# Integrating Apache Kafka with Structured Streaming

**Janani Ravi**
CO-FOUNDER, LOONYCORN

www.loonycorn.com

# Overview

Kafka is a powerful publisher/subscriber messaging technology

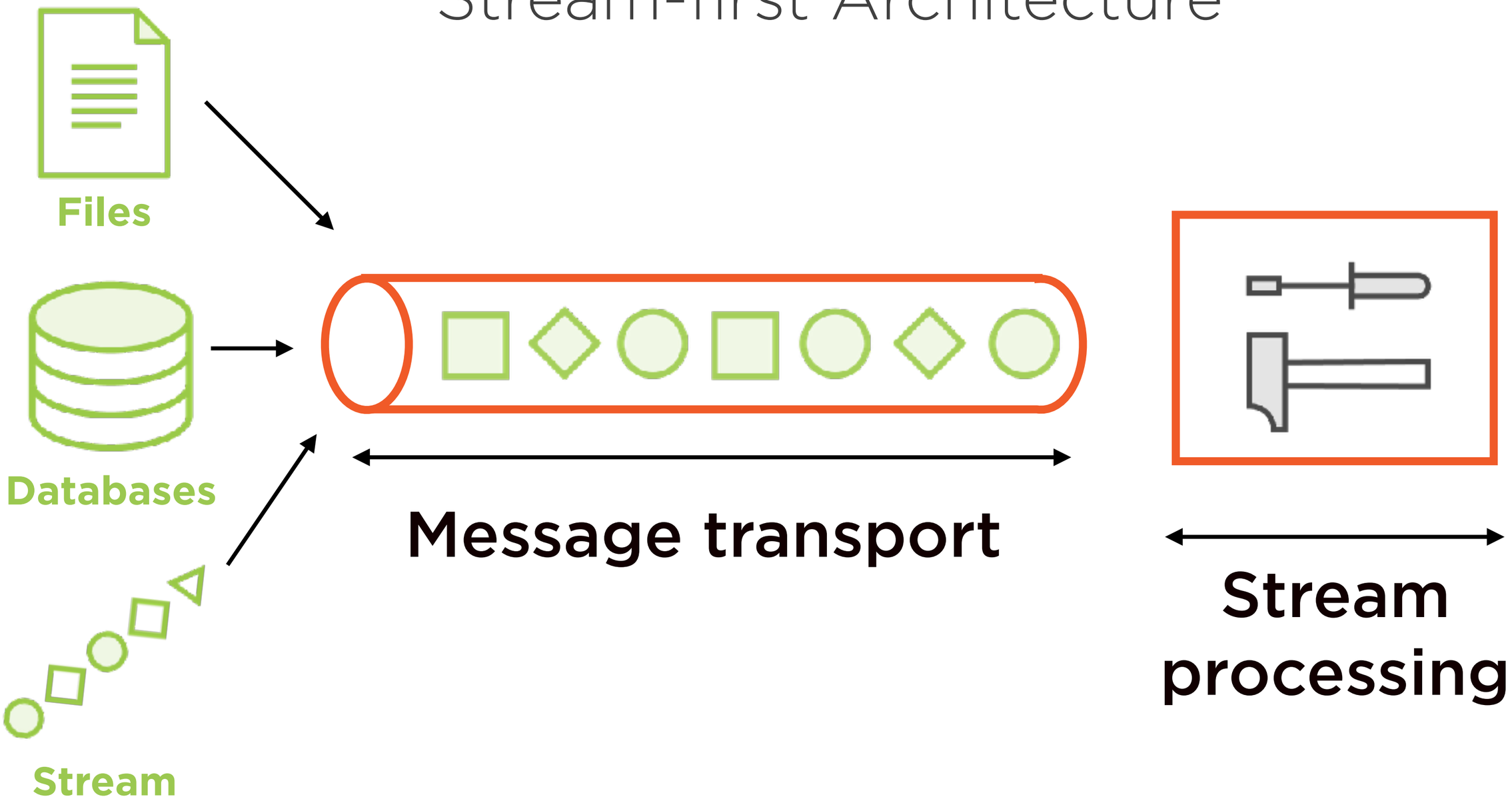
Producers publish, consumers subscribe

Messages are categorized by topic and stored in partitioned, replicated logs

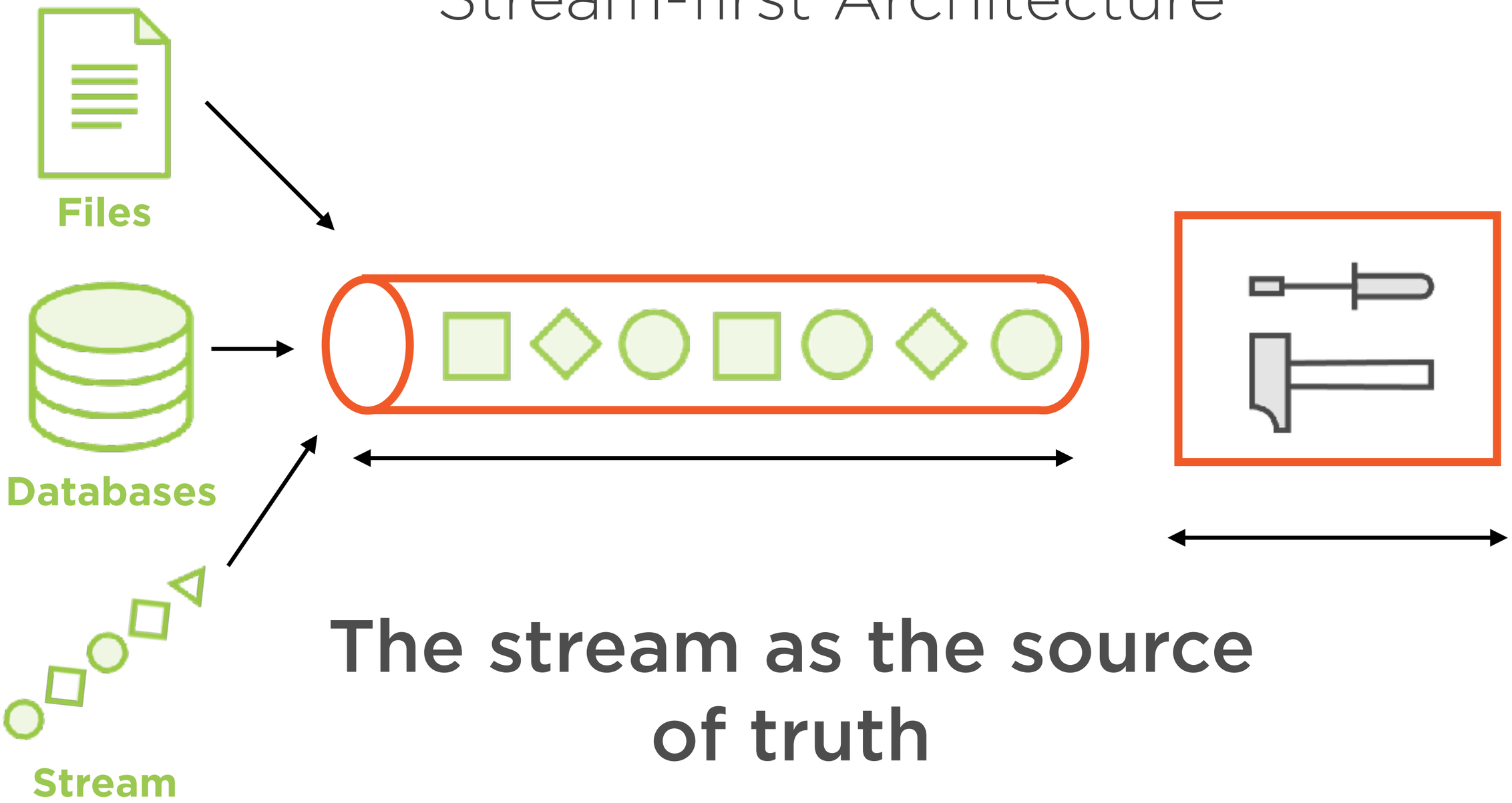
Kafka is distributed and uses Zookeeper internally

Structured Streaming and Kafka interface in powerful ways

# Stream-first Architecture



# Stream-first Architecture




**The stream as the source of truth**

# Kafka

**What**

Distributed publisher/ subscriber messaging

**How**

Internally uses Zookeeper, partitioning

**Why**

Distributed, scalable, low-latency

# Kafka





**What**

Distributed publisher/ subscriber messaging

**How**

Internally uses Zookeeper, partitioning

**Why**

Distributed, scalable, low-latency

# Publishers, Topics and Subscribers

**Publisher**

**Topic**

**Subscriber**

**Messages**

# Publishers, Topics and Subscribers

**Producers**


**Topics**

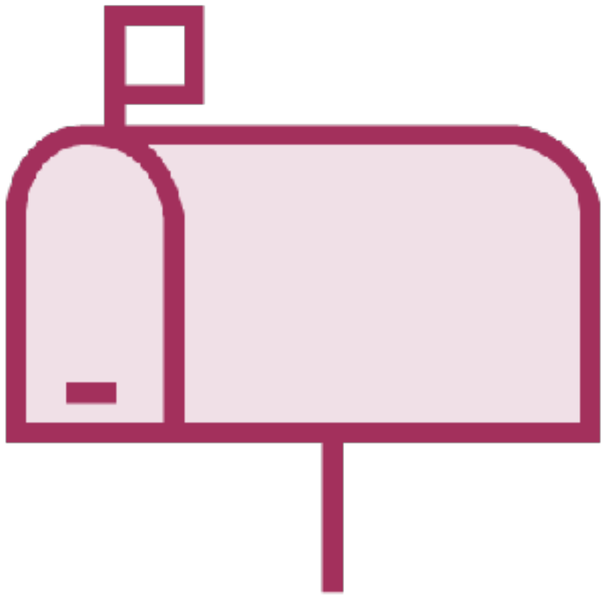

**Consumers**


**Records**

# Kafka

Kafka is a distributed publish-subscribe messaging system that is designed to be fast, scalable, and durable.

*https://blog.cloudera.com/blog/2014/09/apache-kafka-for-beginners/*



# Capabilities

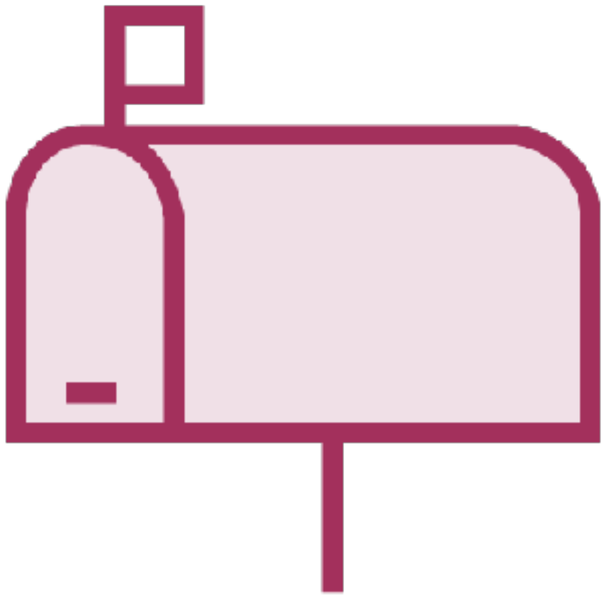
**Publish streams of records**

**Subscribe to streams of records**

**Fault-tolerant, durable record storage**

**Process stream elements as they appear**

# Clusters and Brokers

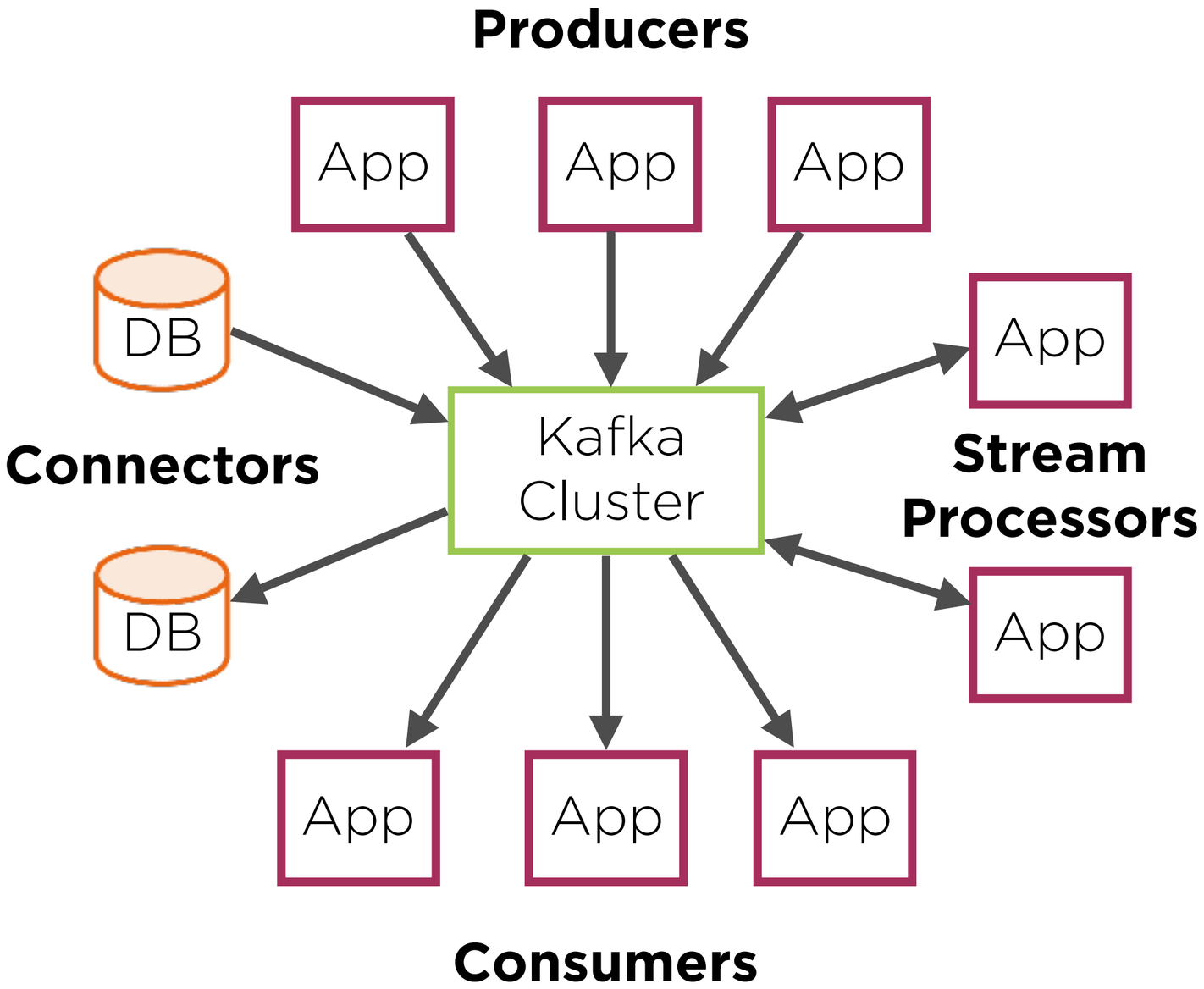


Kafka is a distributed system

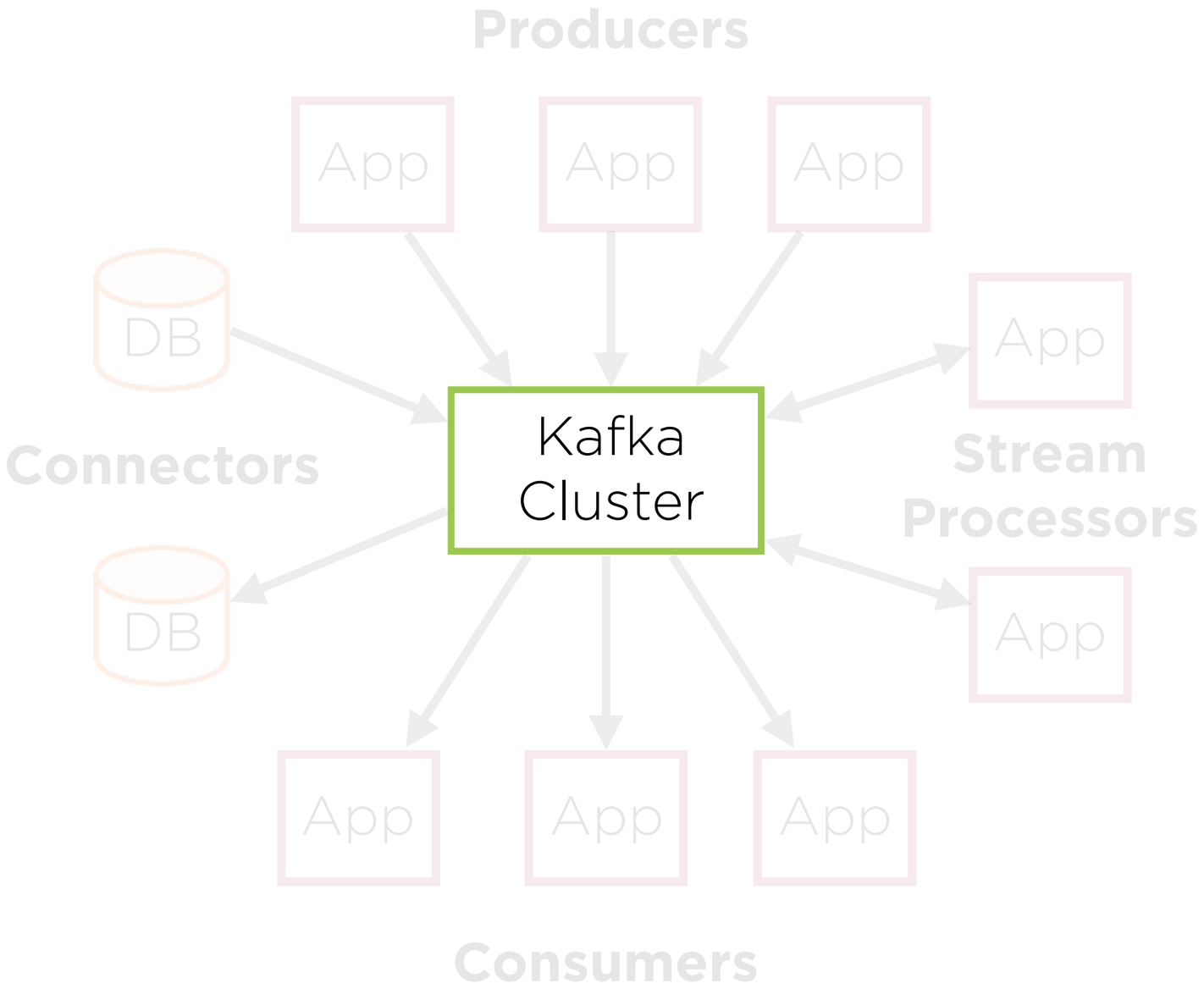
Runs on a cluster

Each node in cluster is called a broker

# Kafka APIs

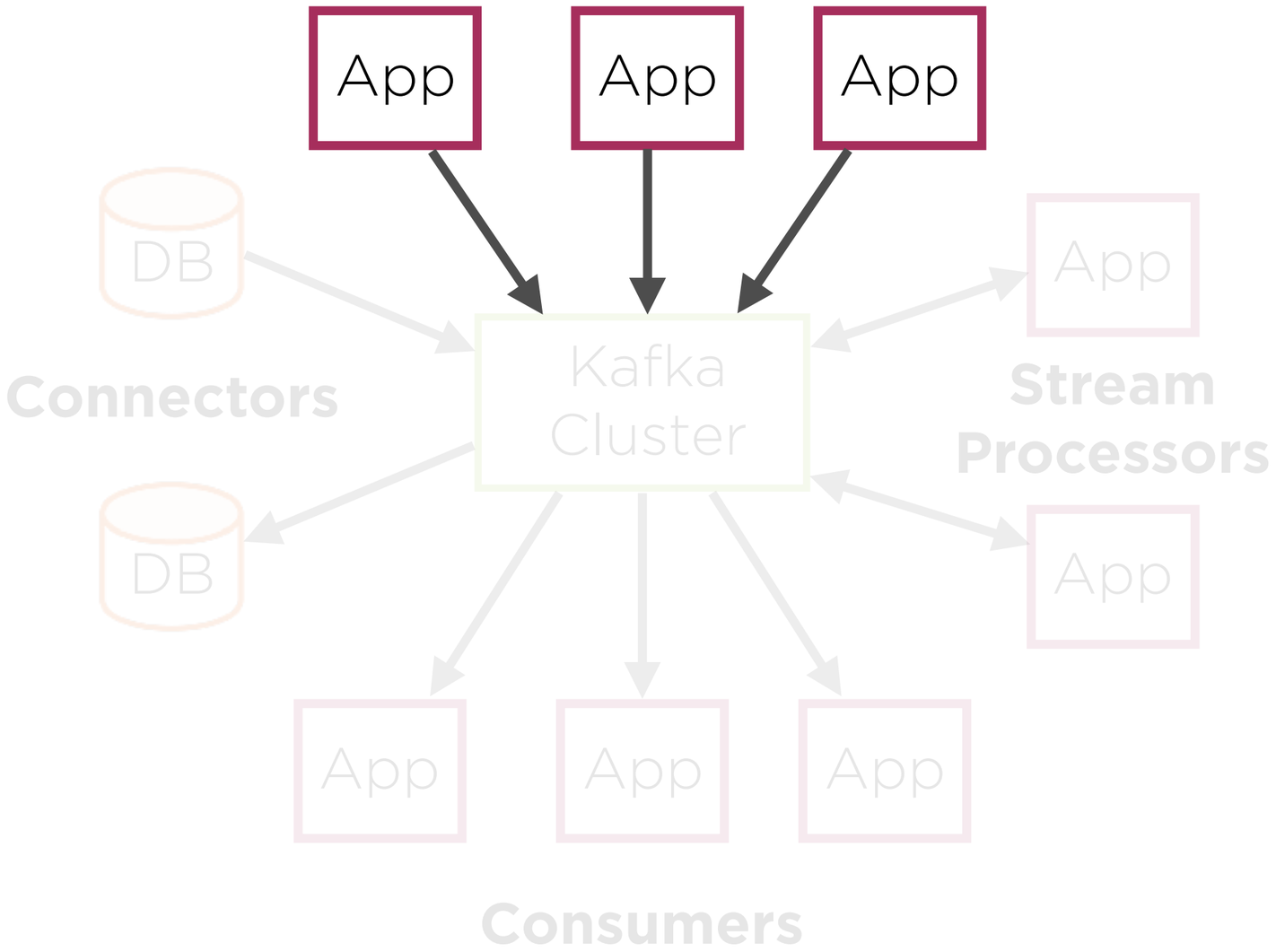

Producers
App
App
App
DB
DB
Connectors
Kafka
Cluster
App
Stream
Processors
App
App
App
App
Consumers

# Kafka APIs

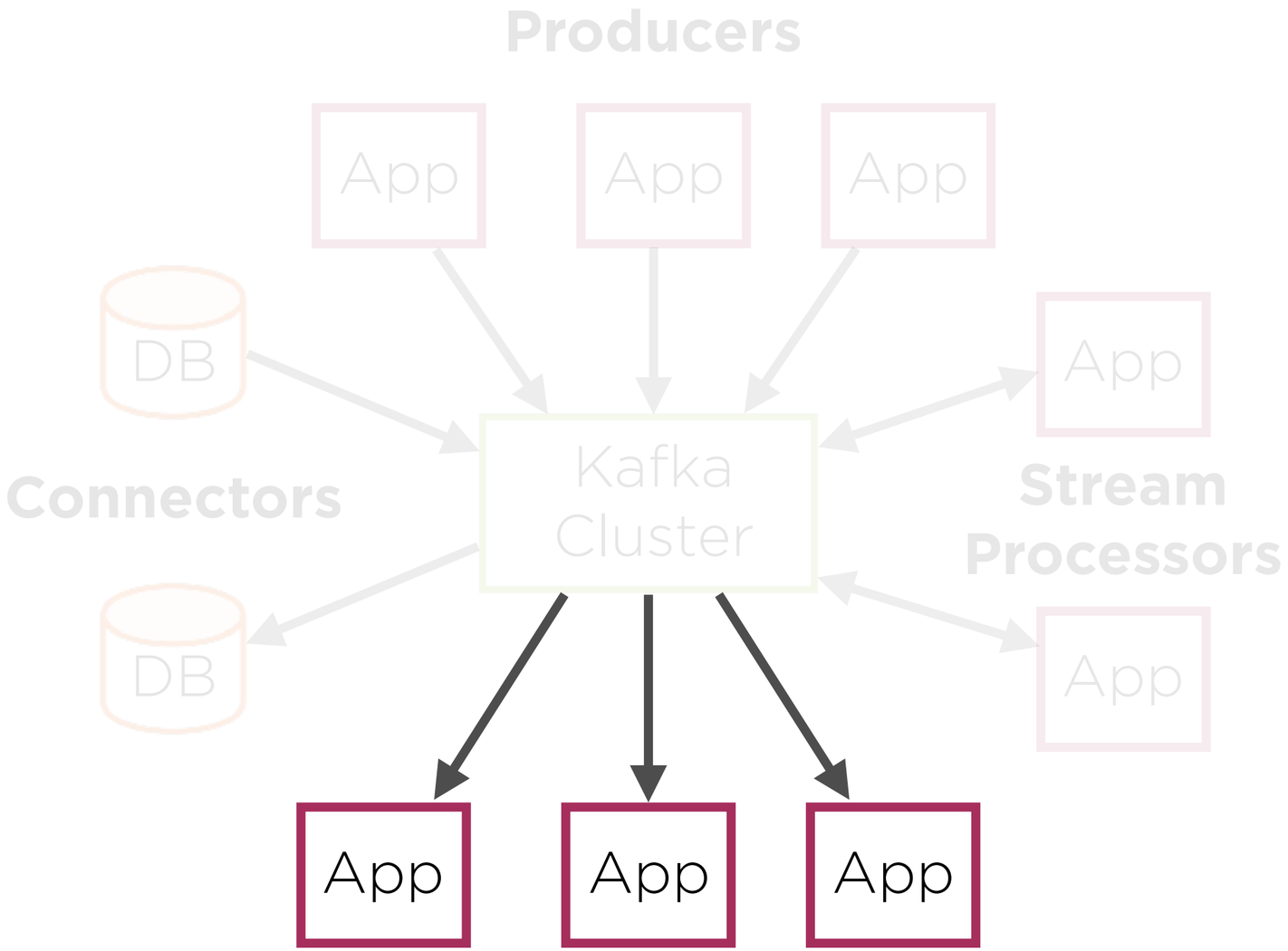

Producers
App
App
App
DB
DB
Connectors
Kafka
Cluster
App
Stream
Processors
App
App
App
App
Consumers

# Kafka APIs

**Producers**

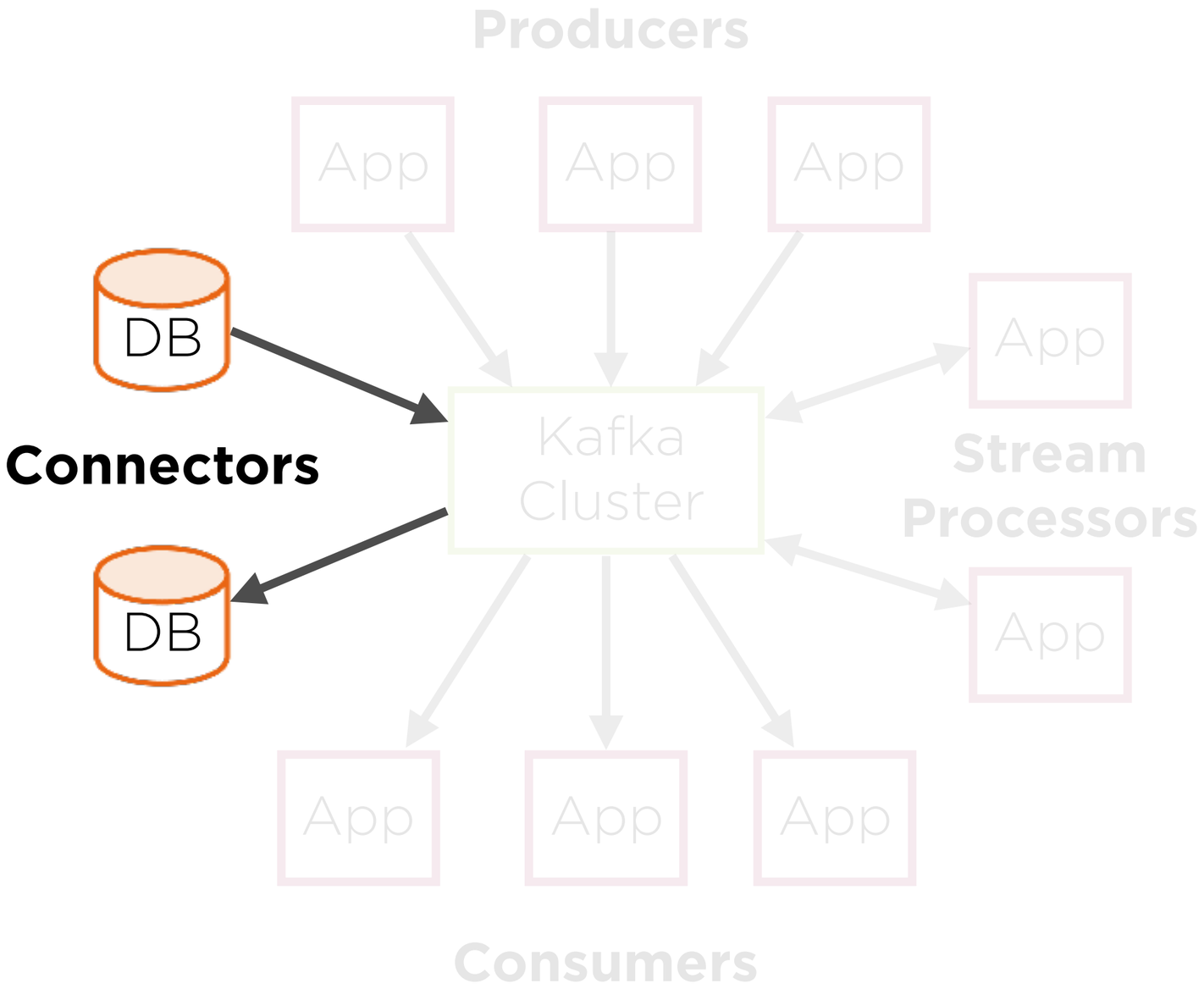


# Kafka APIs

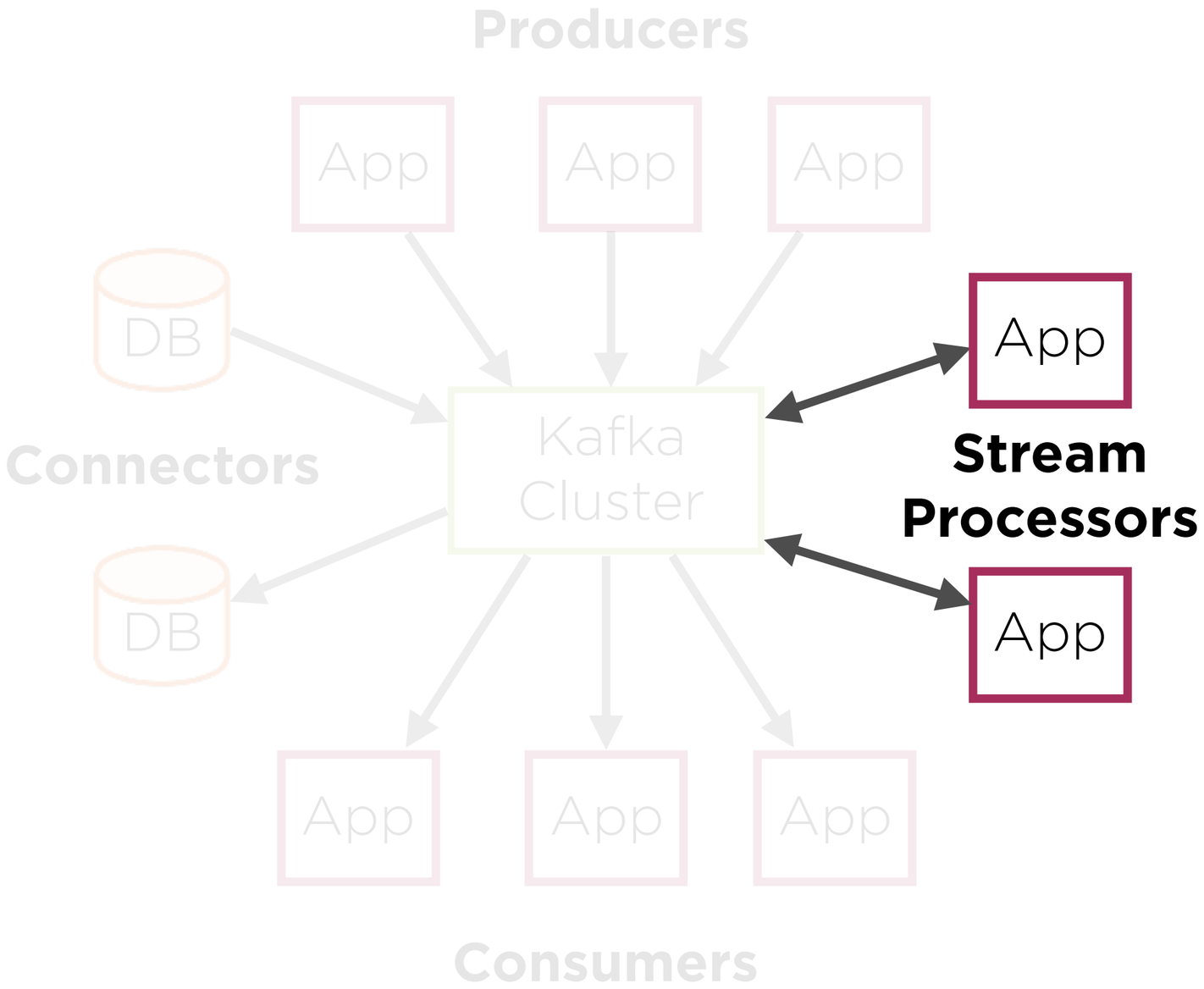


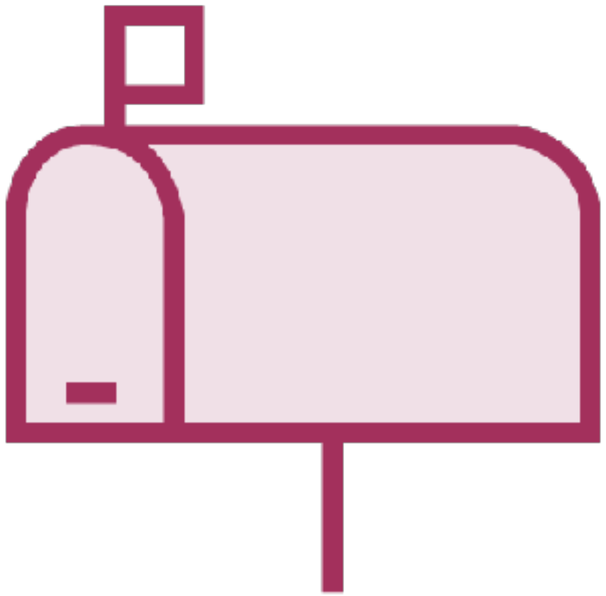

**Consumers**

# Kafka APIs


Producers
App
App
App
DB
Connectors
DB
Kafka
Cluster
App
Stream
Processors
App
App
App
App
Consumers

# Kafka APIs


Producers
App
App
App
DB
DB
Connectors
Kafka
Cluster
App
Stream
Processors
App
App
App
App
Consumers



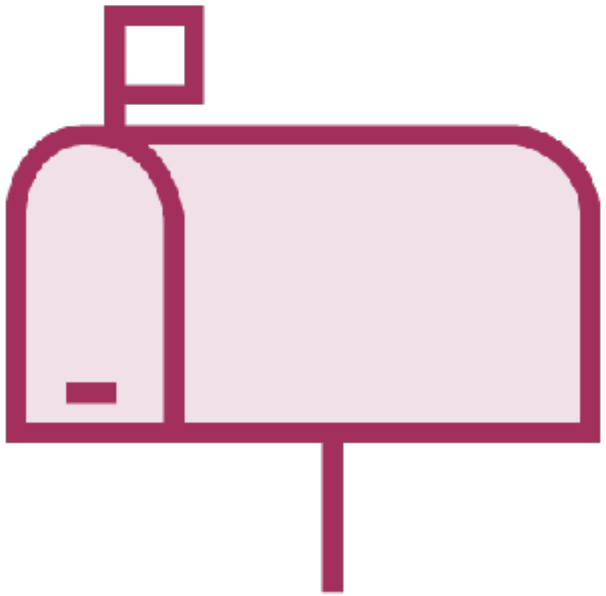
# Capabilities

**Producer API**

**Consumer API**

**Connector API**

**Streams API**



# Features of Kafka

**Scalability**

**Data partitioning**

**Low latency**

**Fan-in and fan-out**

# Kafka

What

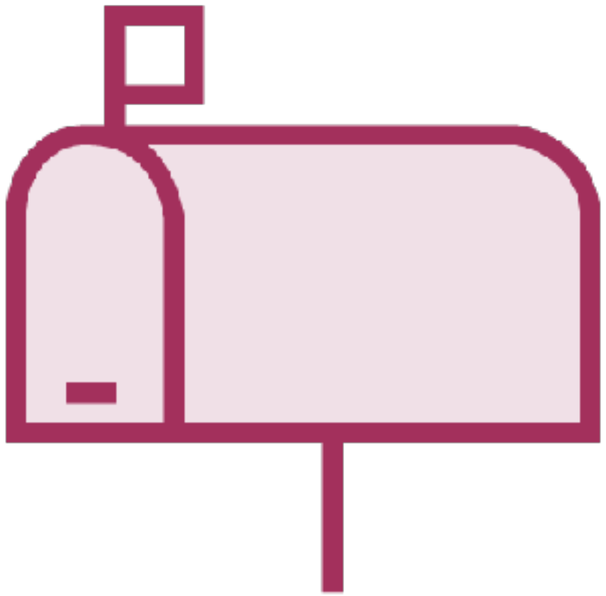
Pub/Sub messaging middleware

How

**Internally uses Zookeeper, partitioning**

Why

Distributed, scalable, low-latency

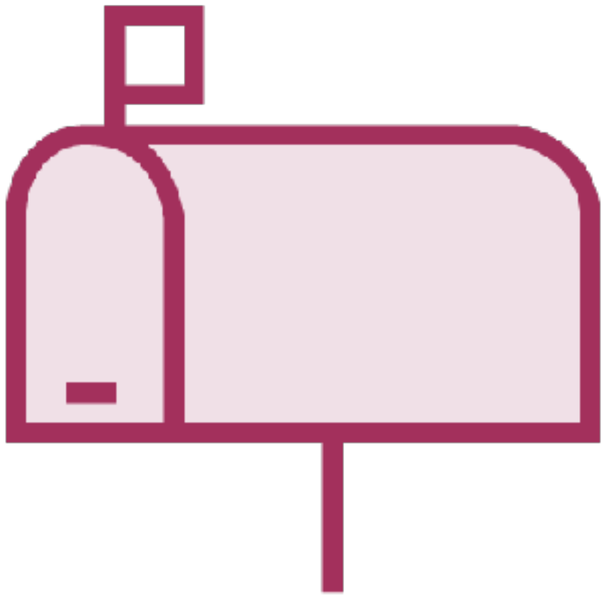


# Internals

**Distributed - spans servers and physical infra**

**Uses Zookeeper for high availability coordination**

**Use MirrorMaker for geo-replication**



# Internals

**Streams of records**

**Categorized into topics**

**Each record has**

- key
- value
- timestamp

# Demo

**Basic introduction to Kafka producers and consumers**

# Demo

**Use a Kafka producer to stream tweets**

# Demo

**Use a Kafka producer with tweets**

**Analyze tweet sentiment using Afinn**

# Demo

Use a Kafka producer with tweets

Count the number of positive and negative tweets

# Summary

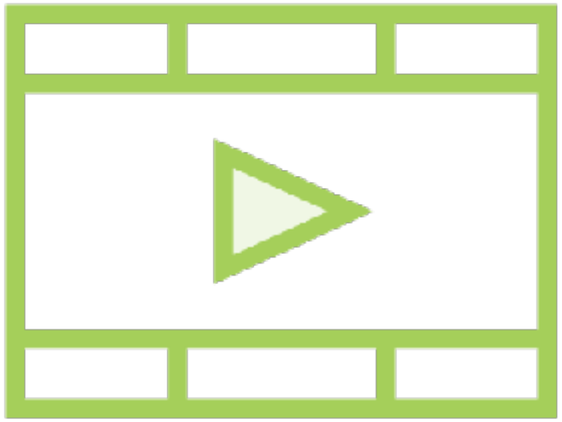
Kafka is a powerful publisher/subscriber messaging technology

Producers publish, consumers subscribe

Messages are categorized by topic and stored in partitioned, replicated logs

Kafka is distributed and uses Zookeeper internally

Structured Streaming and Kafka interface in powerful ways



# Related Courses

## Handling Fast Data with Apache Spark SQL and Streaming

- Spark using Scala

## Building Machine Learning Models in Spark 2

- Spark ML library in Python

## Getting Started with Apache Kafka