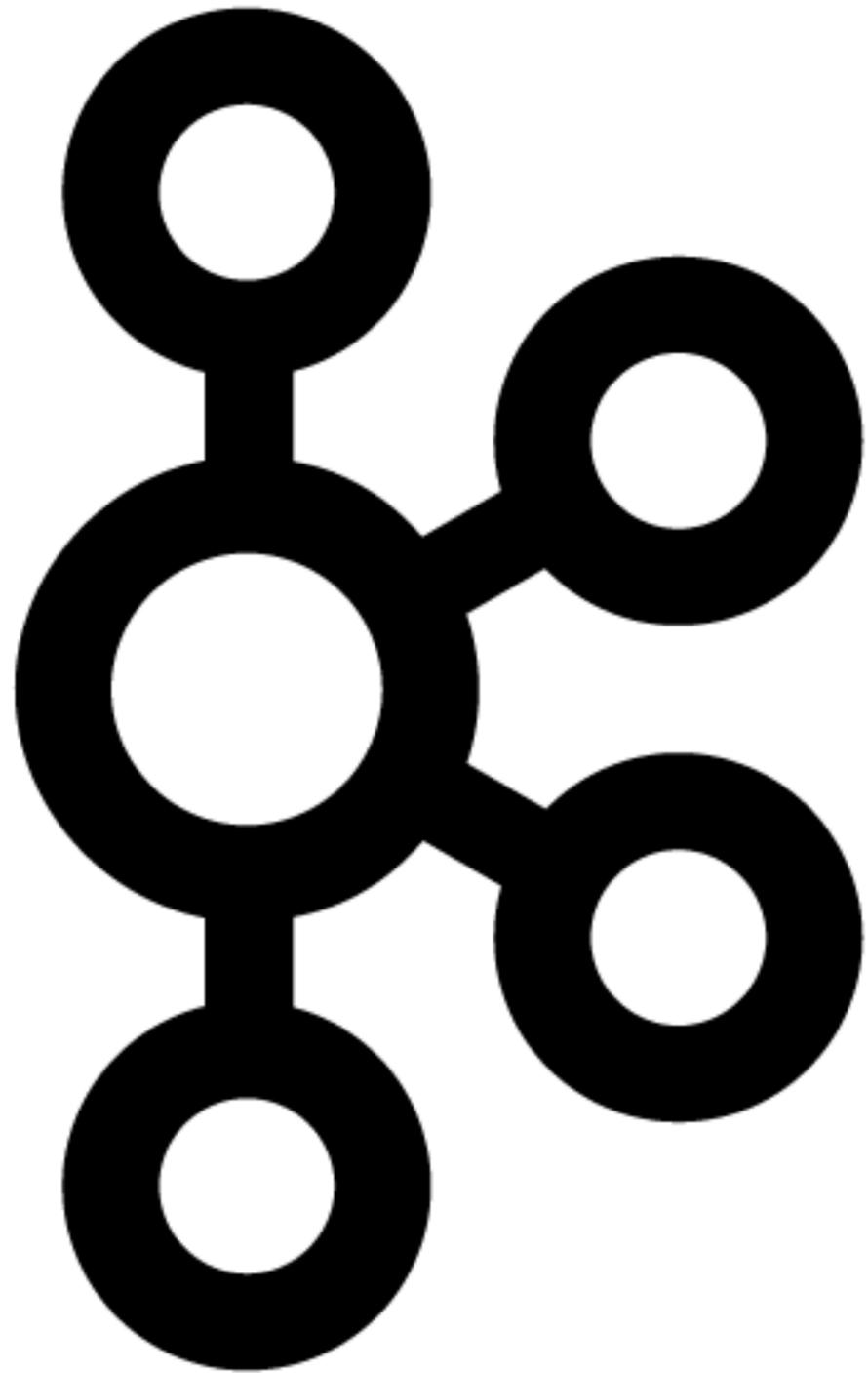


Apache Kafka

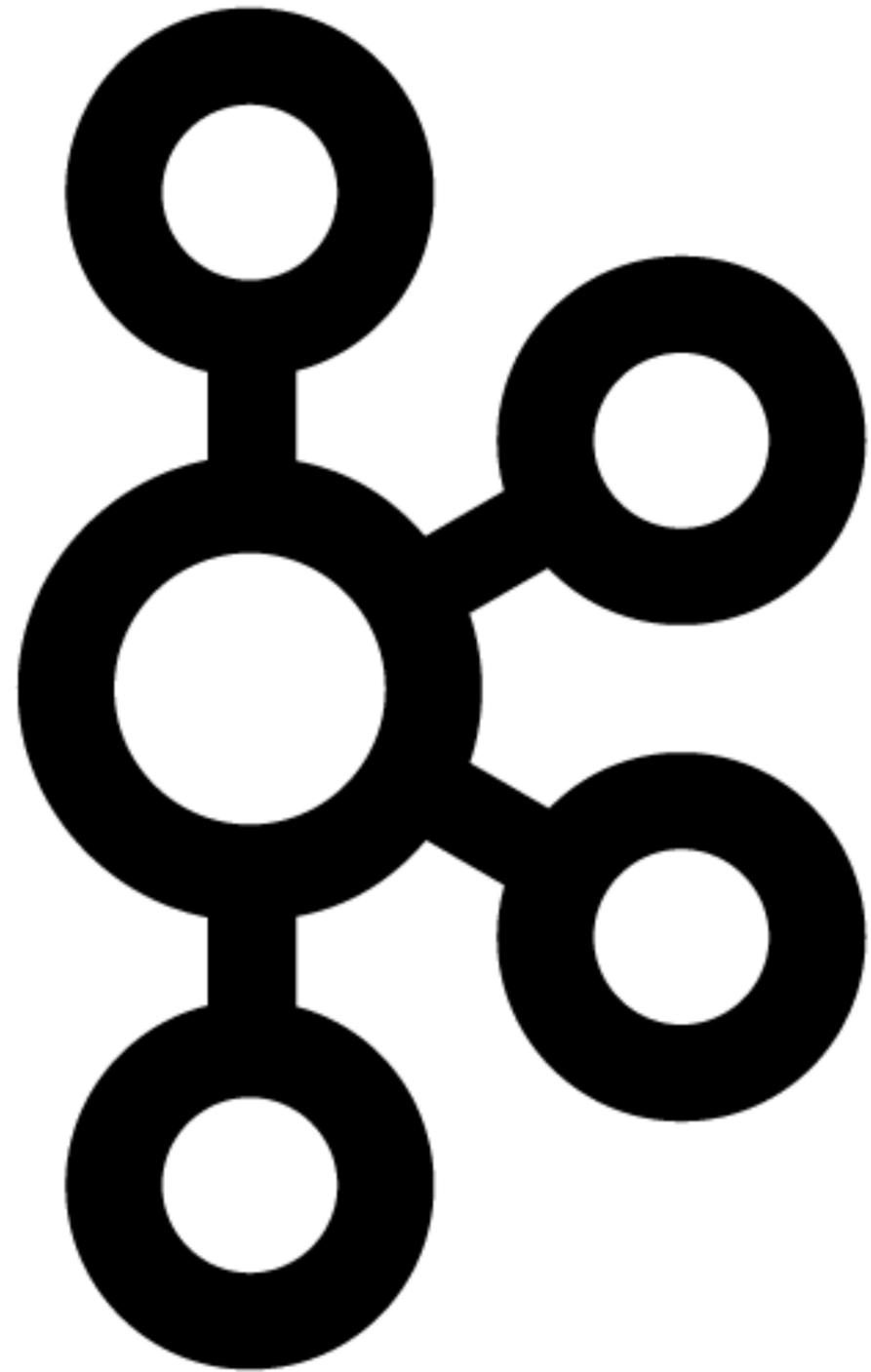
A Streaming Data Platform

and

#javapuzzlersng



APACHE
kafka



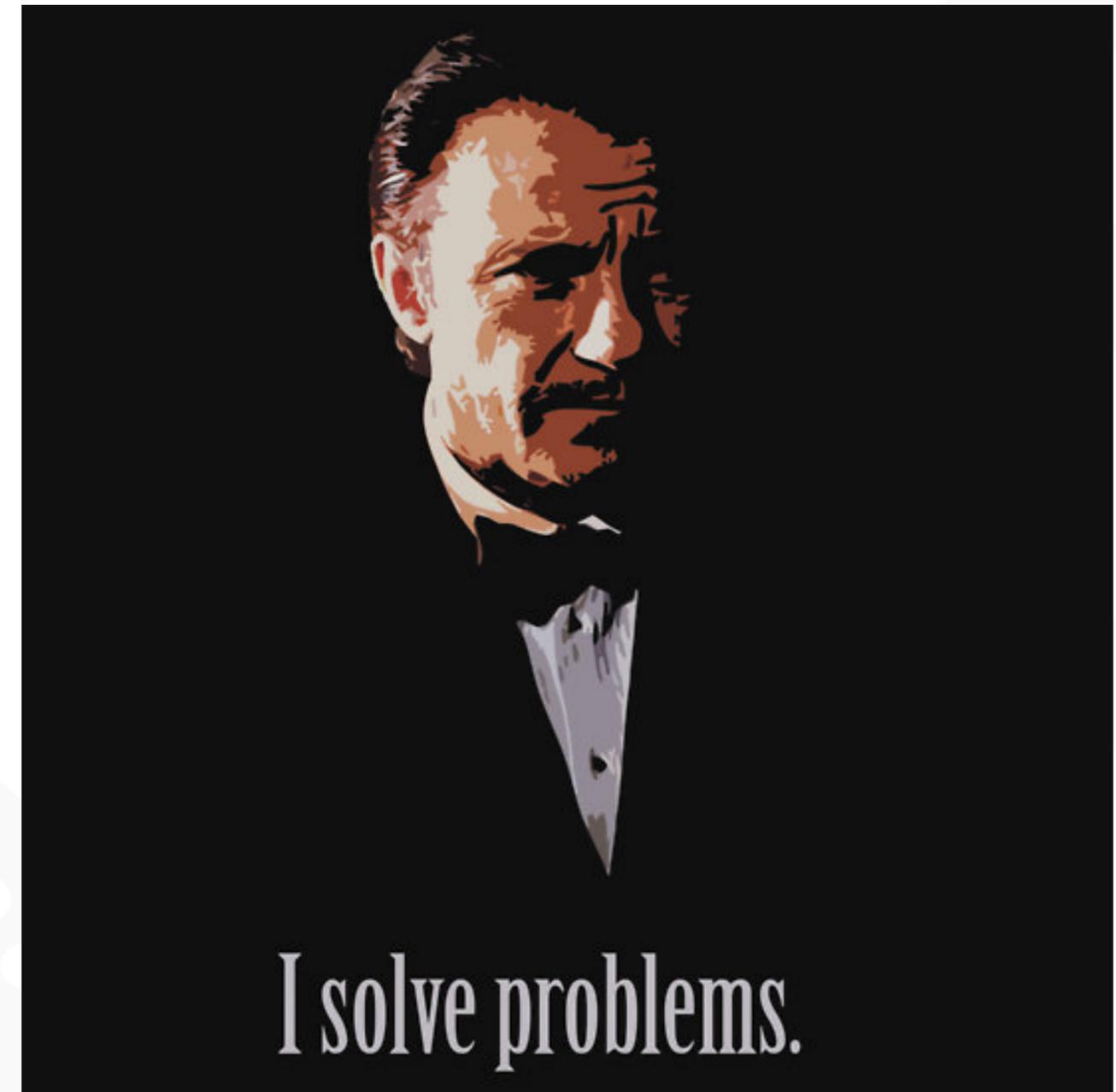
APACHE
kafka



Who am I?



Solutions Architect



@gamussa

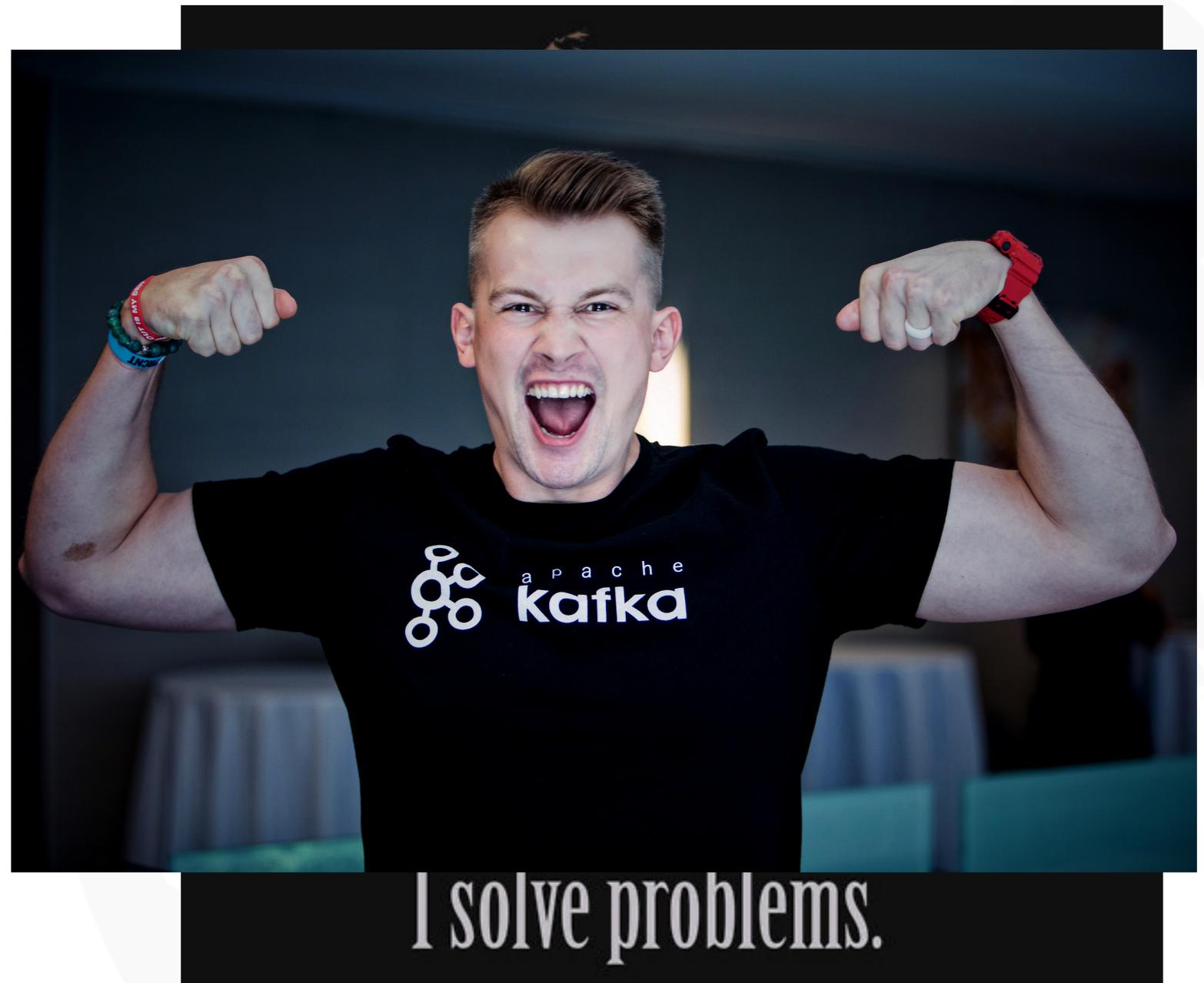
@sfjava

@confluentinc

Who am I?



Solutions Architect
Developer Advocate



@gamussa

@sfjava

@confluentinc

Who am I?



Solutions Architect
Developer Advocate

@gamussa in internetz



@gamussa

@sfjava

@confluentinc

Who am I?



Solutions Architect
Developer Advocate

@gamussa in internetz

Hey you, yes, you,
go follow me in twitter ©



@gamussa

@sfjava

@confluentinc

Kafka & Confluent



We are hiring!



<https://www.confluent.io/careers/>



@gamussa

@sfjava

@confluentinc



@gamussa

@sfjava

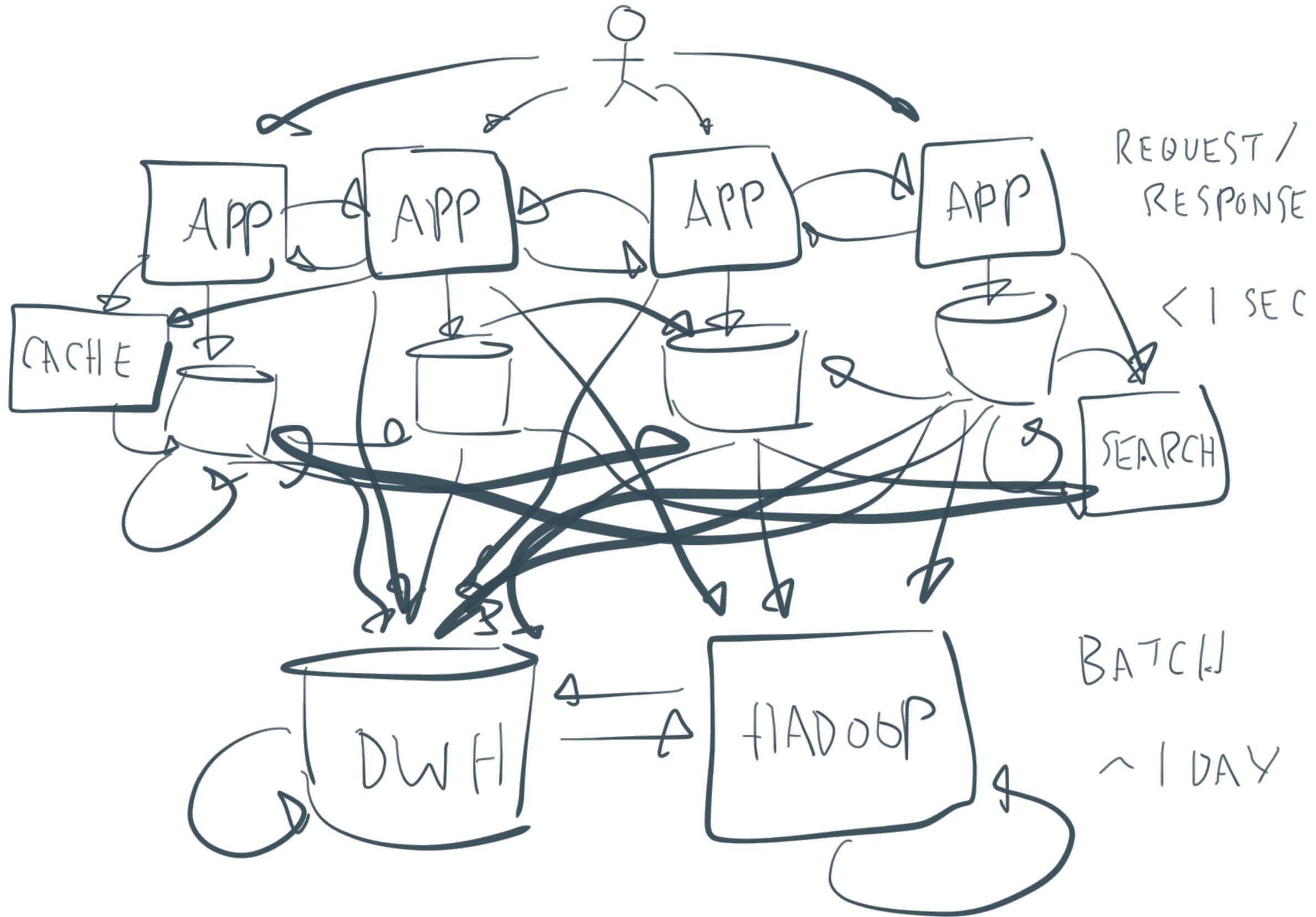
@confluentinc

A company is build on

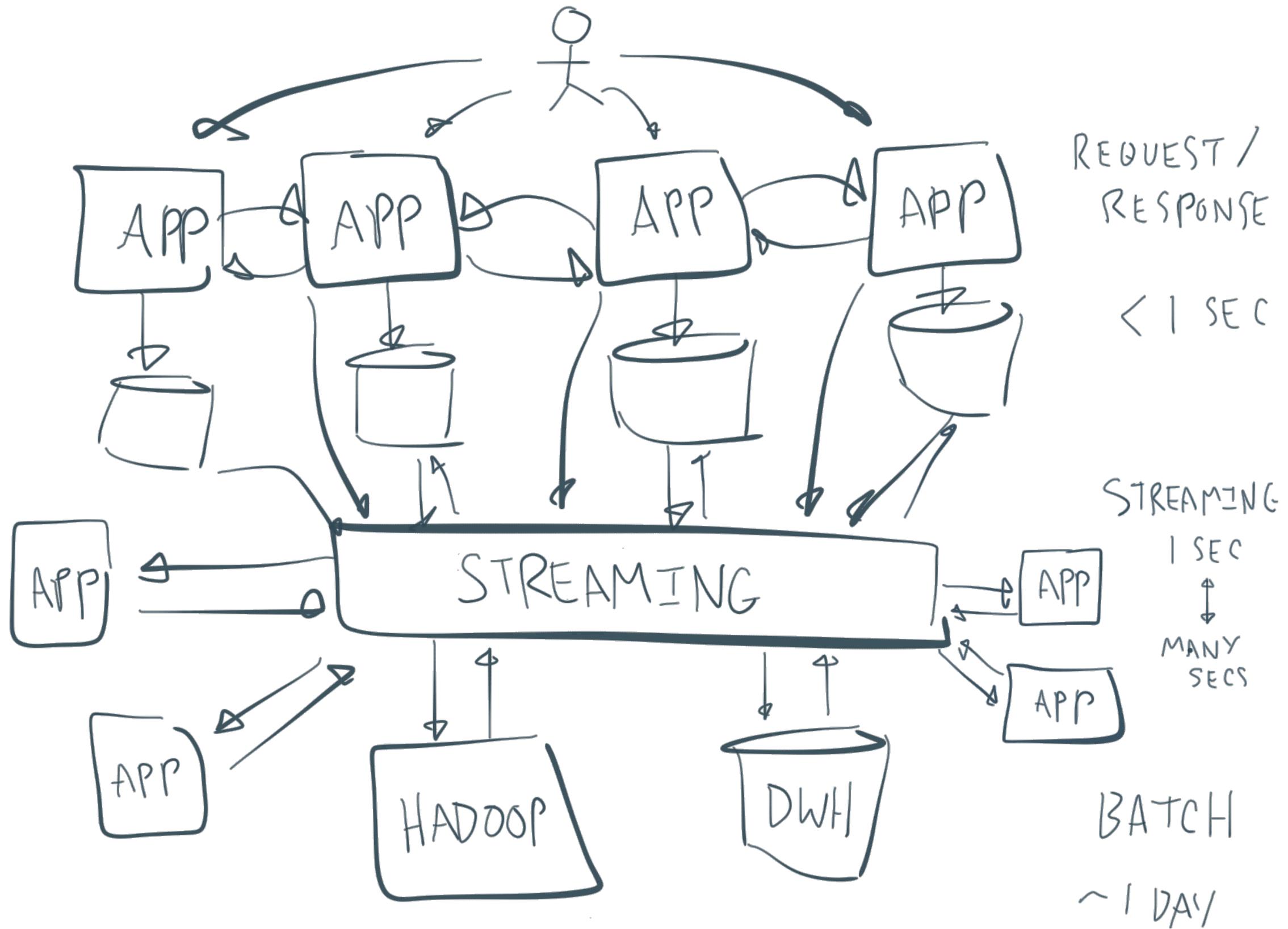
A company is build on
DATA FLOWS

but

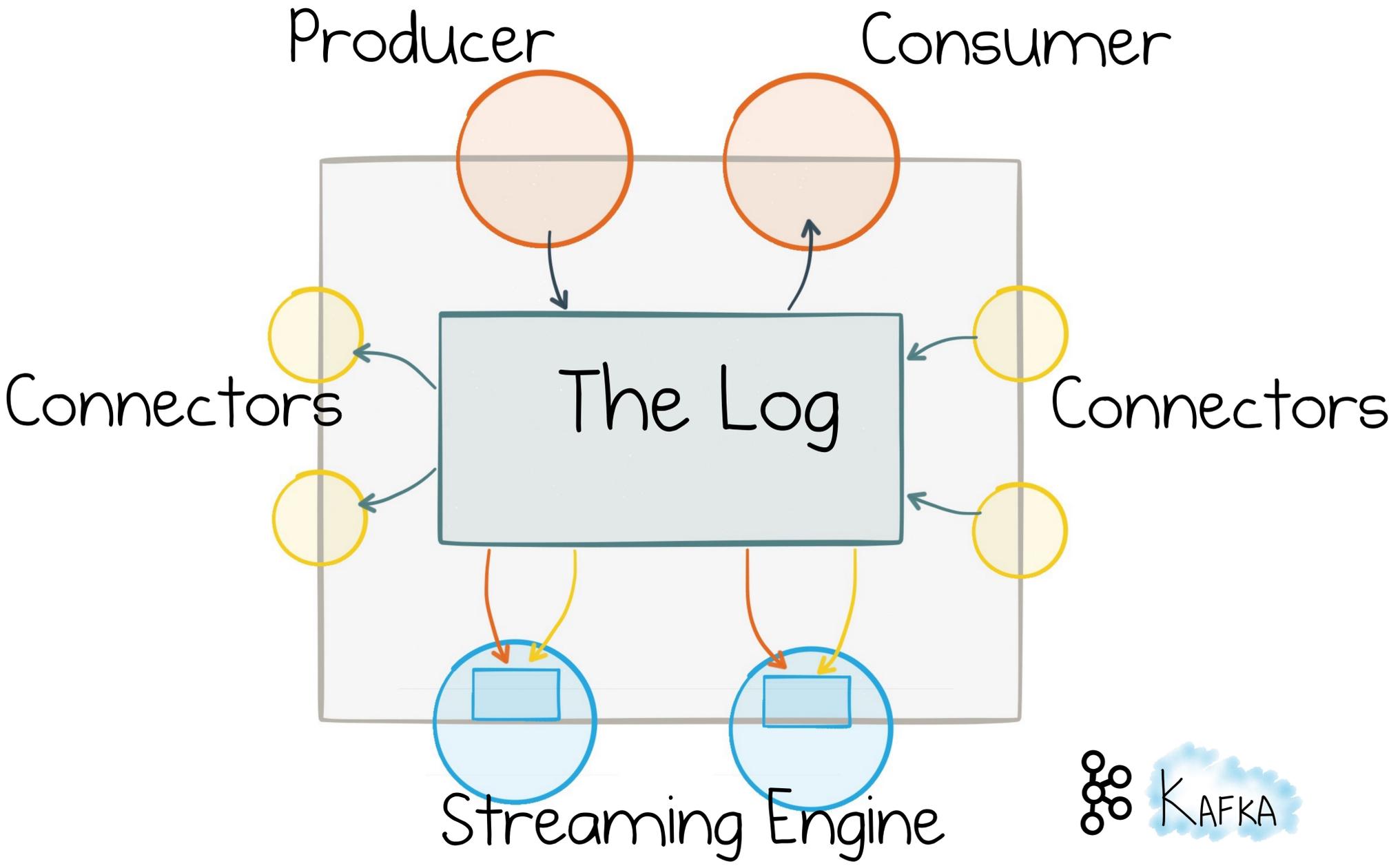
All we have is
DATA STORES



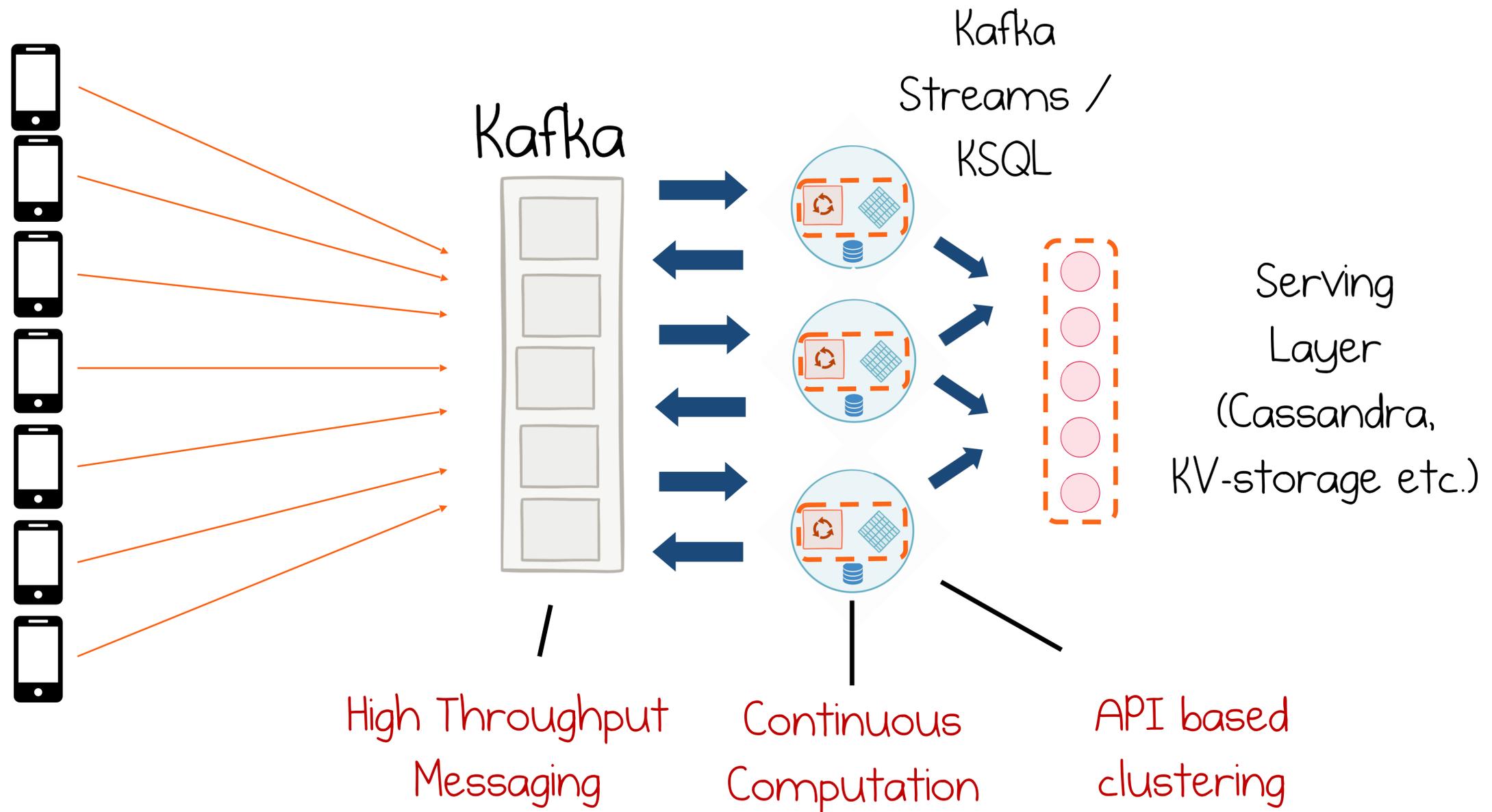




Kafka is a Streaming Platform

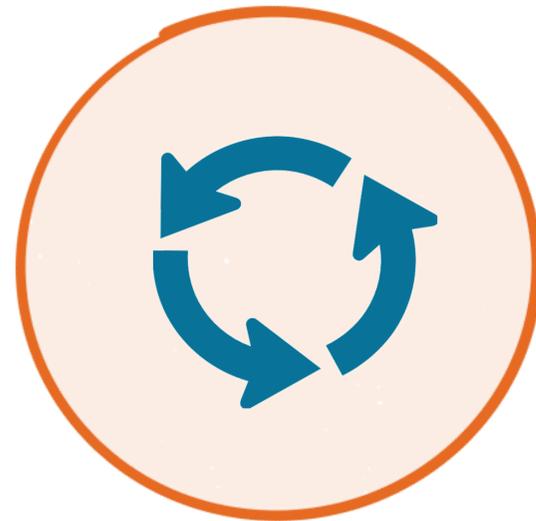
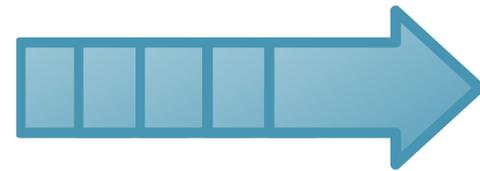


Origins in Stream Processing

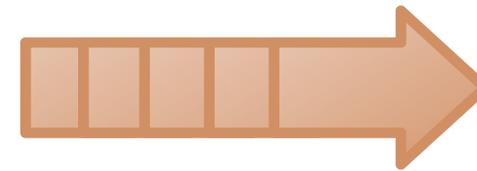


What exactly is Stream Processing?

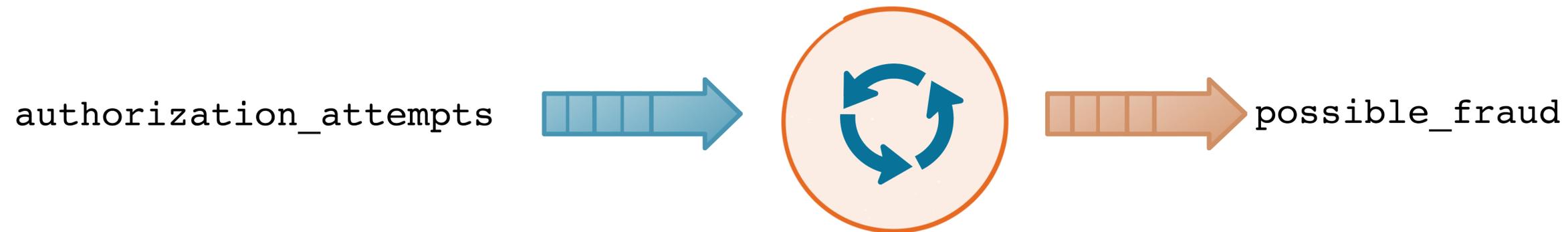
authorization_attempts



possible_fraud

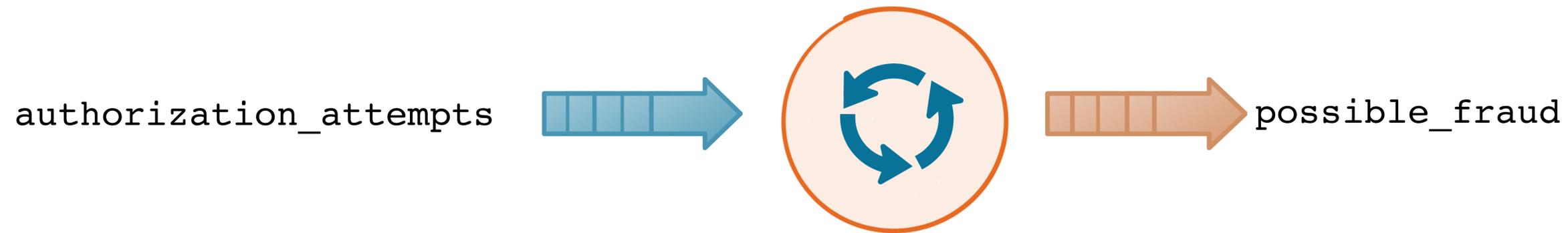


What exactly is Stream Processing?



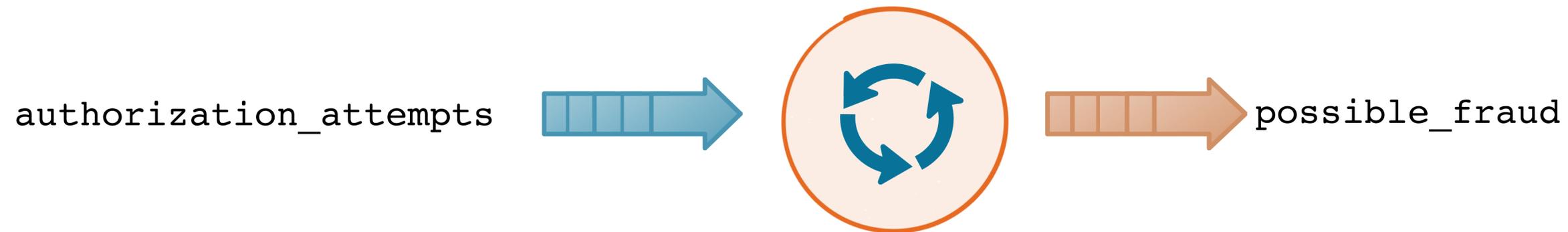
```
CREATE STREAM possible_fraud AS
SELECT card_number, count(*)
FROM authorization_attempts
WINDOW TUMBLING (SIZE 5 MINUTE)
GROUP BY card_number
HAVING count(*) > 3;
```

What exactly is Stream Processing?



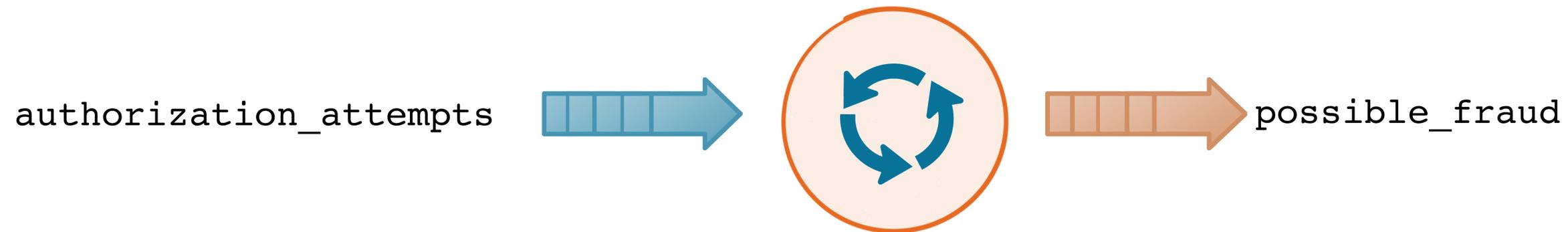
```
CREATE STREAM possible_fraud AS
SELECT card_number, count(*)
FROM authorization_attempts
WINDOW TUMBLING (SIZE 5 MINUTE)
GROUP BY card_number
HAVING count(*) > 3;
```

What exactly is Stream Processing?



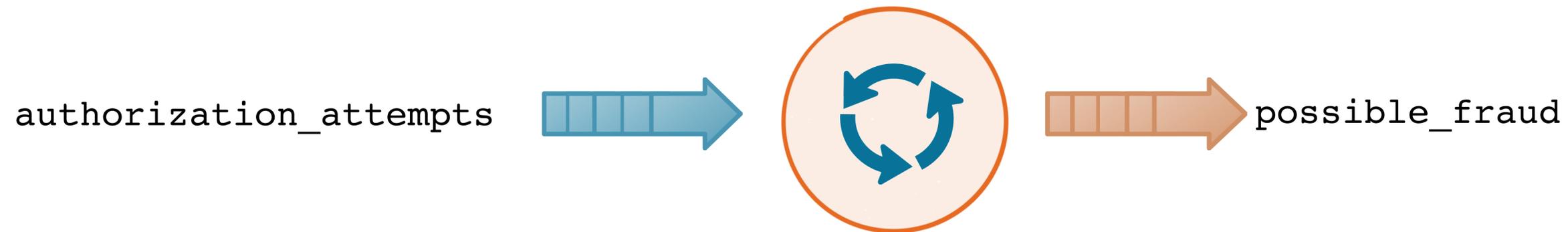
```
CREATE STREAM possible_fraud AS
SELECT card_number, count(*)
FROM authorization_attempts
WINDOW TUMBLING (SIZE 5 MINUTE)
GROUP BY card_number
HAVING count(*) > 3;
```

What exactly is Stream Processing?



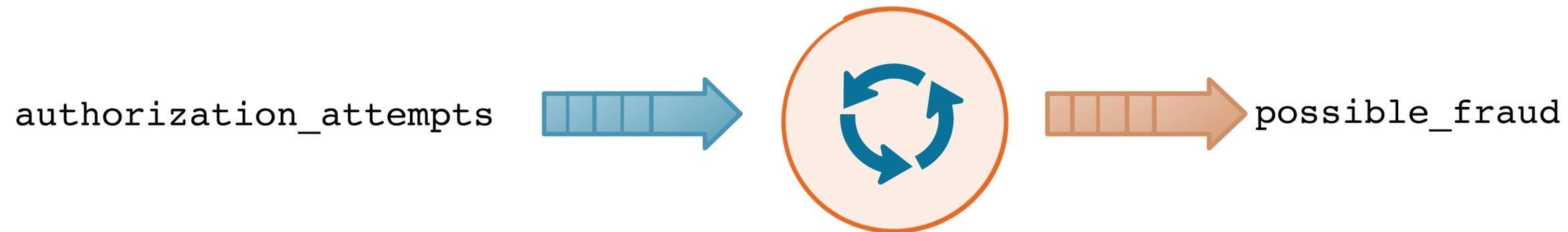
```
CREATE STREAM possible_fraud AS
SELECT card_number, count(*)
FROM authorization_attempts
WINDOW TUMBLING (SIZE 5 MINUTE)
GROUP BY card_number
HAVING count(*) > 3;
```

What exactly is Stream Processing?



```
CREATE STREAM possible_fraud AS
SELECT card_number, count(*)
FROM authorization_attempts
WINDOW TUMBLING (SIZE 5 MINUTE)
GROUP BY card_number
HAVING count(*) > 3;
```

What exactly is Stream Processing?

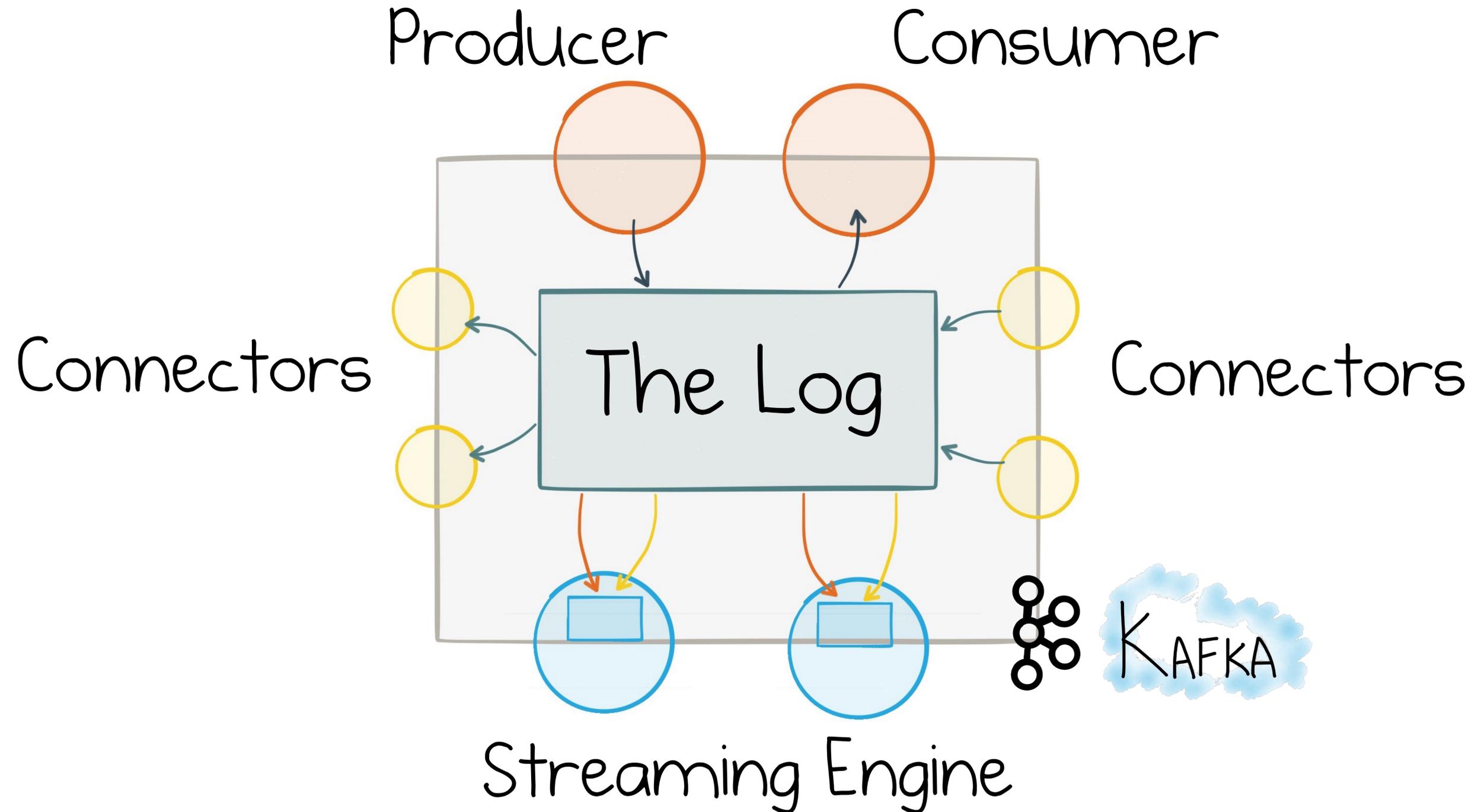


```
CREATE STREAM possible_fraud AS
SELECT card_number, count(*)
FROM authorization_attempts
WINDOW TUMBLING (SIZE 5 MINUTE)
GROUP BY card_number
HAVING count(*) > 3;
```

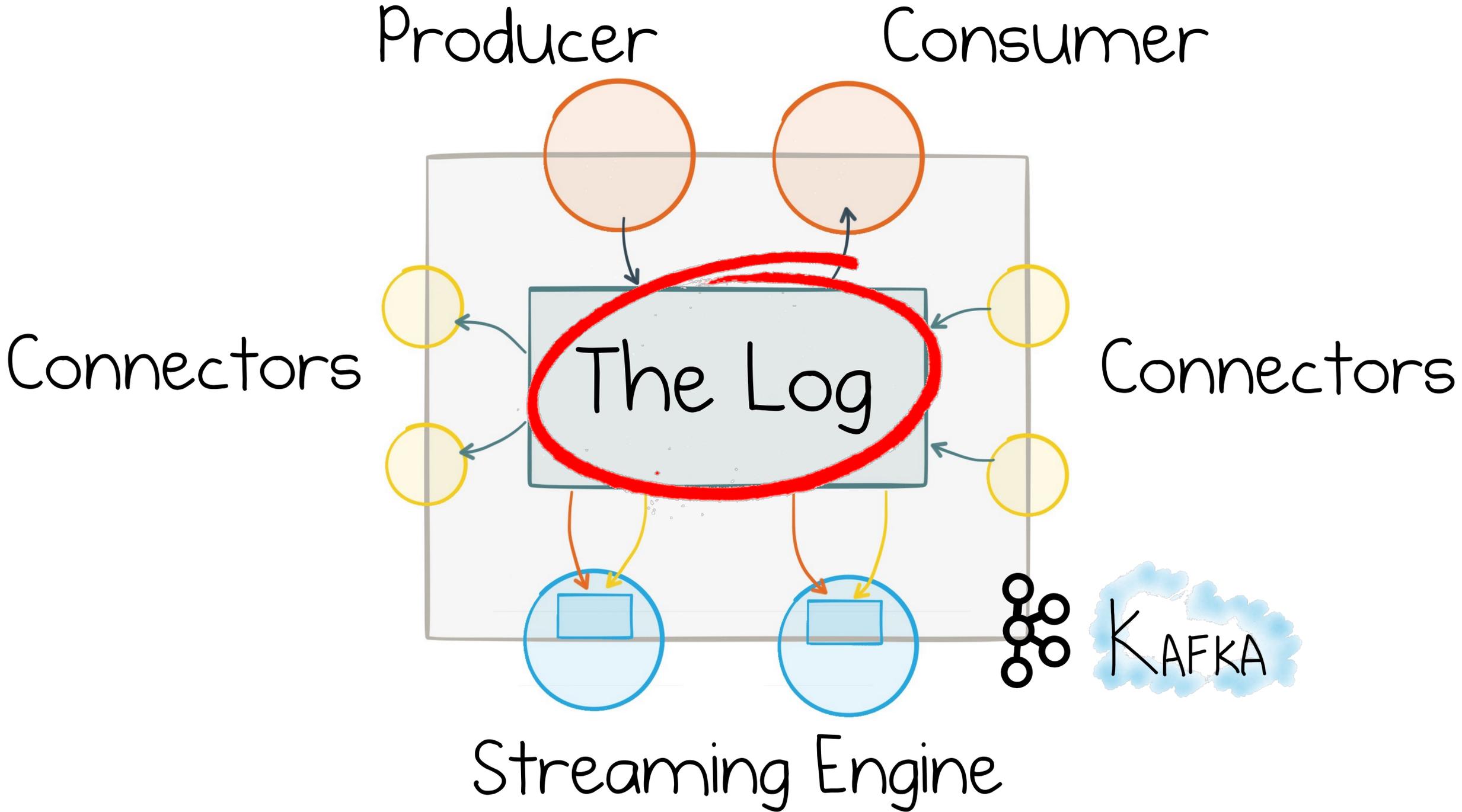
Streaming
is the toolset for dealing
with events
as they move!



What is a Streaming Platform?



Kafka's Distributed Log



The log is a type of durable messaging system

The log is a type of durable messaging system

Similar to a traditional messaging system (ActiveMQ, Rabbit etc)
but with:

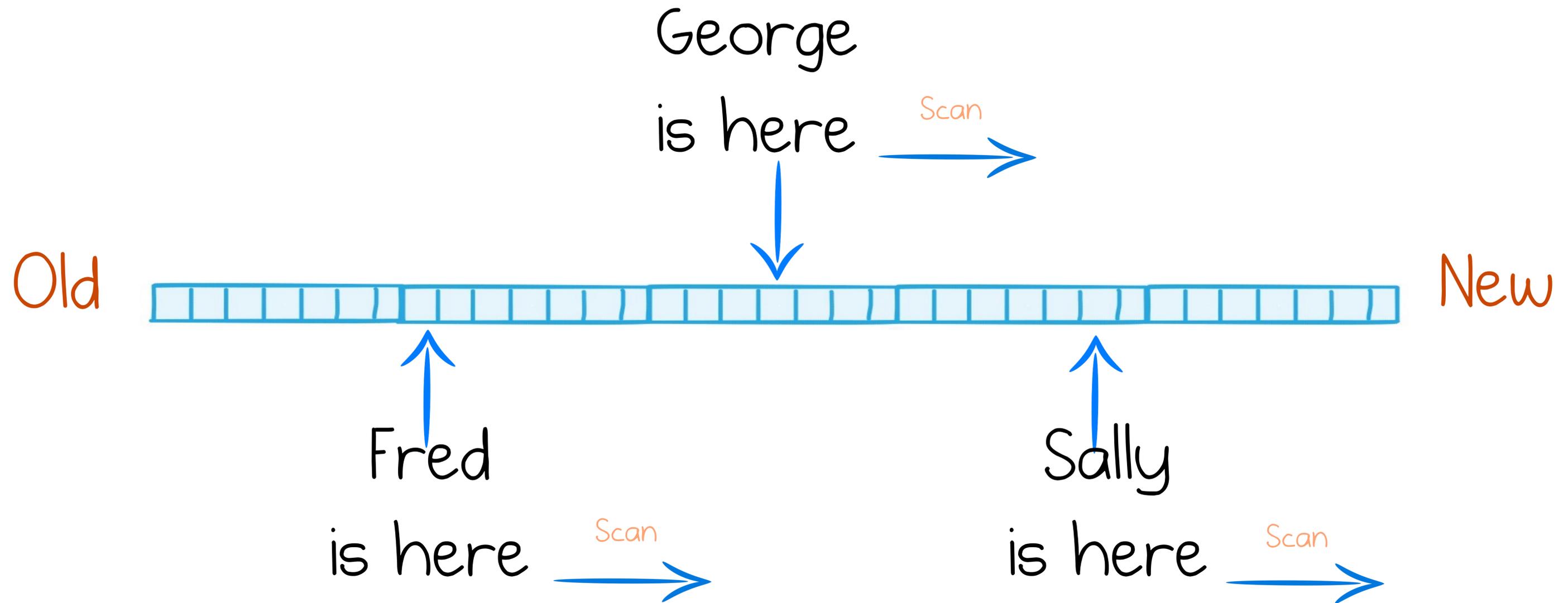
- (a) Far better scalability
- (b) Built in fault tolerance / HA
- (c) Storage

The log is a simple idea

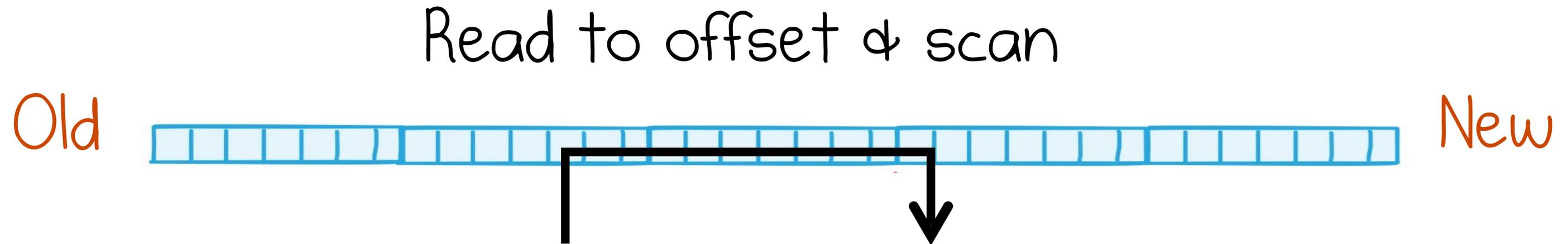


Messages are added
at the end of the log

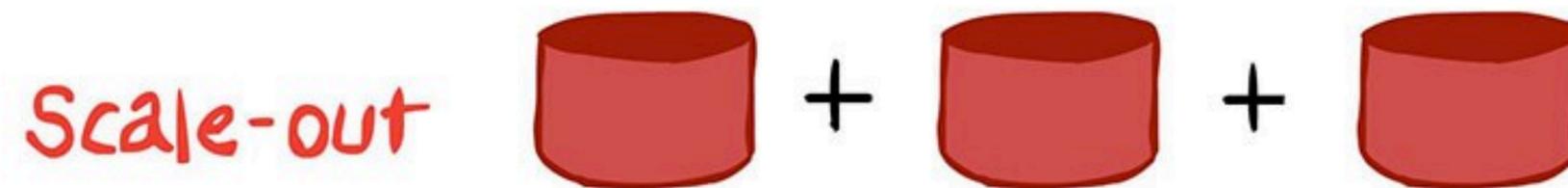
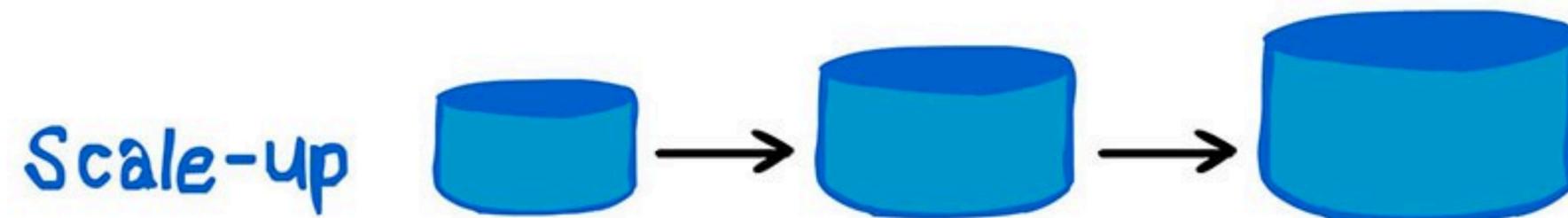
Consumers have a position all of their own



Only Sequential Access

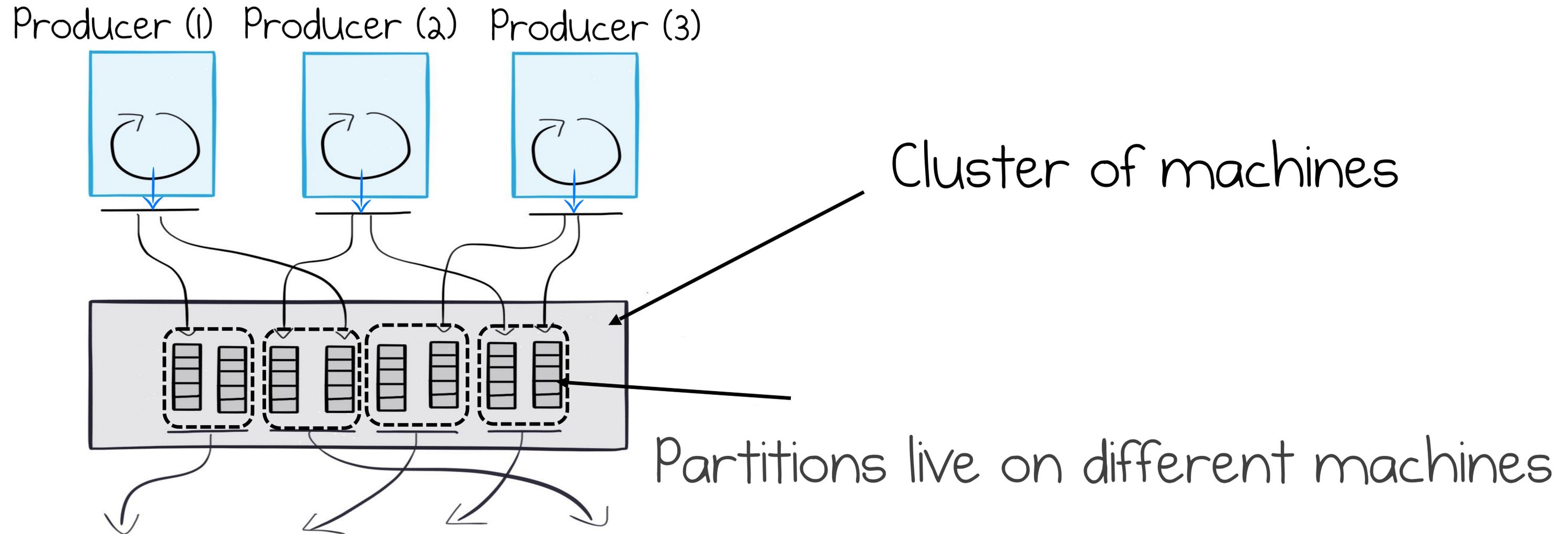


Scaling Out

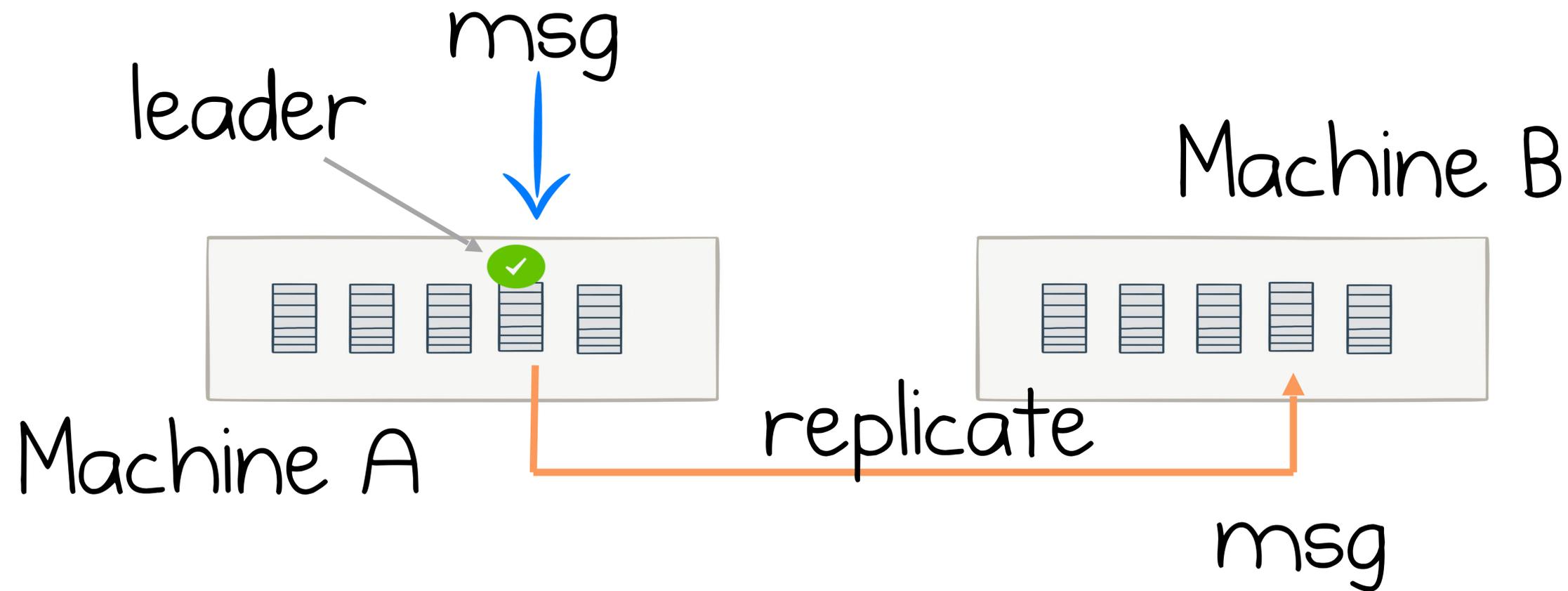


Shard data to get scalability

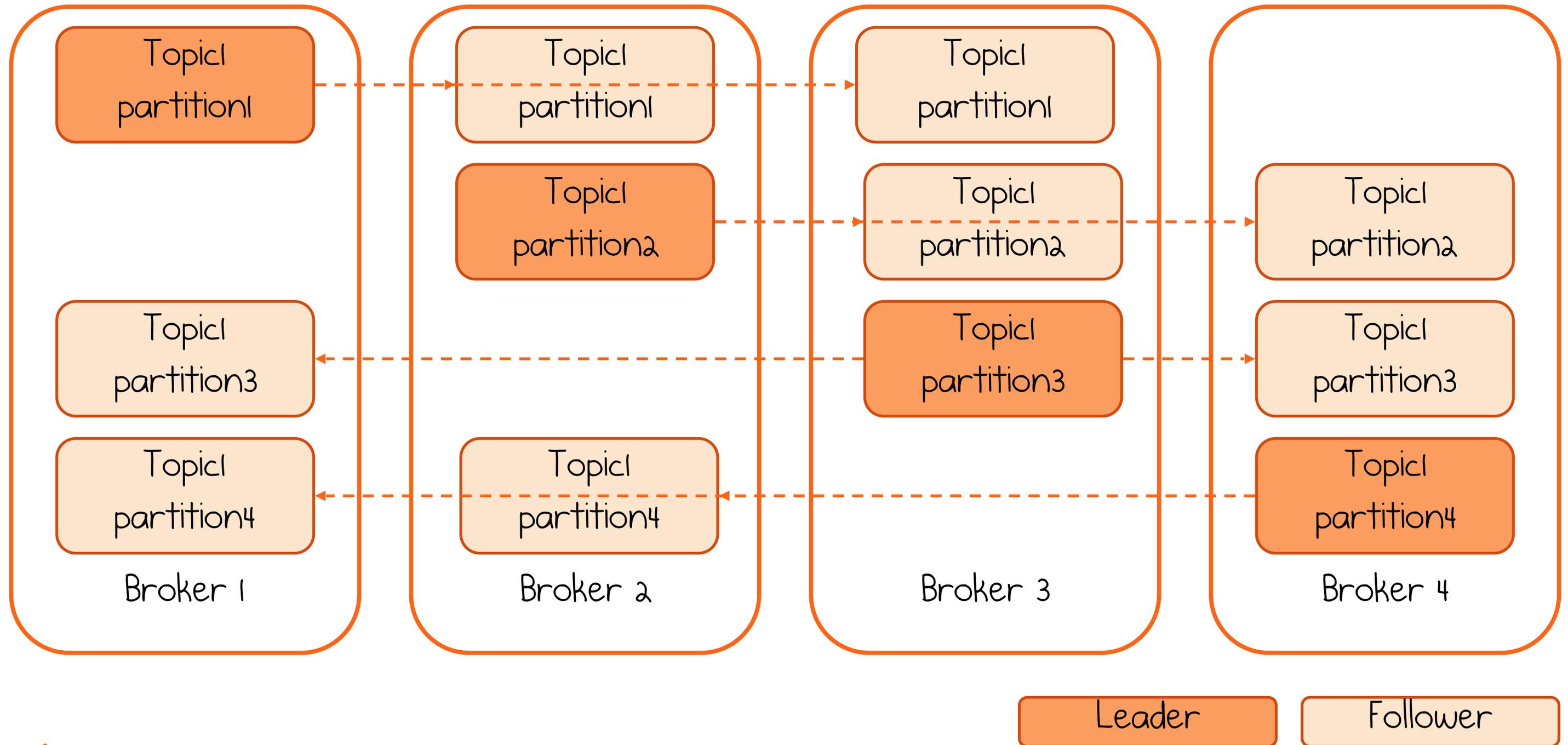
Messages are sent to different partitions



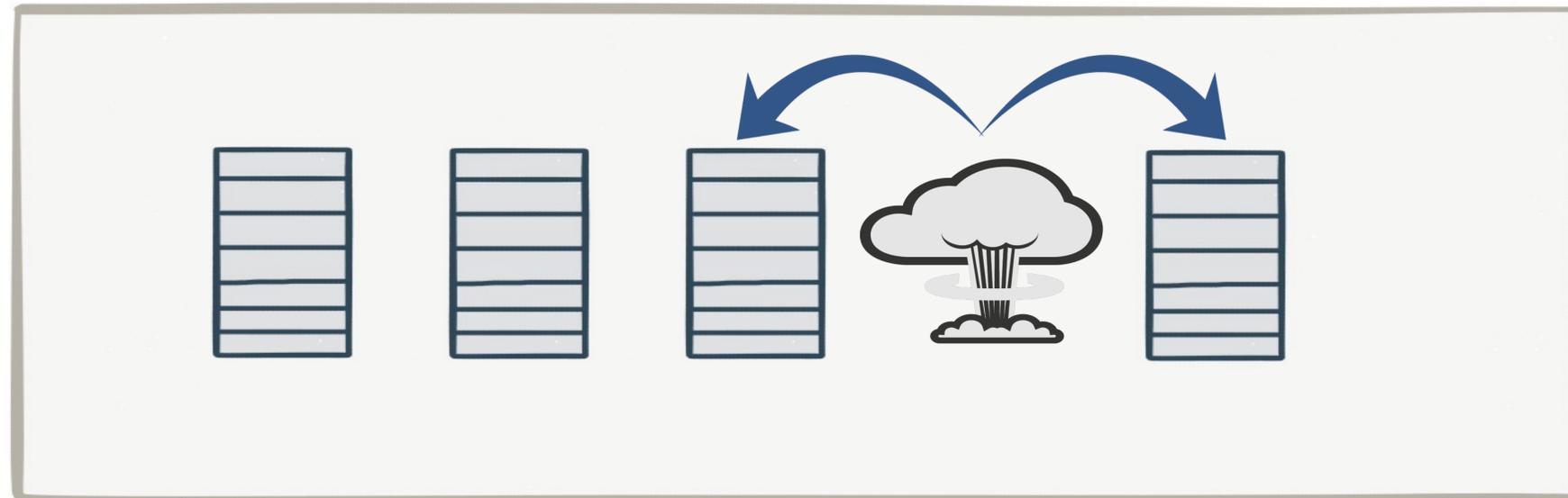
Replicate to get fault tolerance



Partition Leadership and Replication

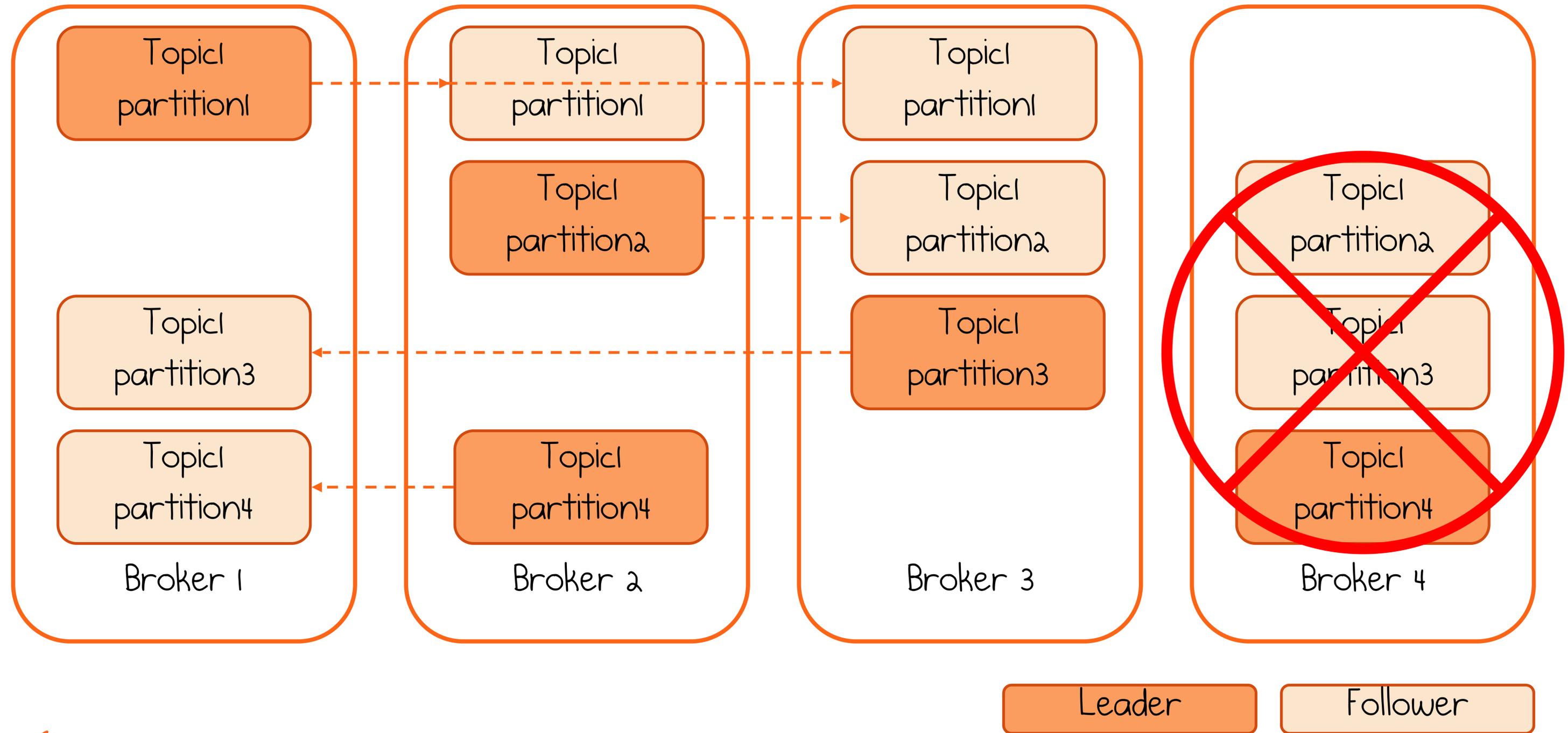


Replication provides resiliency

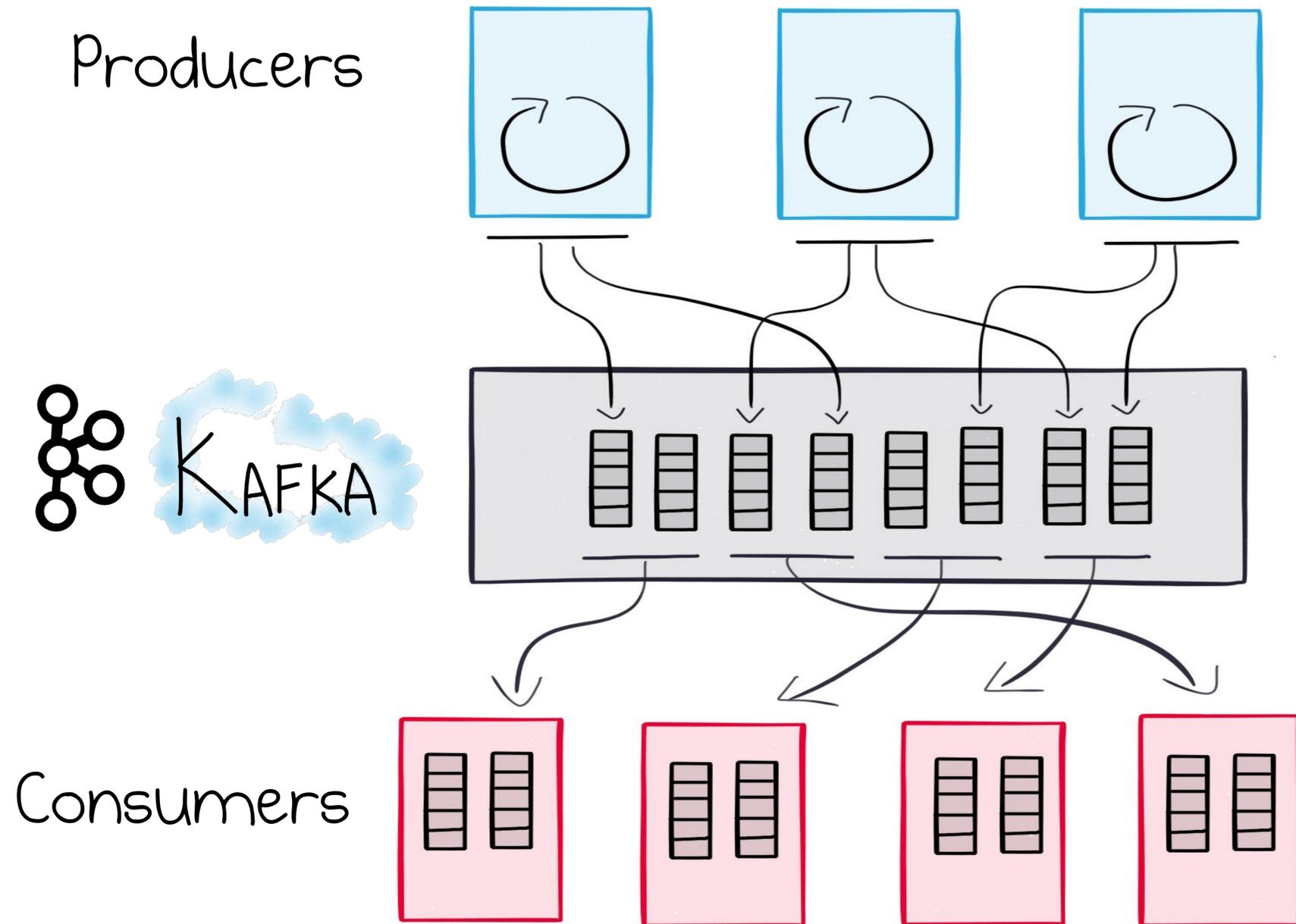


A 'replica' takes over on machine failure

Partition Leadership and Replication - node failure



Linearly Scalable Architecture

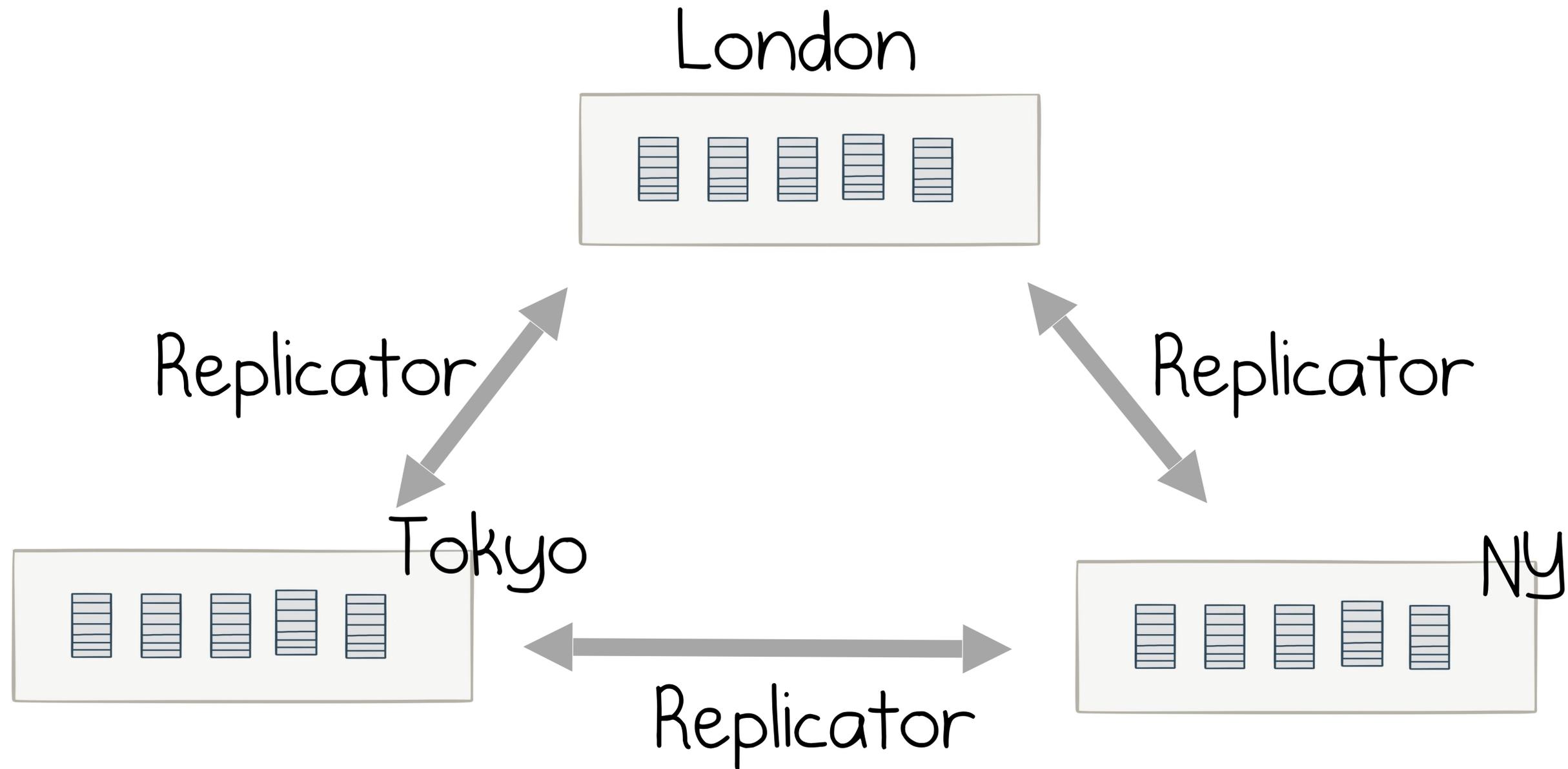


Single topic:

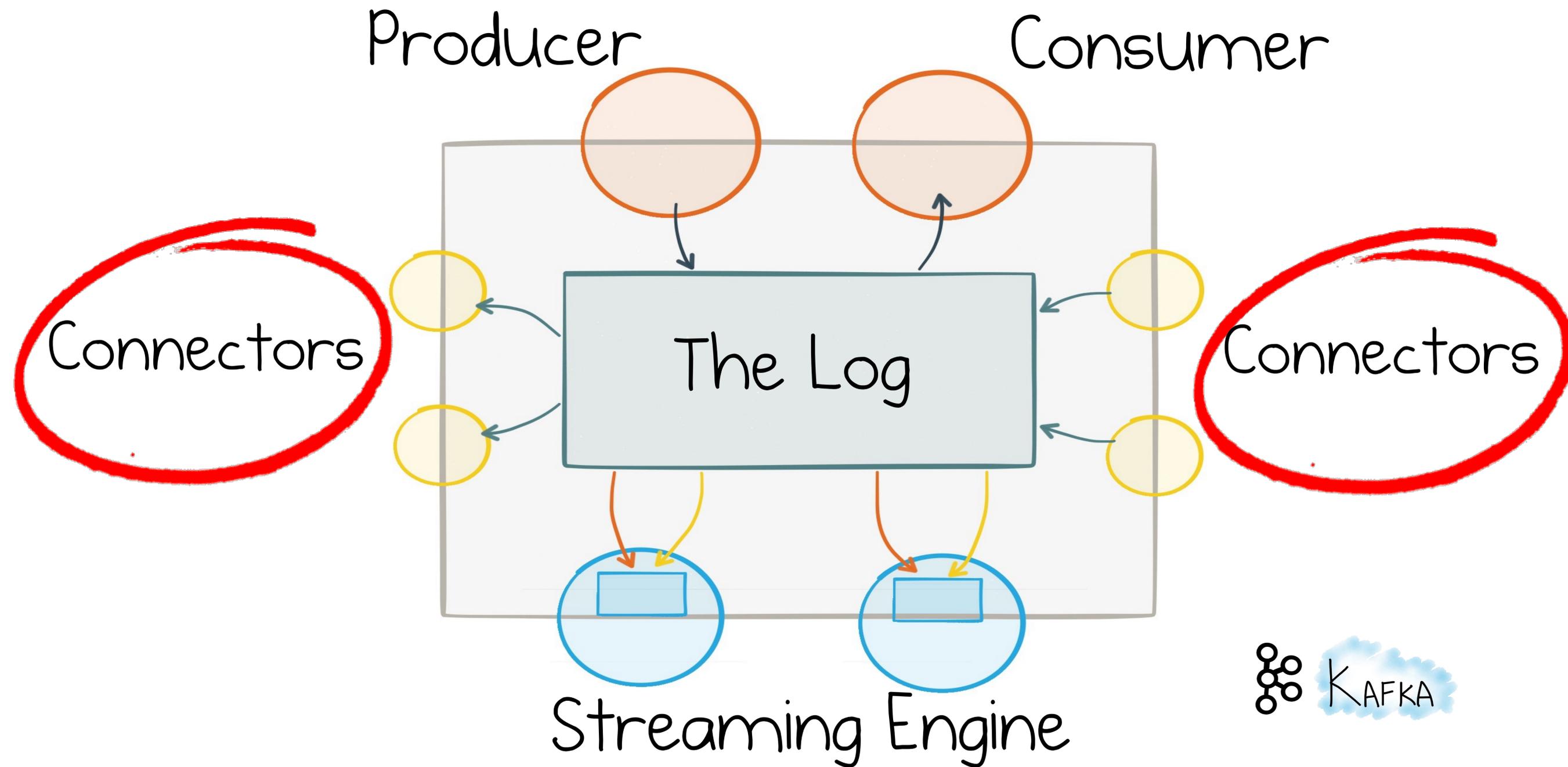
- Many producers machines
- Many consumer machines
- Many Broker machines

No Bottleneck!!

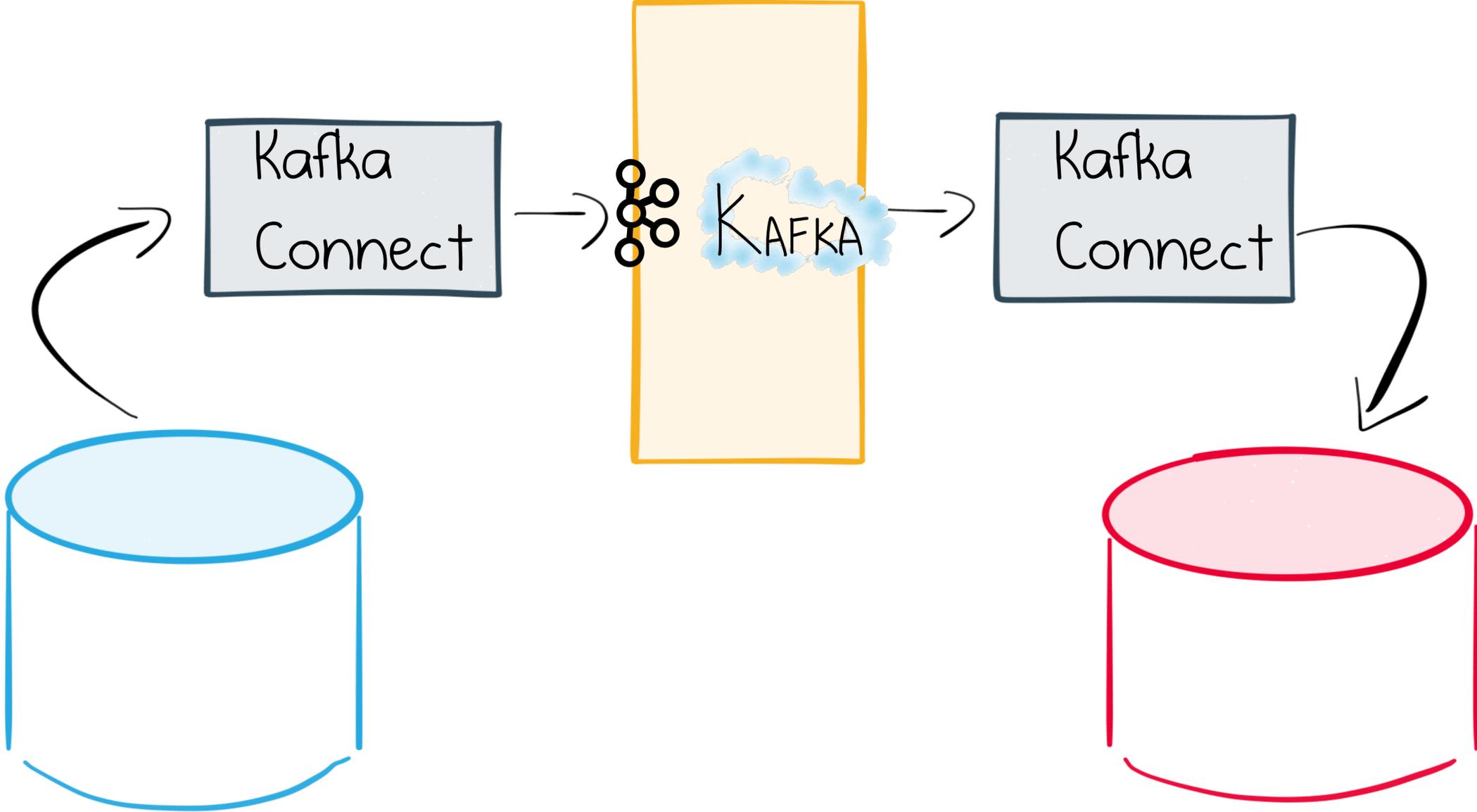
Worldwide, localized views



The Connect API



Ingest / Egest into any data source



Ingest/Egest data from/to data sources

Amazon S3

Elasticsearch

HDFS

JDBC

Couchbase

Cassandra

Oracle

SAP

Vertica

Blockchain

DynamoDB

FTP

Github

BigQuery

Google Pub Sub

RethinkDB

Salesforce

Solr

Splunk

JMX

Kinesis

MongoDB

MQTT

NATS

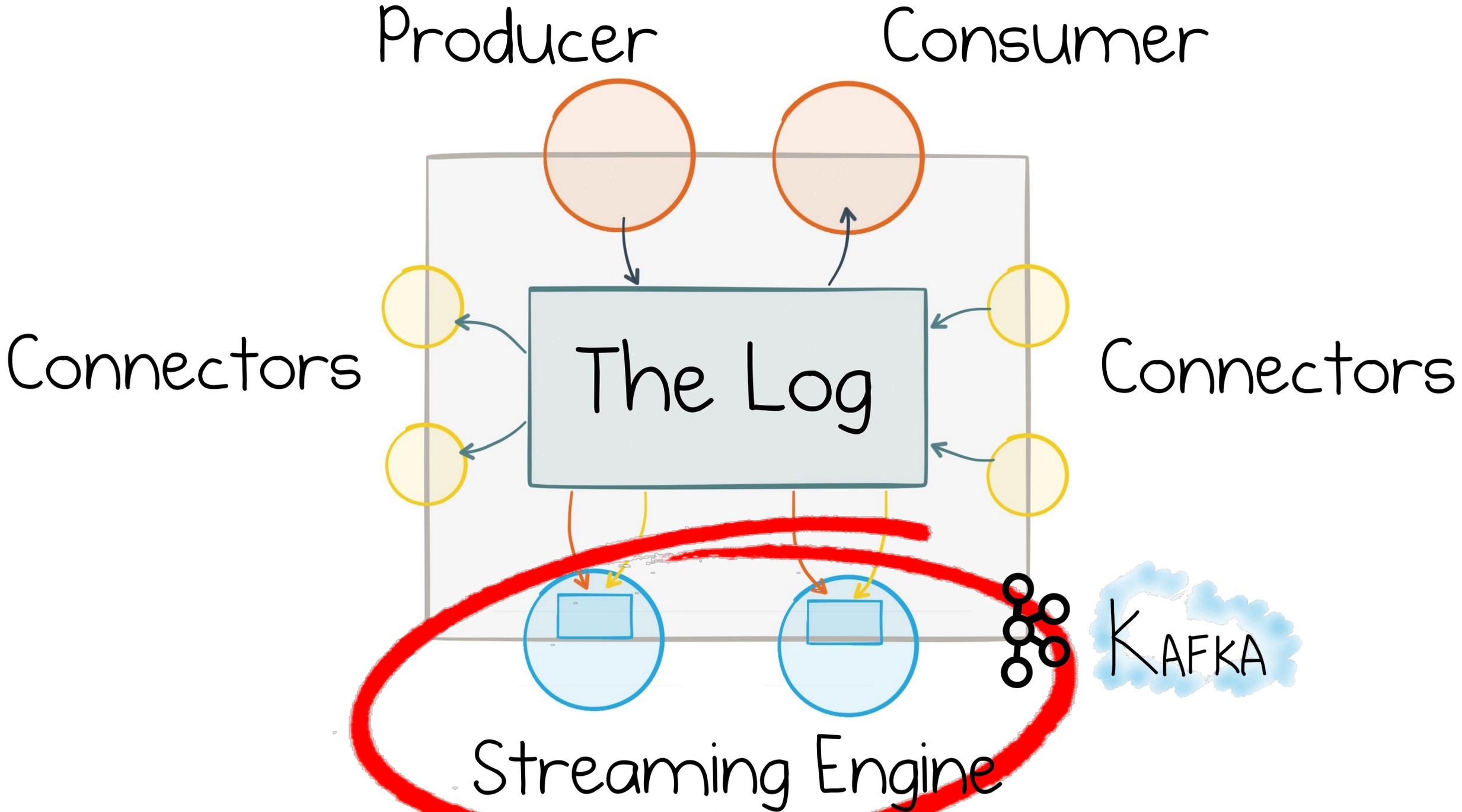
Postgres

Rabbit

Redis

Twitter

Kafka Streams and KSQL



Engine for Continuous Computation



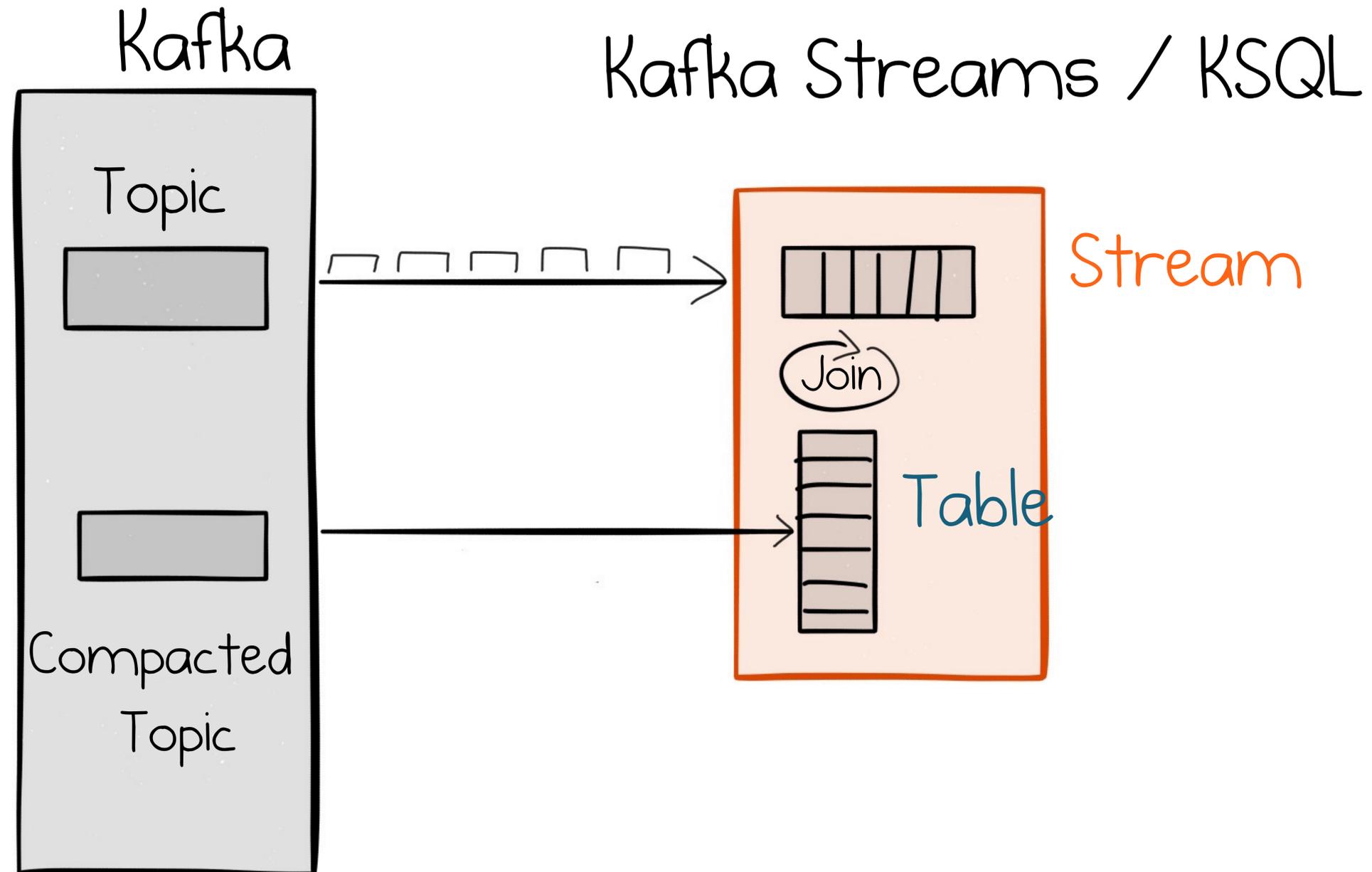
```
SELECT card_number, count(*)  
FROM authorization_attempts  
WINDOW (SIZE 5 MINUTE)  
GROUP BY card_number  
HAVING count(*) > 3;
```



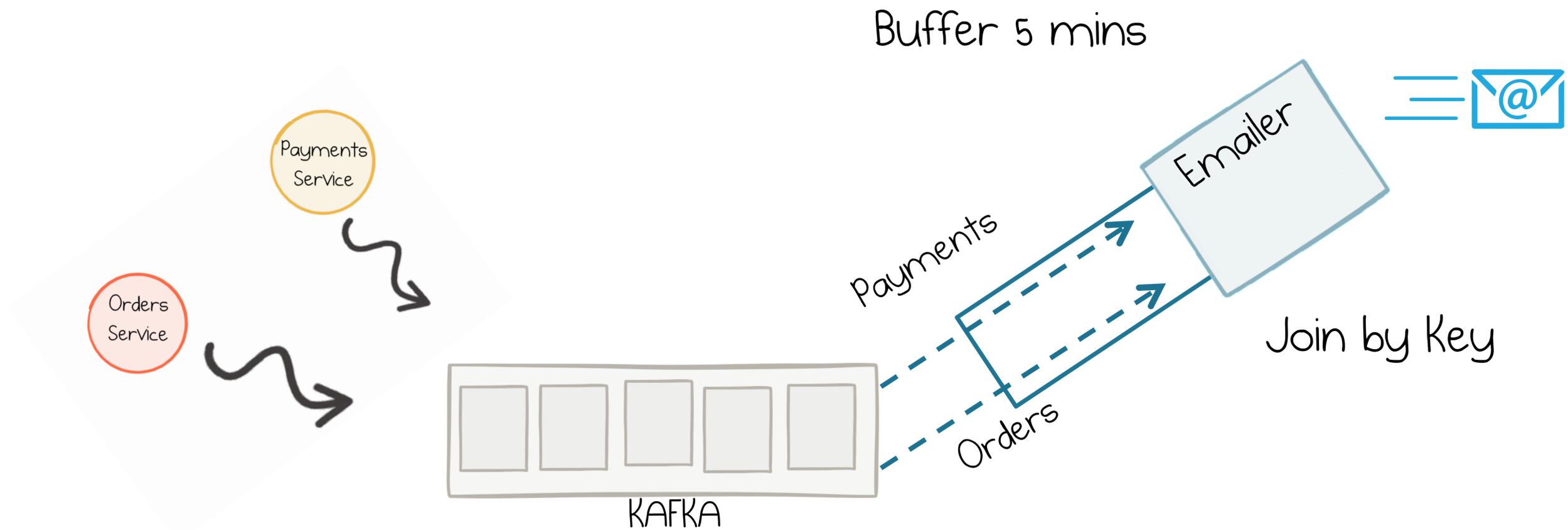
But it's just an API

```
public static void main(String[] args) {  
  
    StreamsBuilder builder = new StreamsBuilder();  
    builder.stream("caterpillars")  
        .map(StreamsApp::coolTransformation)  
        .to("butterflies");  
  
    new KafkaStreams(builder.build(), props()).start();  
  
}
```

Join Streams and Tables



Windows / Retention - Handle Late Events

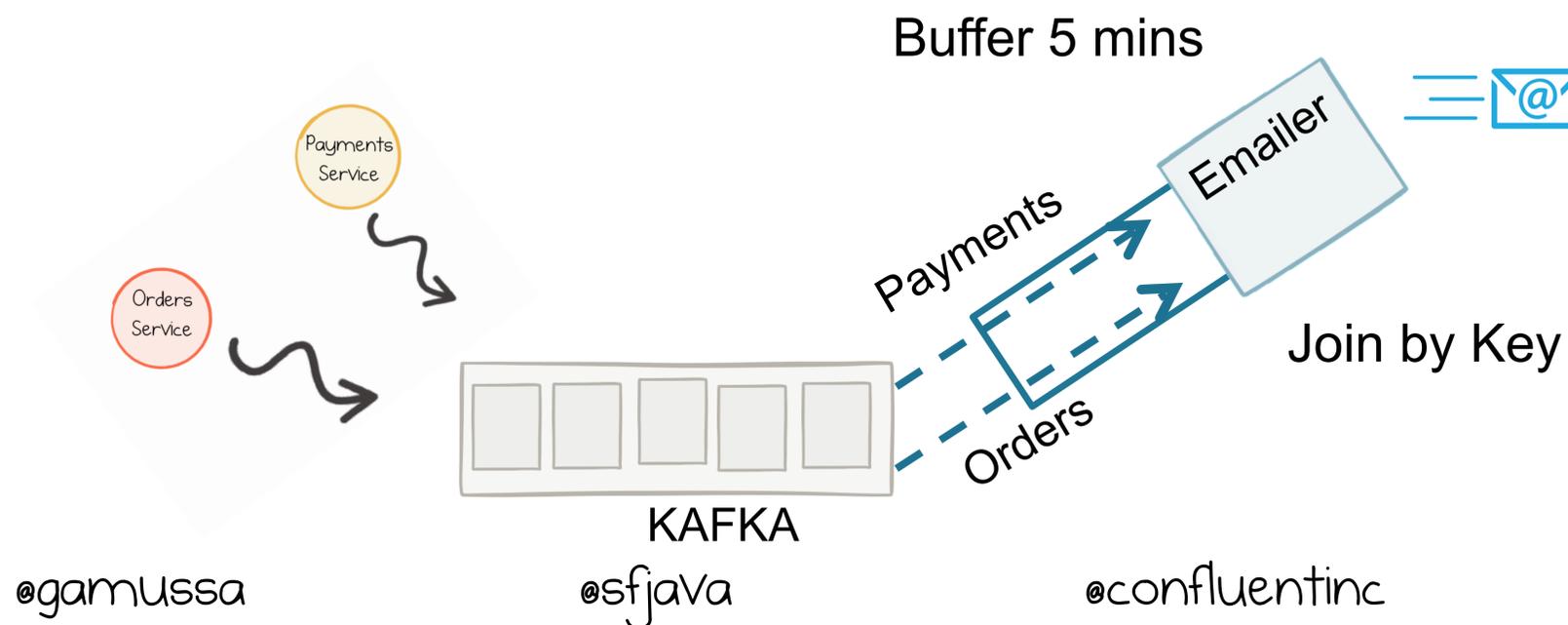


In an asynchronous world, will the payment come first, or the order?

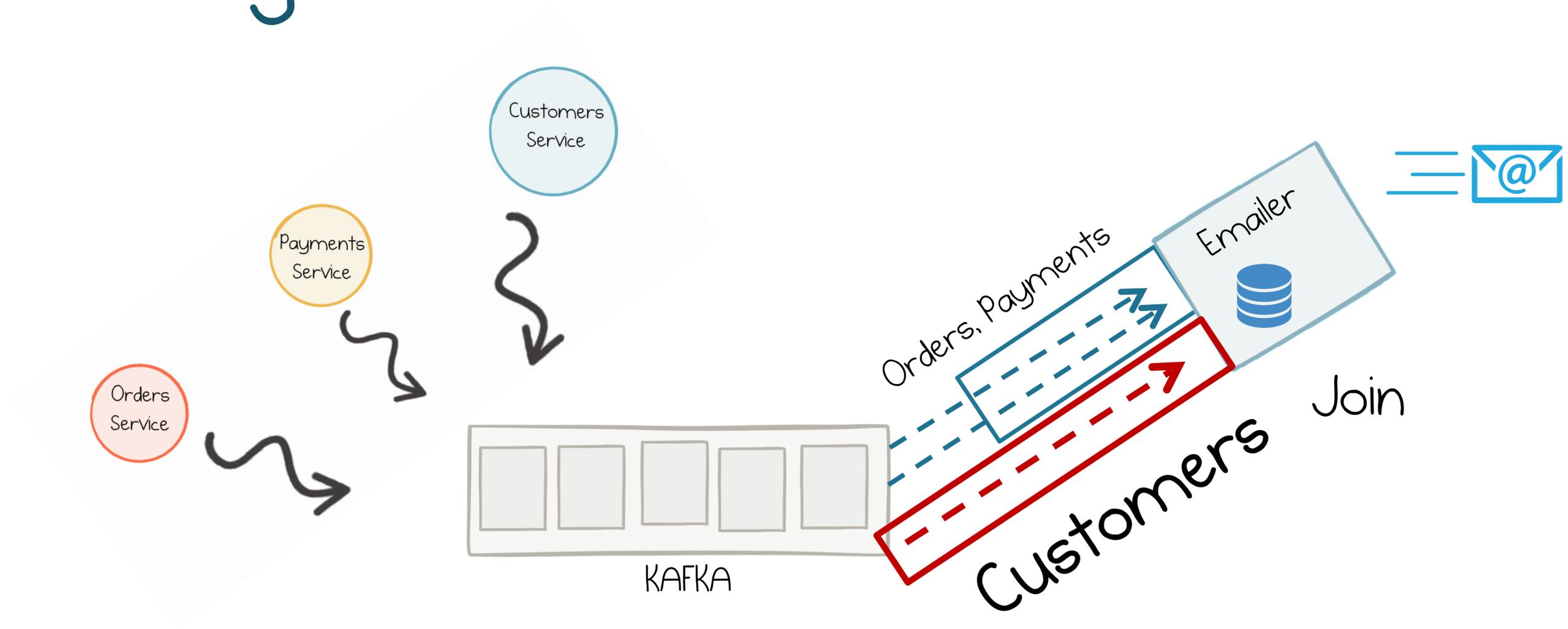
Windows / Retention - Handle Late Events

```
KStream orders = builder.stream("Orders");
KStream payments = builder.stream("Payments");

orders.join(payments,
           KeyValue::new,
           JoinWindows.of(1 * MIN))
      .peek((key, pair) → emailer.sendMail(pair));
```



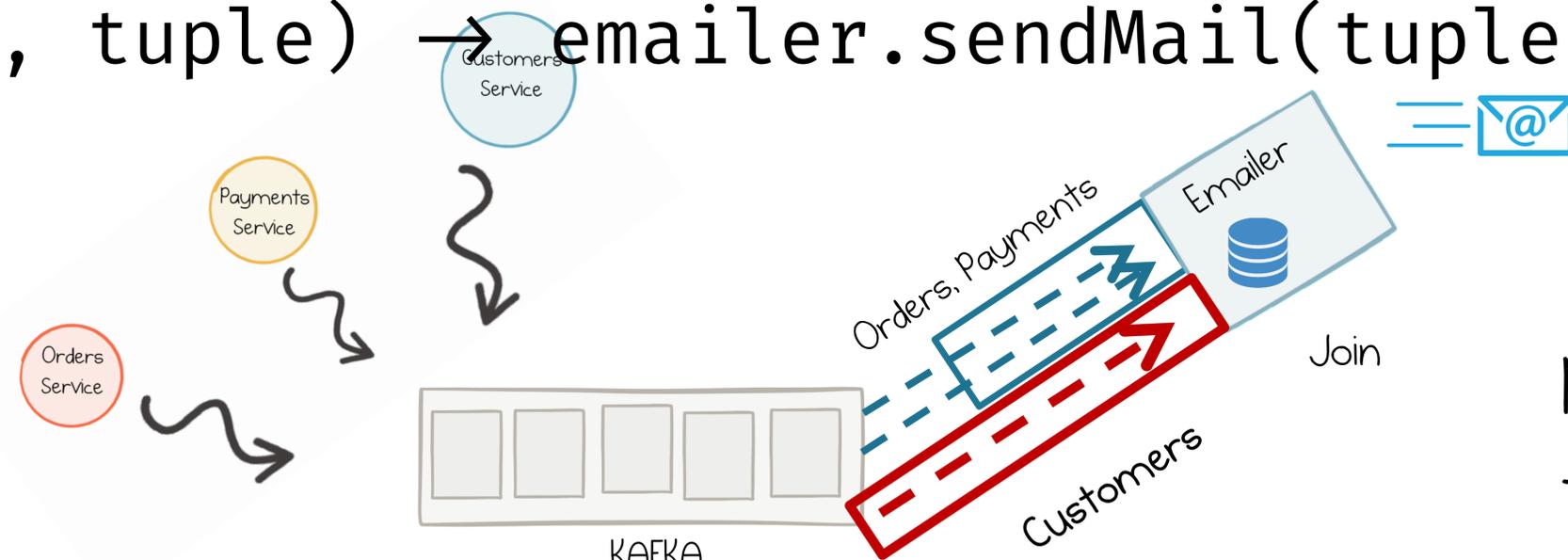
A KTable is just a stream with infinite retention



A KTable is a stream with infinite retention

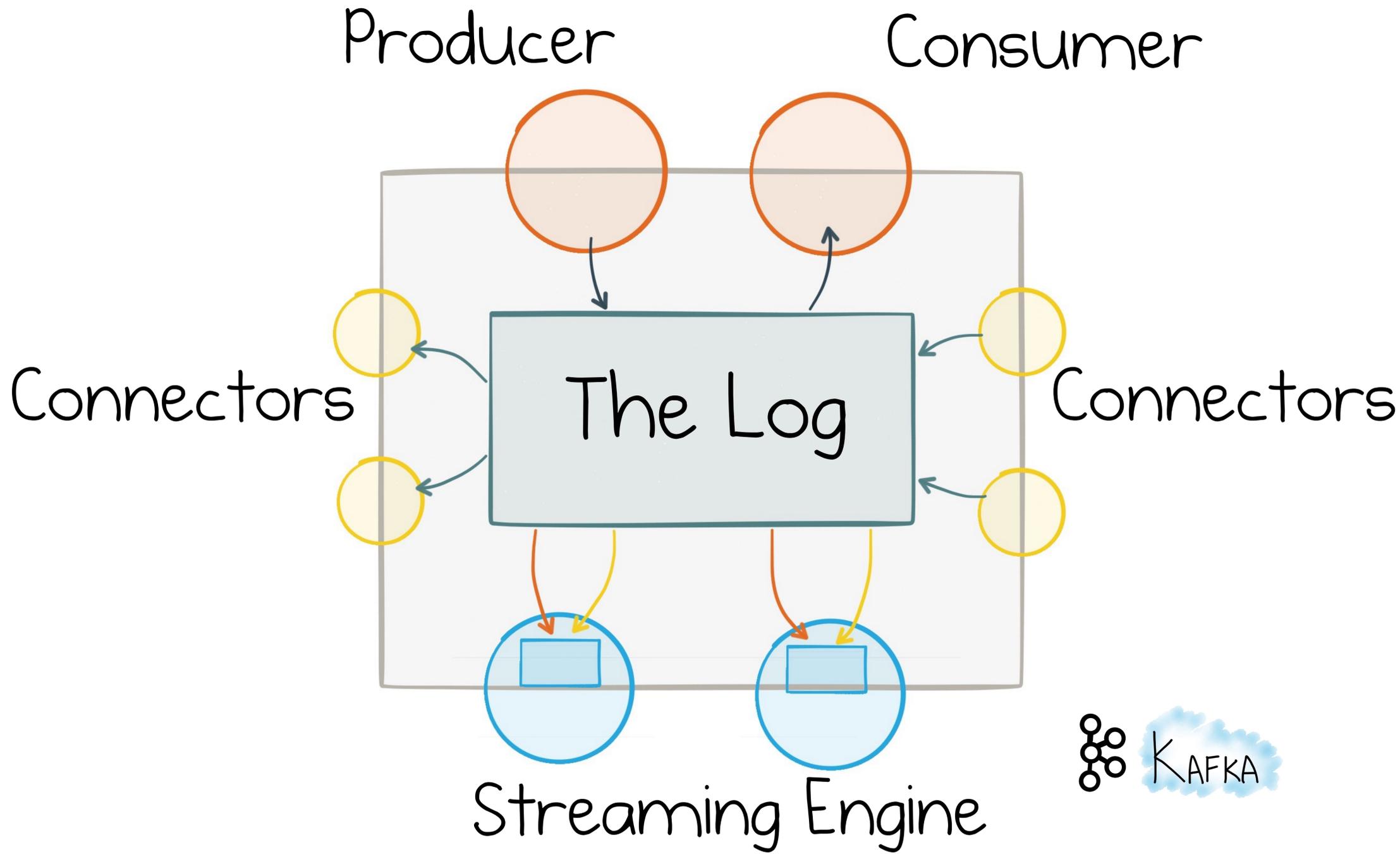
```
KStream orders = builder.stream("Orders");  
KStream payments = builder.stream("Payments");  
KTable customers = builder.table("Customers");
```

```
orders.join(payments, EmailTuple::new, JoinWindows.of(1*MIN))  
    .join(customers, (tuple, cust) → tuple.setCust(cust))  
    .peek((key, tuple) → emailer.sendMail(tuple));
```

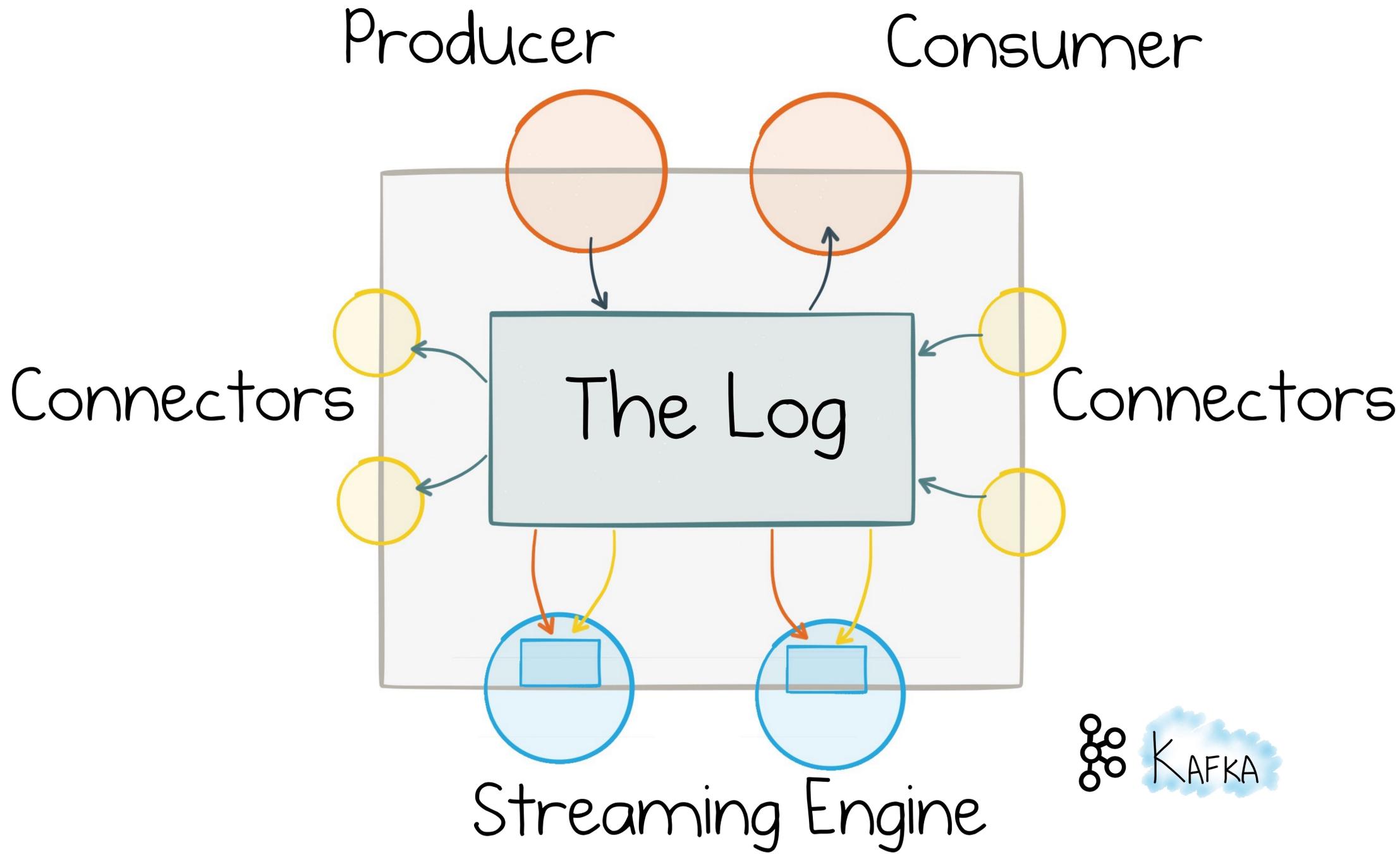


Materialize a table in
two lines of code!

Kafka is a complete Streaming Platform



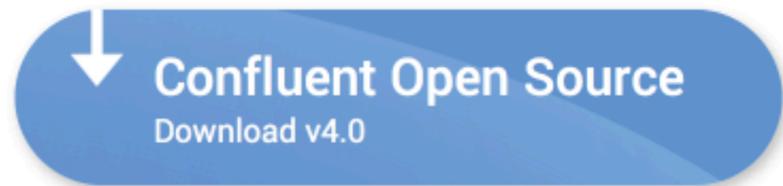
Kafka is a complete Streaming Platform



THE LEADING DISTRIBUTION OF APACHE KAFKA

Downloads <https://www.confluent.io/download/>

Choose between Confluent Open Source or Confluent Enterprise edition. Both are built on the world's most popular streaming platform, Apache Kafka®.



Free & Open Source

Confluent Open Source is a tested, complete package around Apache Kafka that includes all the features needed to build a streaming platform.

- Apache Kafka ...
- Build in any language ...
- Manage your schema ...
- Connect popular systems ...

30-Day Free Trial

Confluent Enterprise includes Confluent Open Source, and adds features that improve scale, performance and reliability in production.

- Monitor your deployment ...
- Achieve high availability ...
- Automate resource allocation ...
- Have 24/7 support ...



@gamussa

@sfjava

@confluentinc

We are hiring!



<https://www.confluent.io/careers/>



@gamussa

@sfjava

@confluentinc

One more thing...

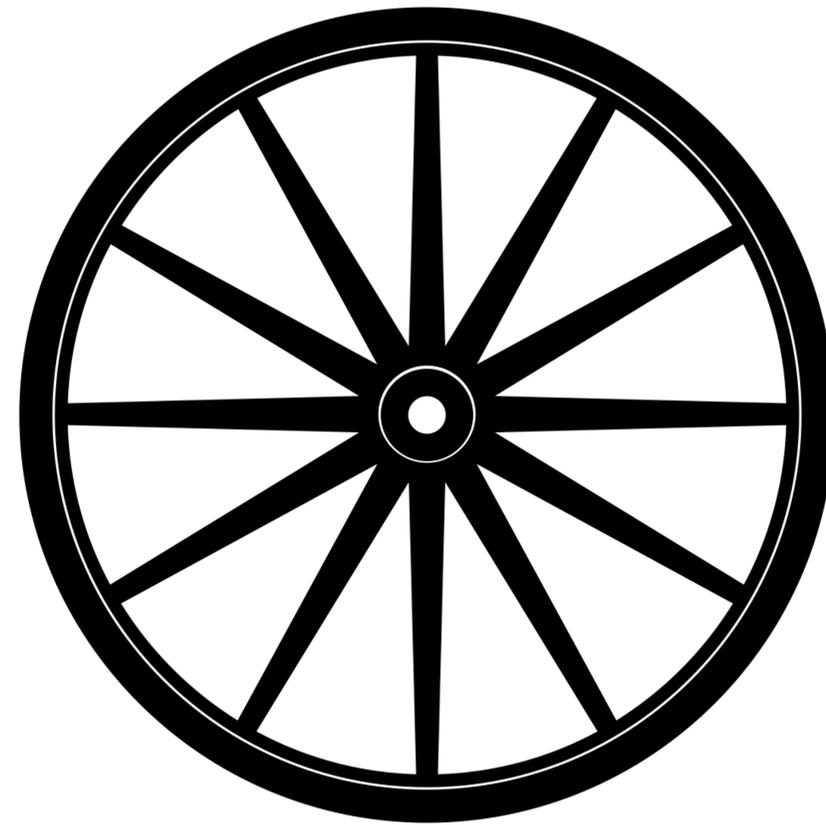


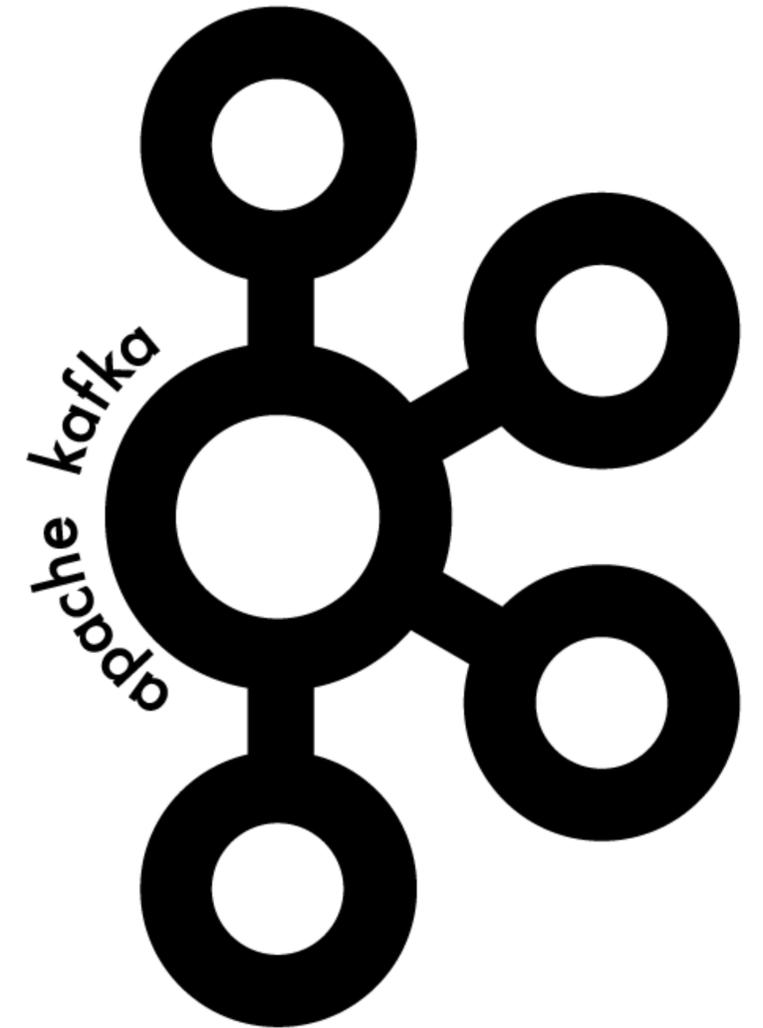
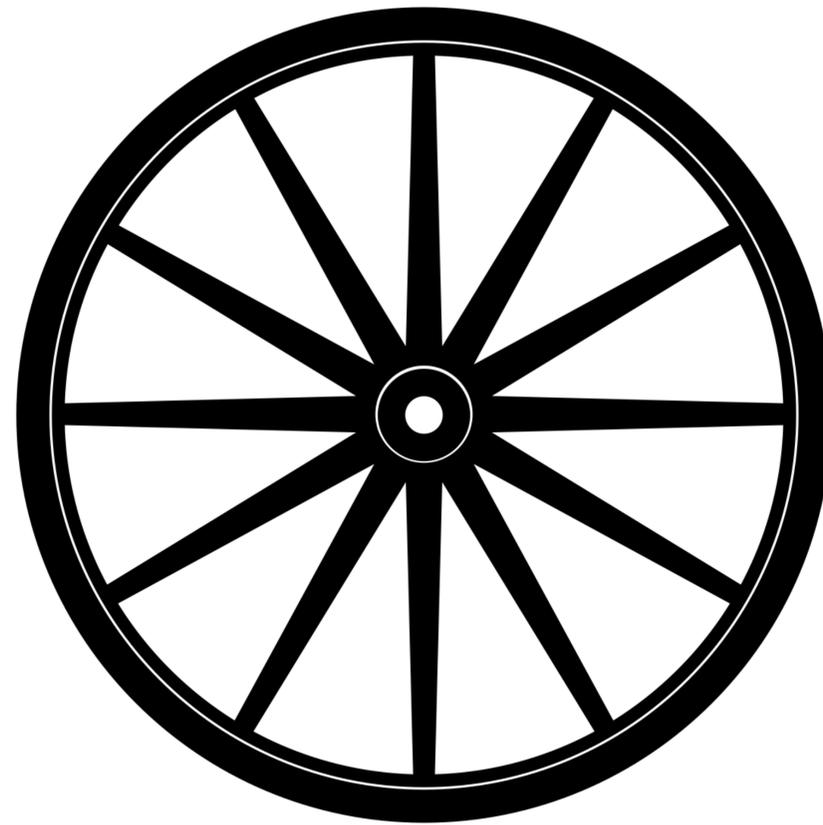
@gamussa

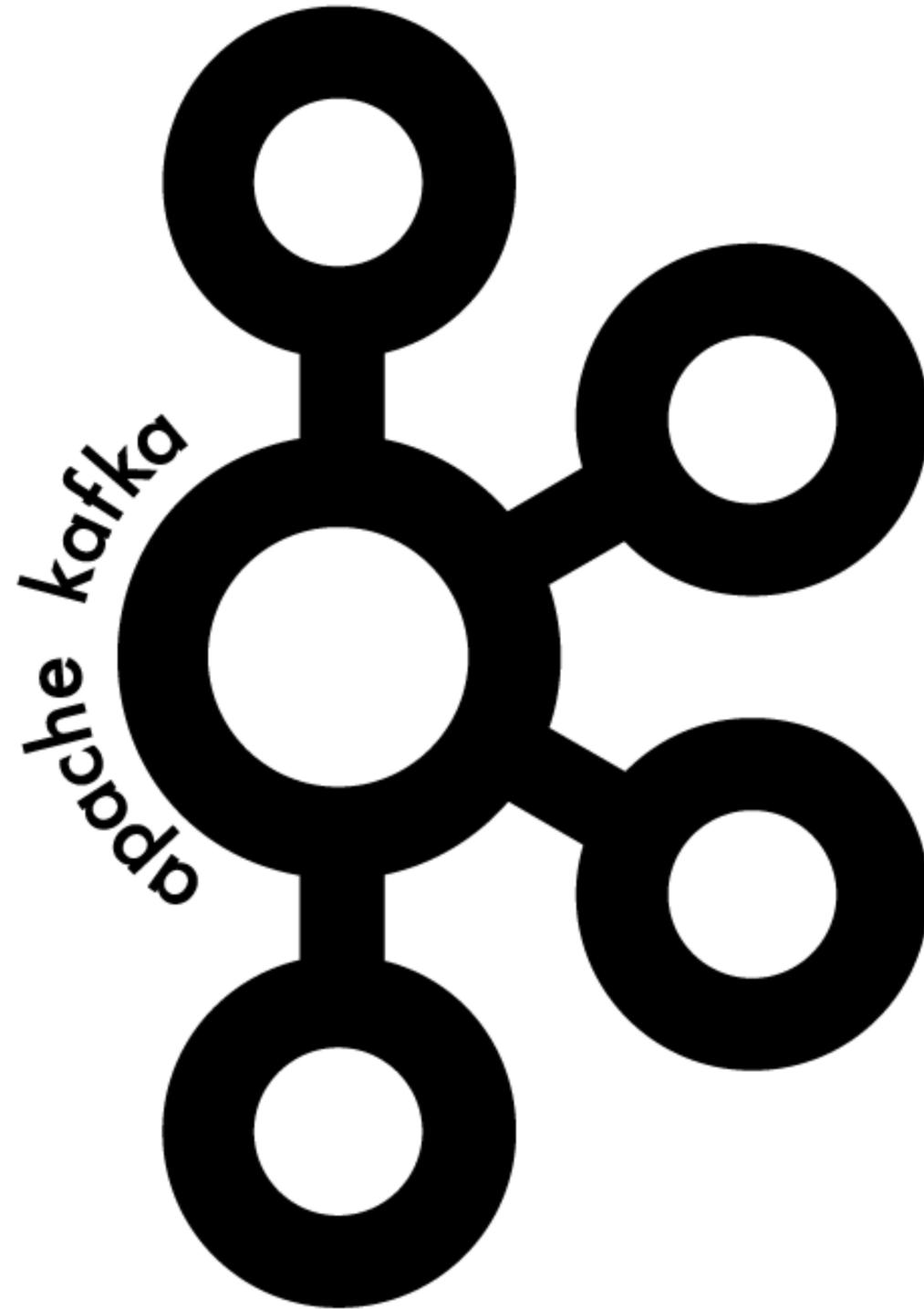
@sfjava

@confluentinc

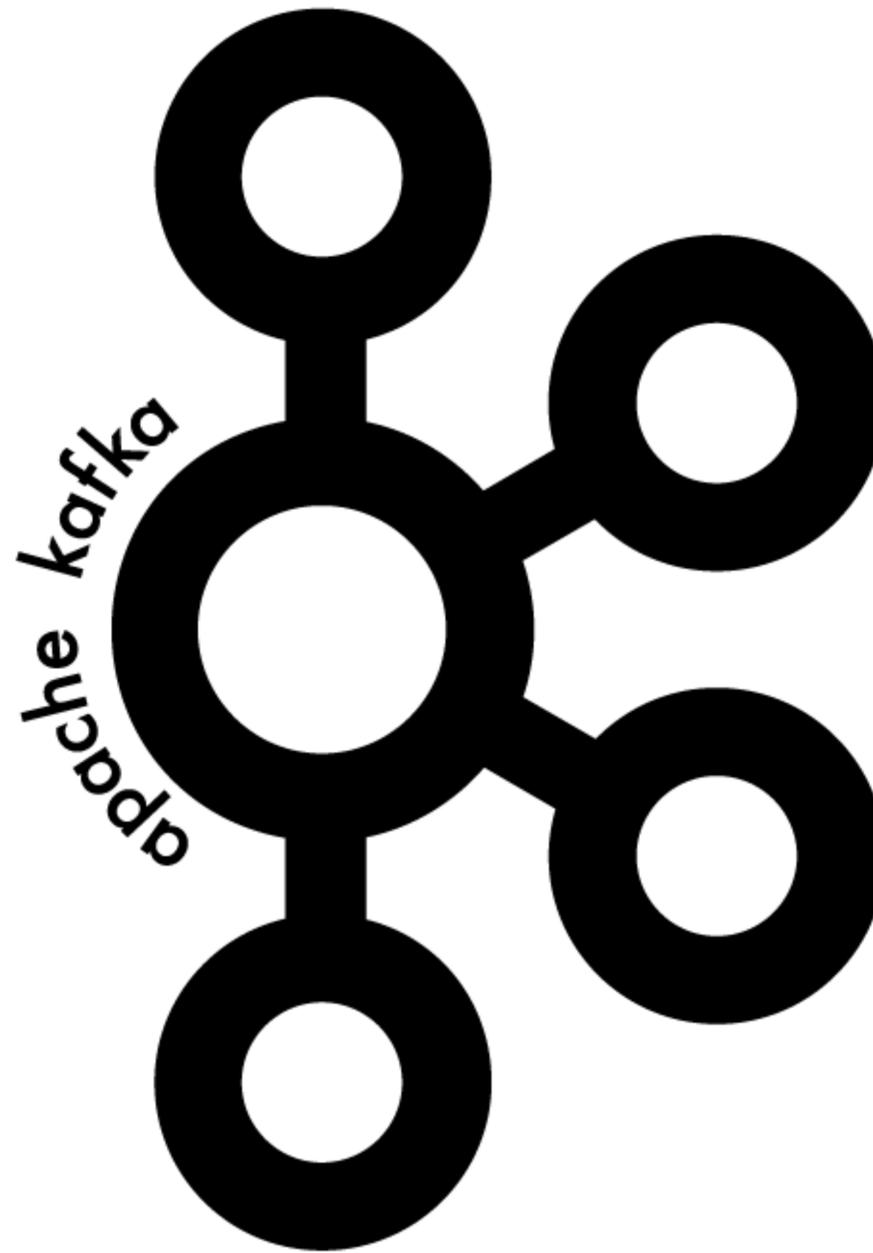








A Major New Paradigm





THANKS!

Stay for #javapuzzlersng!!!

@gamussa

viktor@confluent.io

We are hiring!

<https://www.confluent.io/careers/>