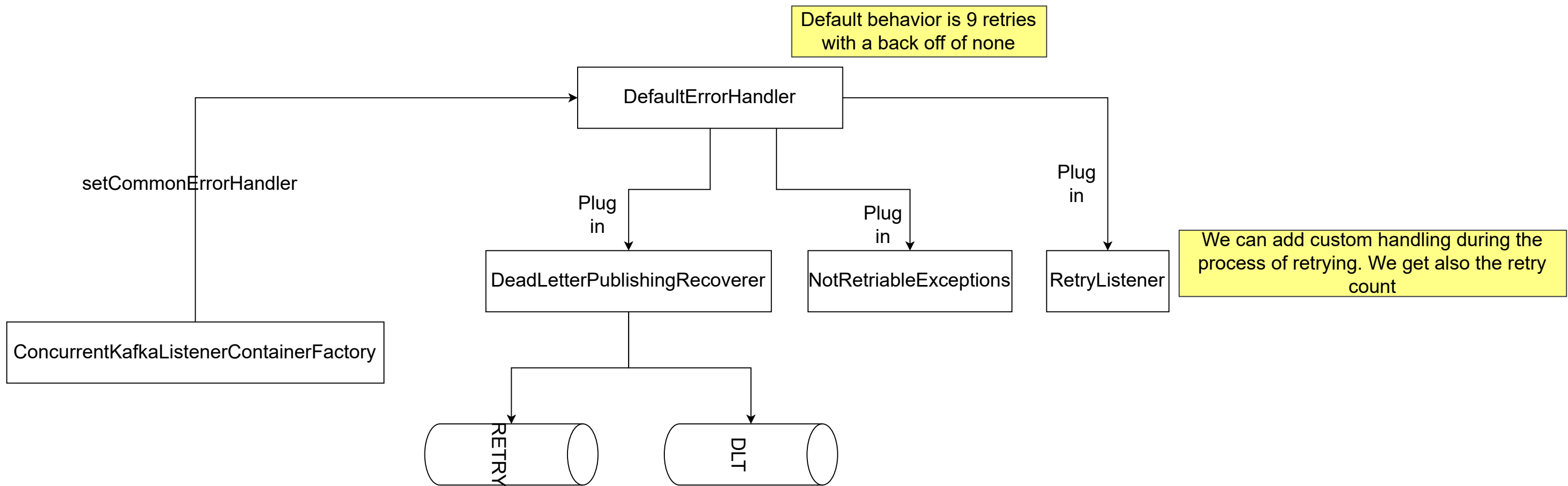


Producer Test

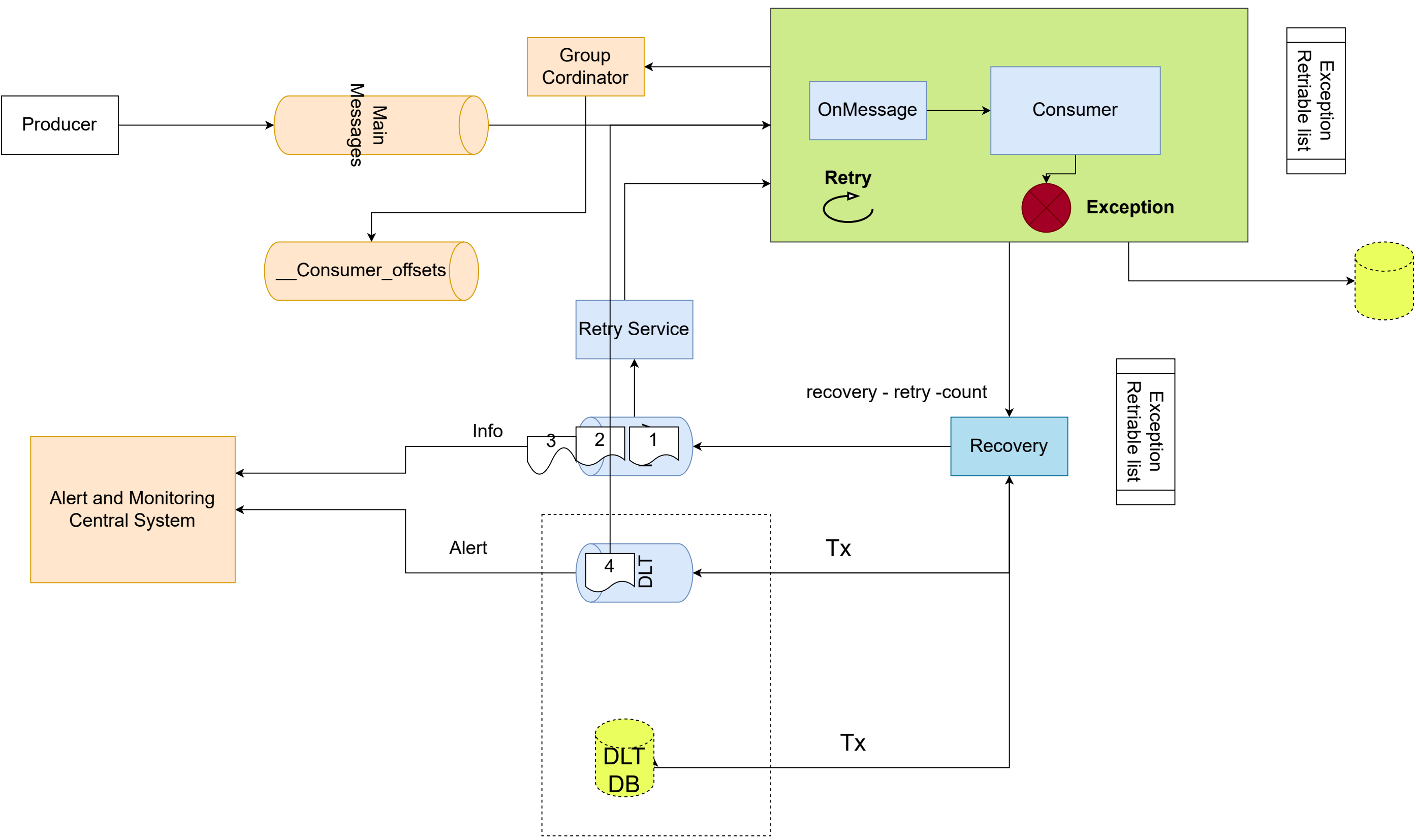


Consumer Error Handling & Recovery Mechanism

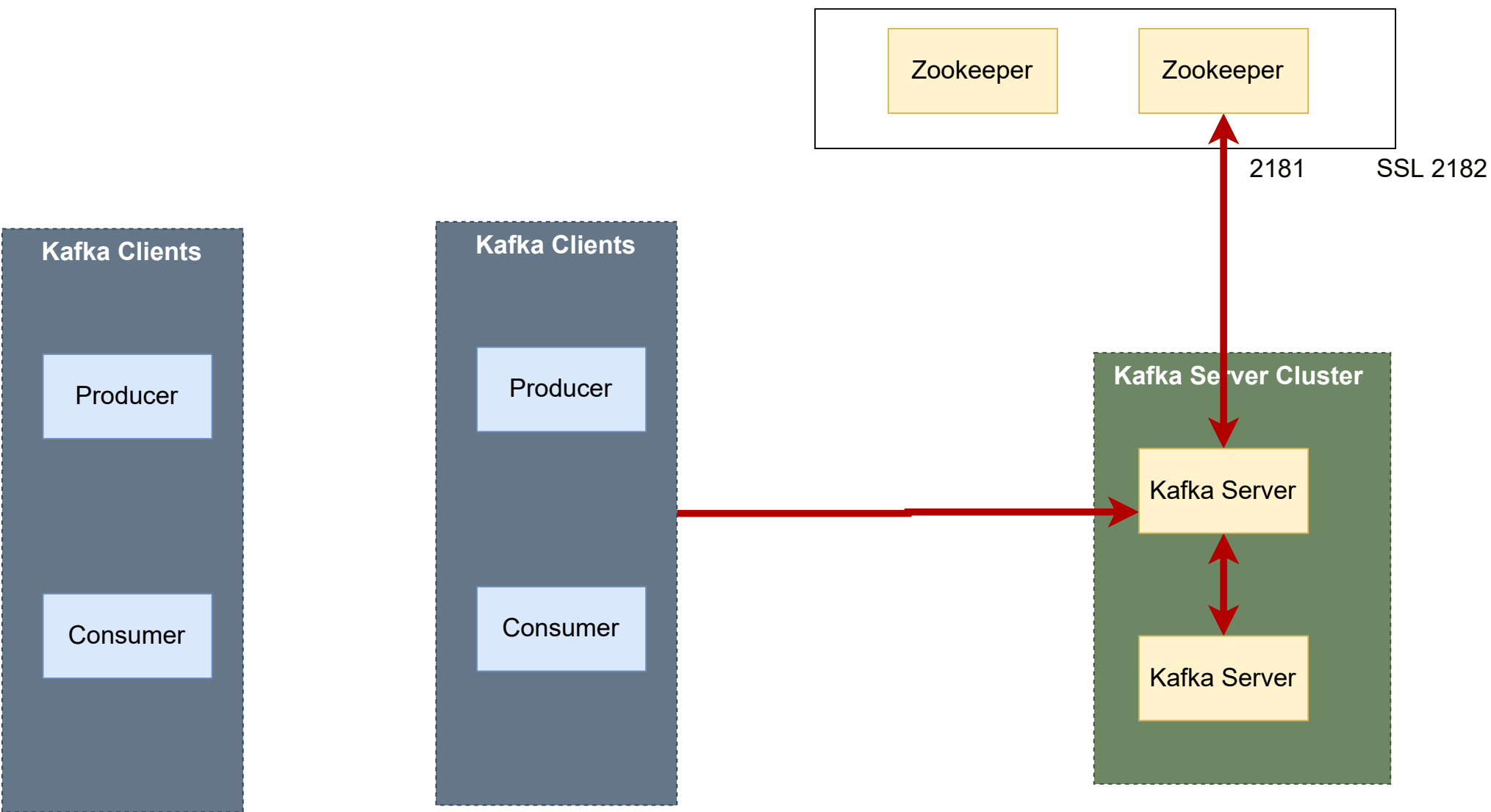
Spring Kafka Consumer Classes



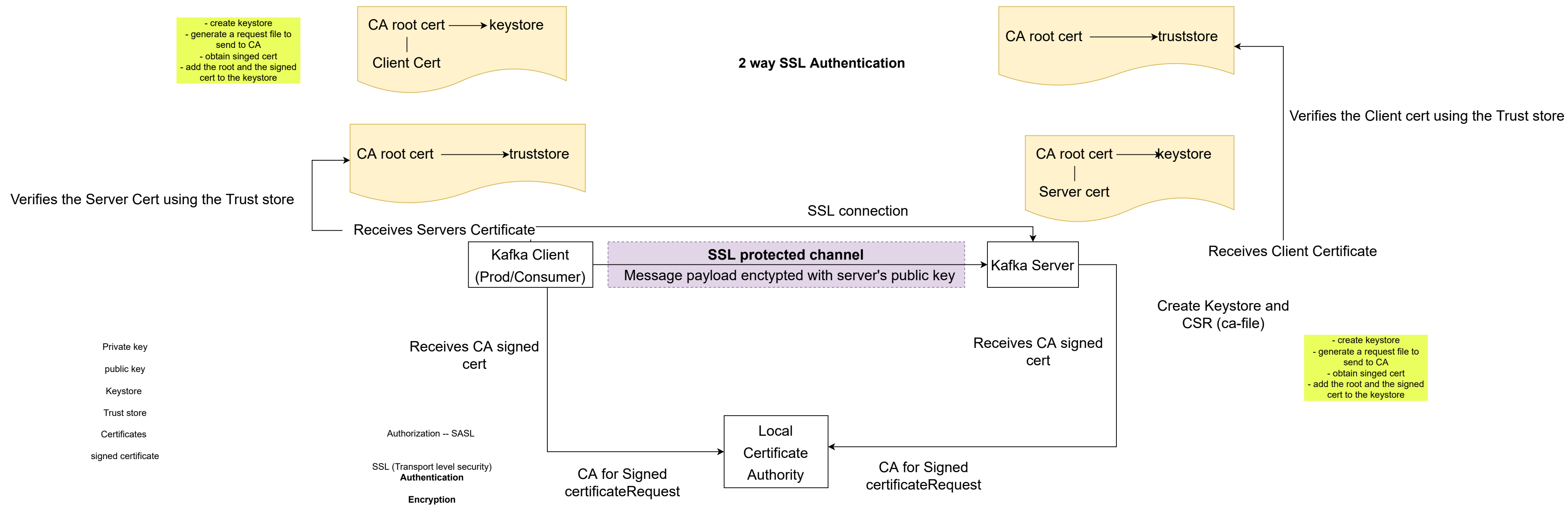
Recovery



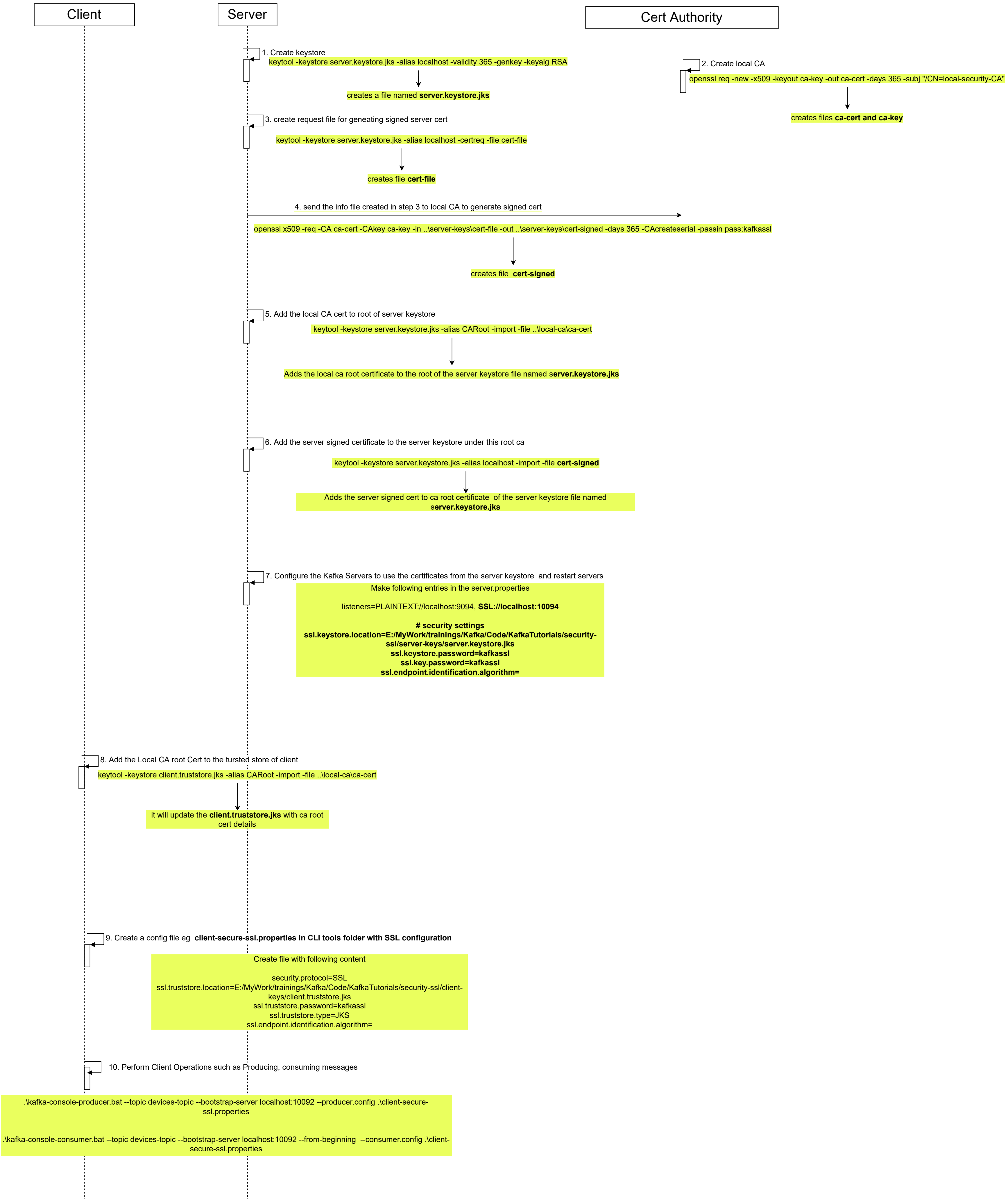
Communications in Kafka Infra To Be Secured



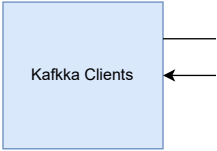
Securing Kafka Client and Server using SSL



SSL Setup for securing Kafka Server and Clients - 1 Way



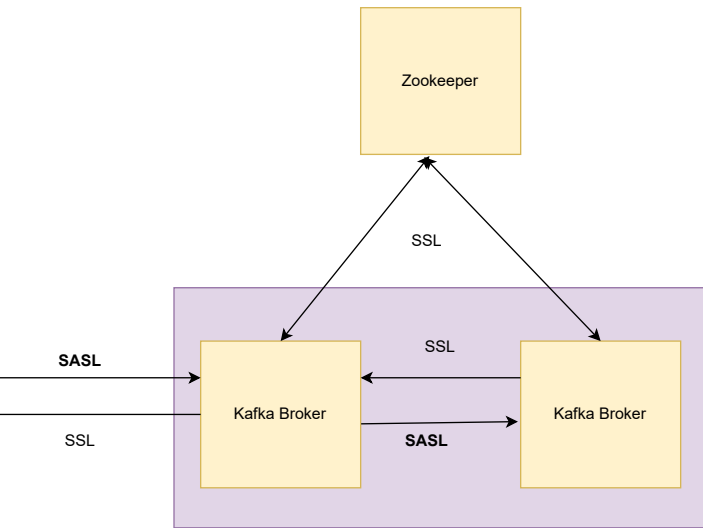
2101 100



User creation

ACL creation

SASL Topology



Important commands

Admin	<code>./bin/kafka-configs.sh --zookeeper localhost:2182 --zk-tls-config-file ./bin/zookeeper-client.properties --alter --add-conf</code>
Producer	<code>./bin/kafka-configs.sh --zookeeper localhost:2182 --zk-tls-config-file ./bin/zookeeper-client.properties --alter --add-conf</code>
Consumer	<code>./bin/kafka-configs.sh --zookeeper localhost:2182 --zk-tls-config-file ./bin/zookeeper-client.properties --alter --add-conf</code>
Producer	<code>./bin/kafka-acls.sh --authorizer-properties zookeeper.connect=localhost:2182 --zk-tls-config-file ./bin/zookeeper-client.properties --add-acl</code>
Consumer	<code>./bin/kafka-acls.sh --authorizer-properties zookeeper.connect=localhost:2182 --zk-tls-config-file ./bin/zookeeper-client.properties --add-acl</code>
Consumer to group	<code>./bin/kafka-acls.sh --authorizer-properties zookeeper.connect=localhost:2182 --zk-tls-config-file ./bin/zookeeper-client.properties --add-acl</code>

fig 'SCRAM-SHA-512=[password=kafkassl]' --entity-type users --entity-name broker-admin

fig 'SCRAM-SHA-512=[password=kafkassl]' --entity-type users --entity-name kafkaprod

fig 'SCRAM-SHA-512=[password=kafkassl]' --entity-type users --entity-name kafkacon

ookeeper-client.properties --add --allow-principal User:kafkaprod --topic secure-topic --operation WRITE --operation DESCRIBE --t

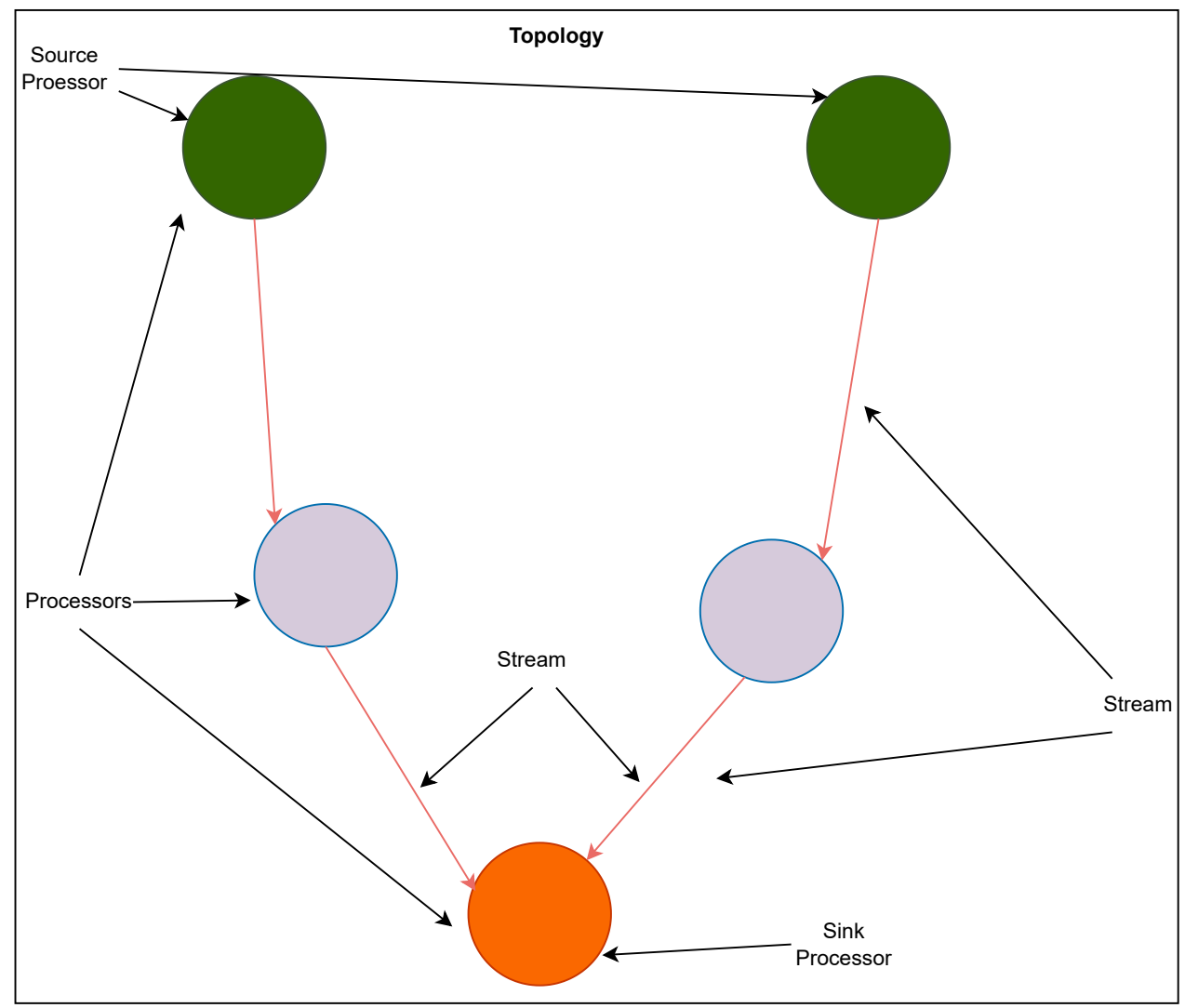
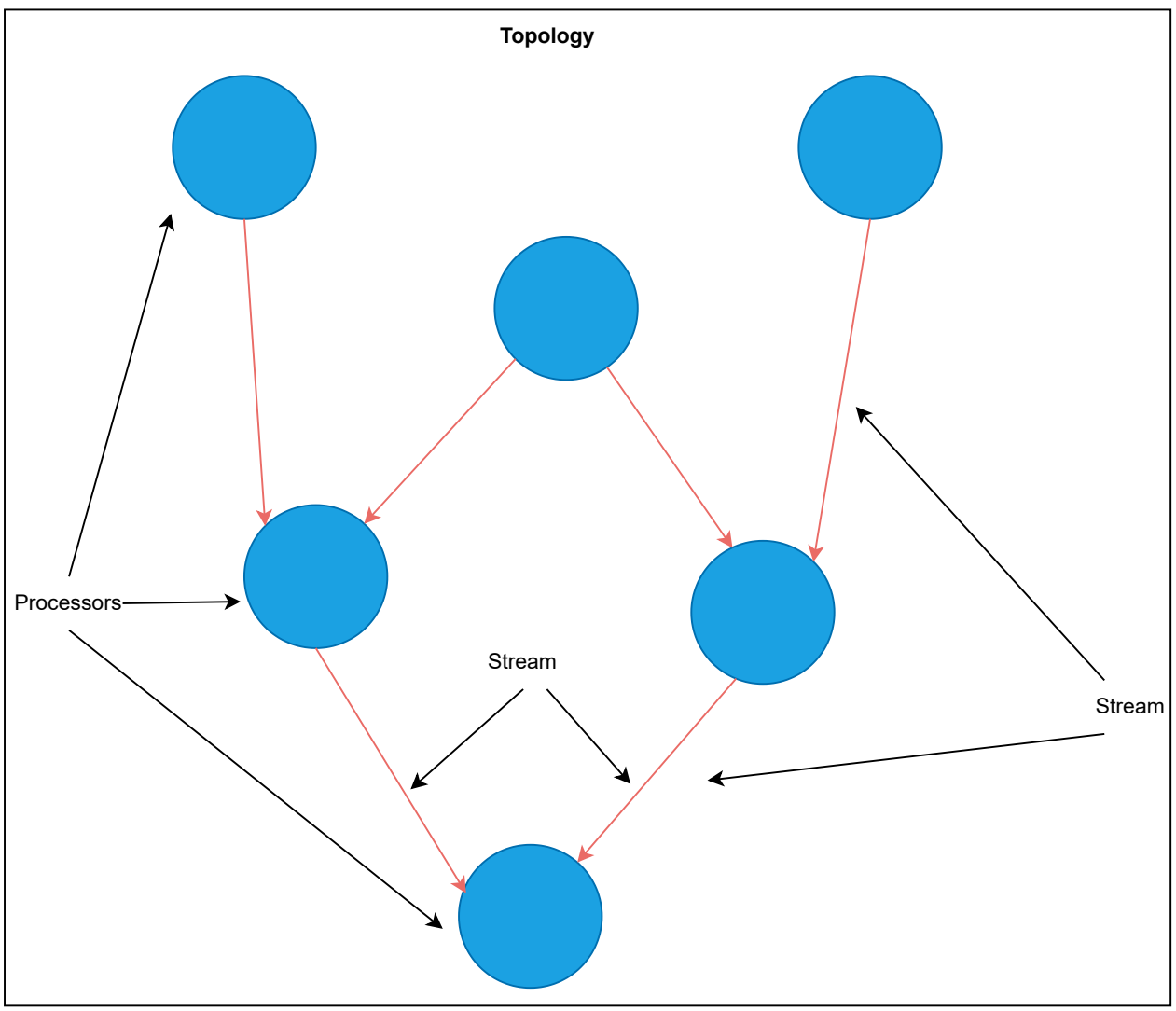
ookeeper-client.properties --add --allow-principal User:kafkacon --topic secure-topic --operation READ --operation DESCRIBE --oper:

ookeeper-client.properties --add --allow-principal User:kafkacon --group secure-group --operation READ

operation DESCRIBECONFIGS

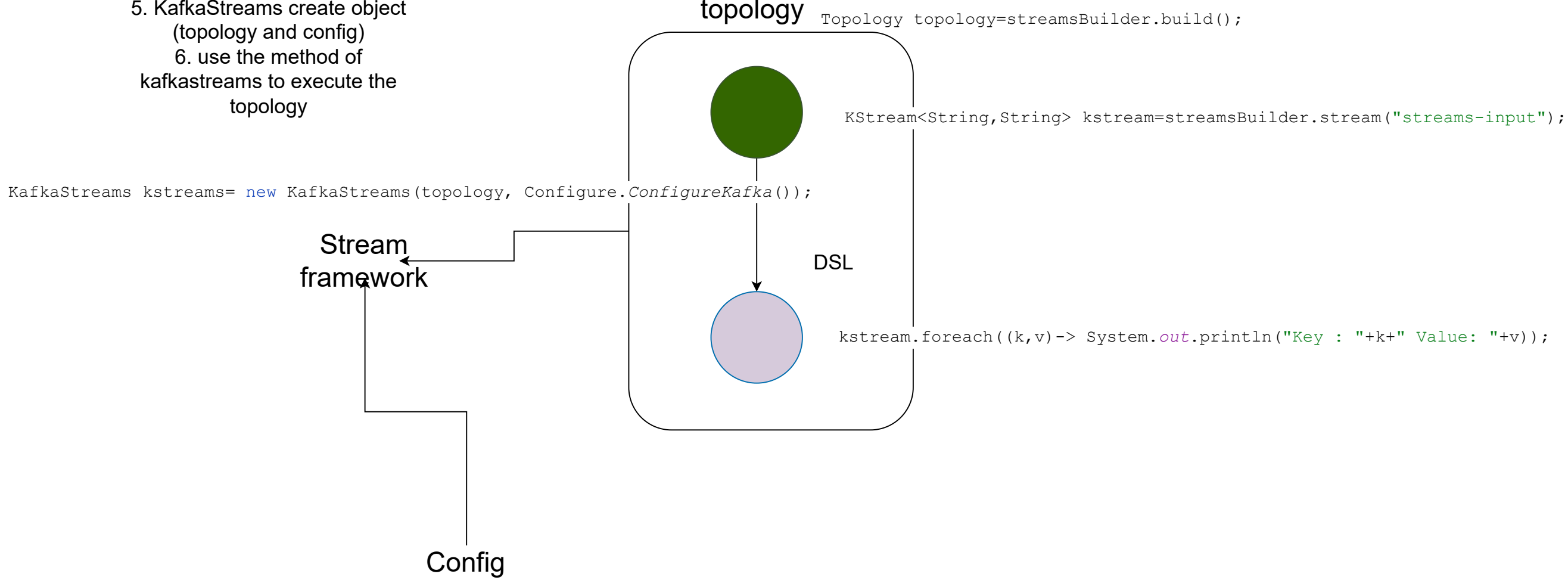
ation DESCRIBECONFIGS

Kafka Streams

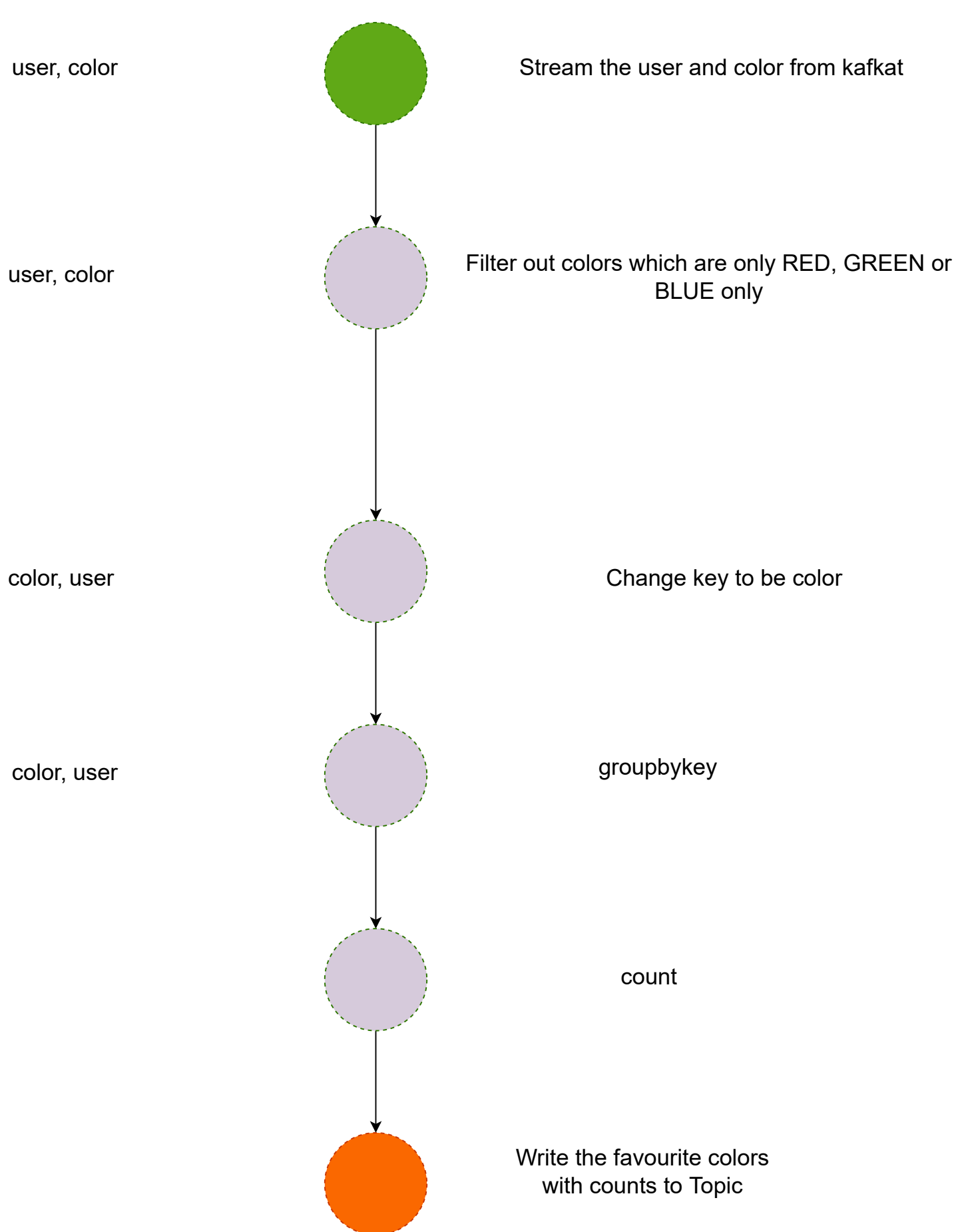
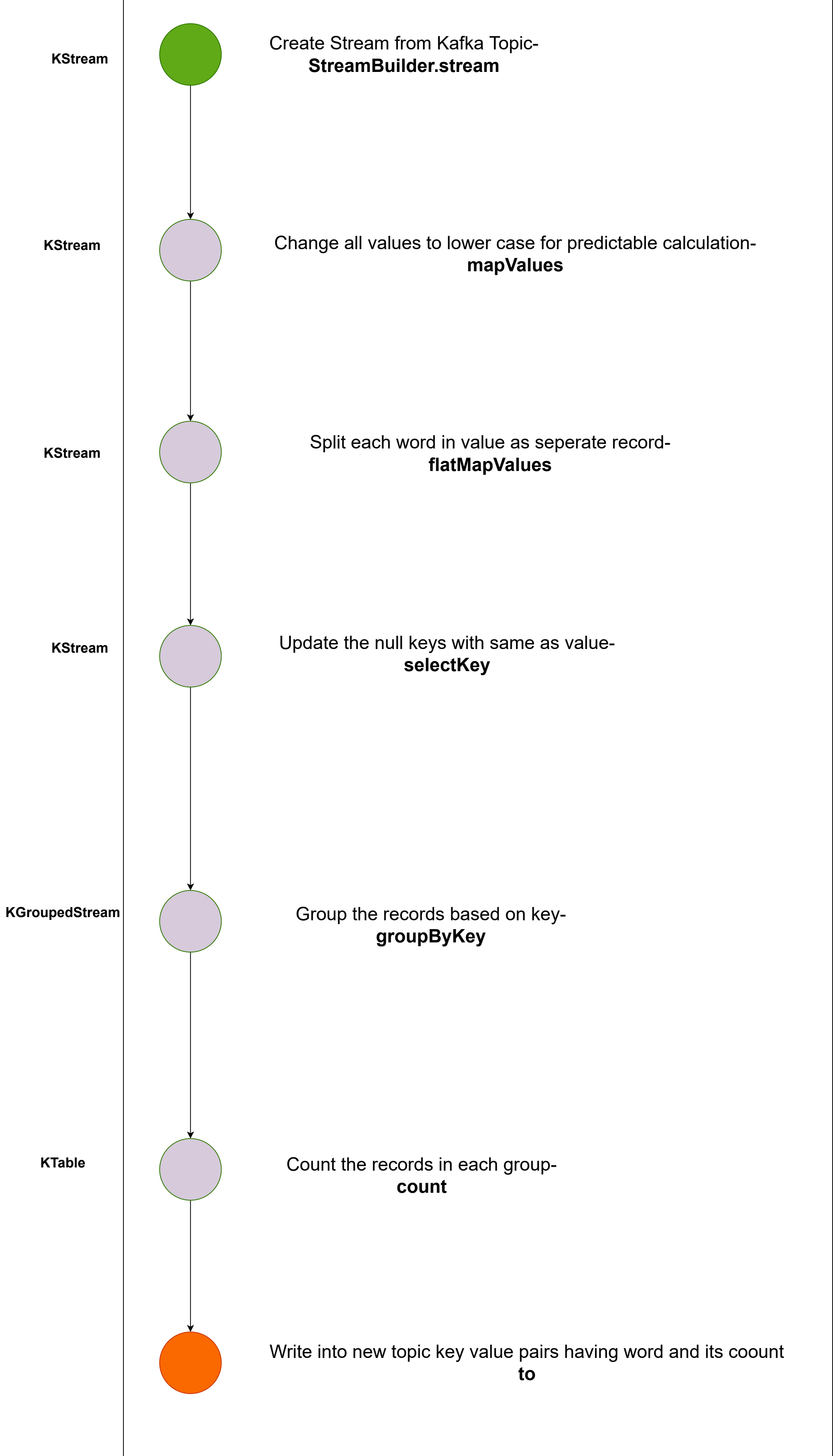
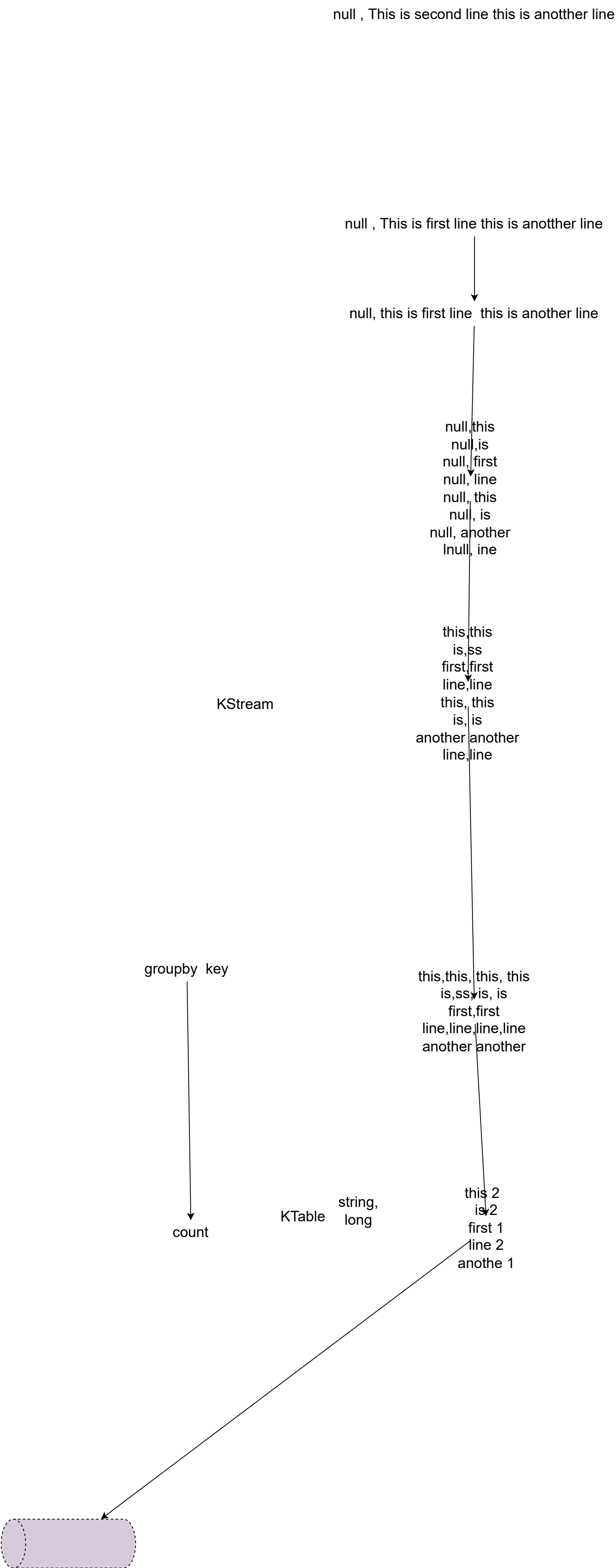


- 1. create StreamBuilder
- 2. Source Stream processor
- 3. Processing processors
- 4. build the builder to create topology
- 5. KafkaStreams create object (topology and config)
- 6. use the method of kafkaStreams to execute the topology

Basic Topology



WordCount Topology



Kafka Streaming Architecture

