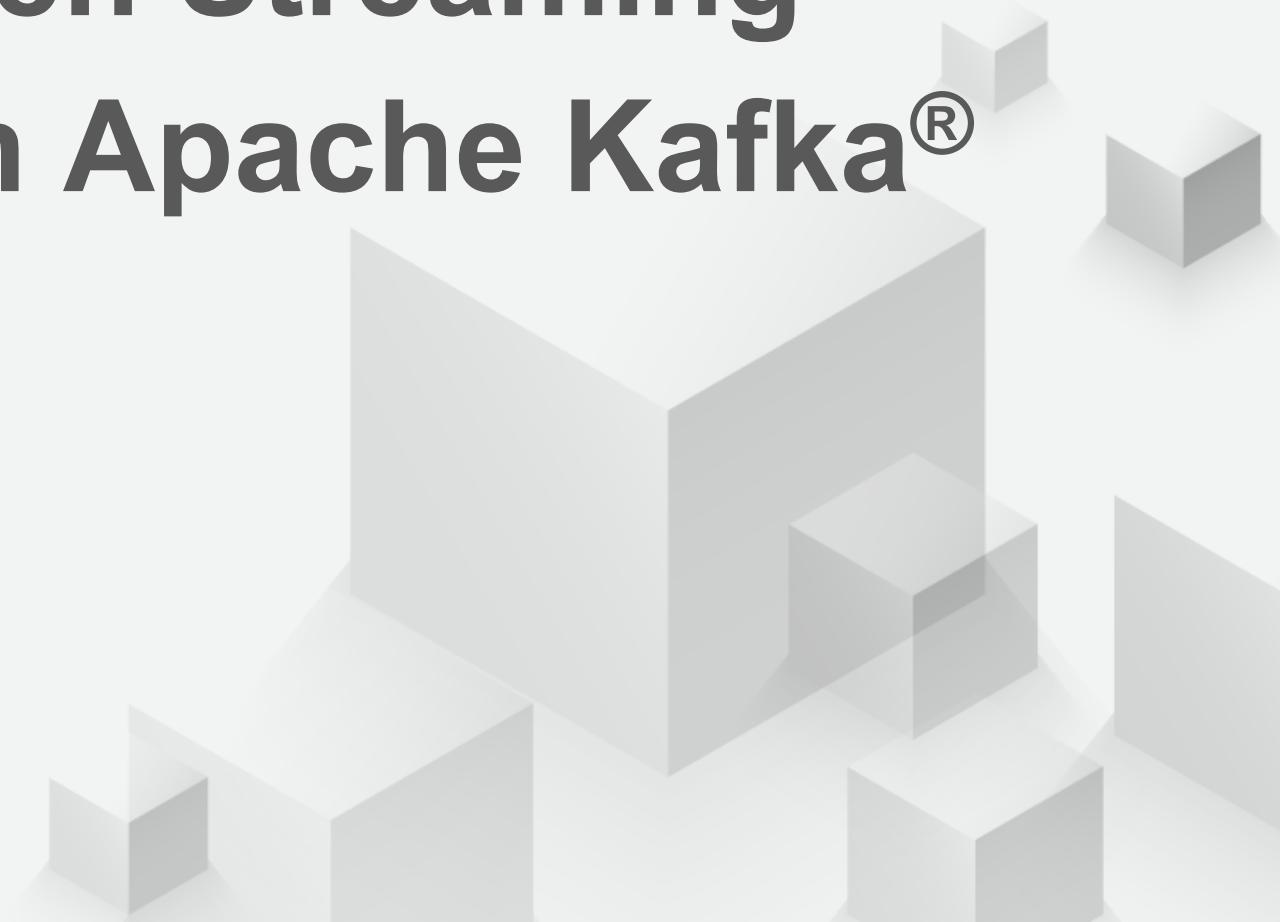




# **Building ML-Driven Streaming Applications with Apache Kafka®**

A subtle, abstract graphic in the background consists of several light gray, 3D-rendered cubes of various sizes and orientations, some casting soft shadows, which create a sense of depth and data flow across the slide.



## Joseph Moraes

AWS Evangelist/Cloud Partner Solutions Architect, Confluent  
theejosephmoraes [in](#)



## Kanchan Waikar

Senior Partner Solutions Architect at AWS  
kanchanwaikar [in](#)





**Joseph Moraes**

AWS Evangelist/Cloud Partner Solutions Architect, Confluent  
theejosephmoraes

# Event Streaming

And why you might care

# Every Industry is Moving from Batch/Manual to Software-Defined

	Software-using	Software-defined
<b>Auto / Transport</b>	Spreadsheet-driven driver schedule	Real-time ETA
<b>Banking</b>	Nightly credit-card fraud checks	Real-time credit card fraud prevention
<b>Retail</b>	Batch inventory updates	Real-time inventory management
<b>Healthcare</b>	Batch claims processing	Real-time claims processing
<b>Oil and Gas</b>	Batch analytics	Real-time analytics
<b>Manufacturing</b>	Scheduled equipment maintenance	Automated, predictive maintenance
<b>Defense</b>	Reactive cyber-security forensics	Automated SIEM and Anomaly Detection



JPMORGAN CHASE & CO.



Morgan Stanley



U.S. Defense Agencies

# Becoming Software-Defined is a Competitive Requirement

By 2020 event-sourced, real-time situational awareness will be a required characteristic for 80% of digital business solutions. And 80% of new business ecosystems will require support for event processing."

- Gartner

# Data Platform Requirements for Becoming Software-Defined

Software-using

Built for Historical Data

Scalable for Transactional Data

Transient

Raw data

Software-defined

① Built for Real-Time Events

- State vs. change
- Historical analysis vs. real-time operations

② Scalable for ALL data

- Non-transactional data is 10x transactional data
- IoT, logs, security events...

③ Persistent + Durable

- Mission critical apps require zero data loss
- Mission critical systems require replay

④ Enriched data

- Stream processing (SQL on RT events)
- Context & situational awareness (ex. ETA)

# Only Event Streaming Has All 4 Requirements

	BUILT FOR REAL-TIME EVENTS	SCALABLE FOR ALL DATA	PERSISTENT & DURABLE	CAPABLE OF ENRICHMENT
Databases		Good for transactional applications		
Messaging		Good for ultra low-latency, fire-and-forget use cases		
ETL		Good for batch data integration		
Data Warehouse		Good for historical analytics and reporting		
Event Streaming		<b>The Essential Data Platform for Becoming Software-Defined</b> (Scalable Messaging + Real-Time Data Integration + Stream Processing)		

# The Rise of Event Streaming

60%

Fortune 100 Companies  
Using Apache Kafka



GoPro

stripe

airbnb



fitbit

box

WIKIPEDIA  
the Free Encyclopedia

pinterest

uber

# What or who is Kafka?

The Append-only log



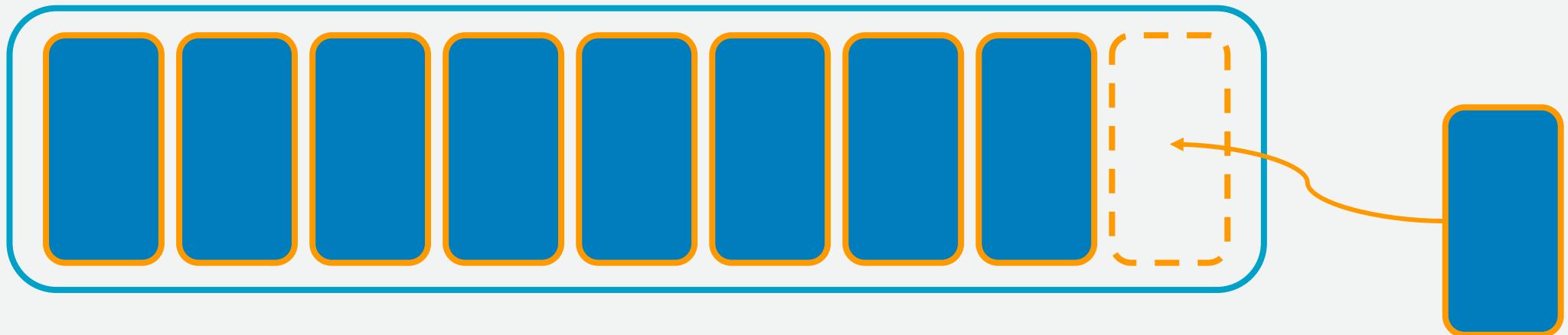


© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

 |  aws marketplace



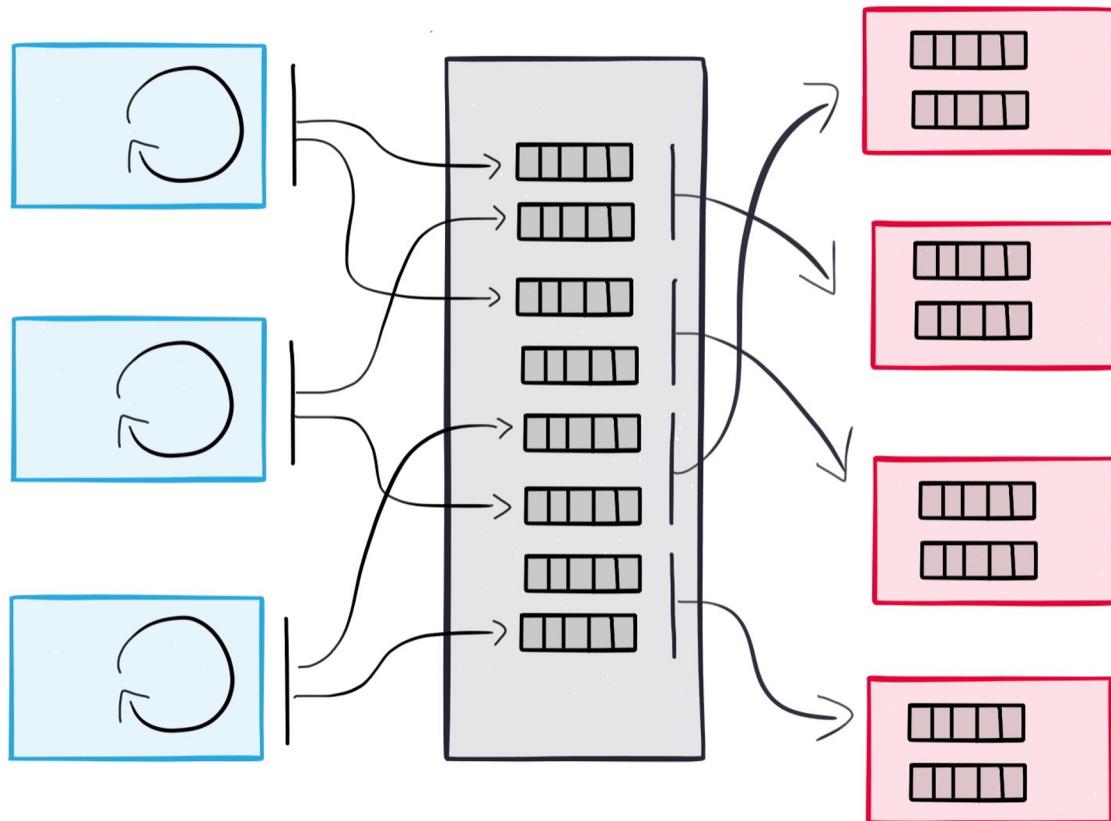
# Both Kafkas like to write



# Writers

# Kafka cluster

# Readers



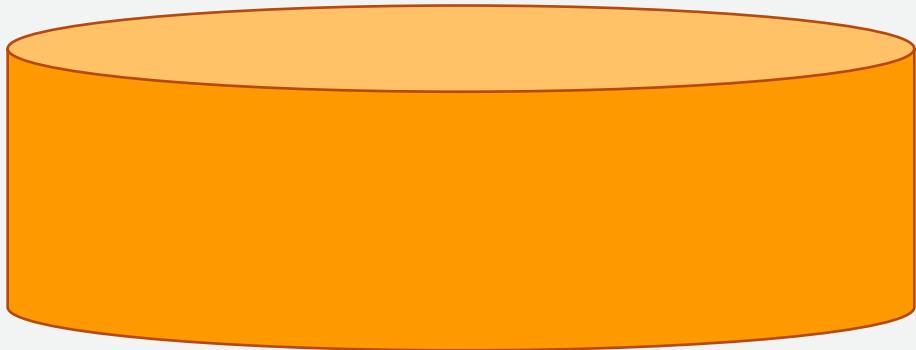
# Scalability of a filesystem

- Hundreds of MB/s throughput
- Many TB per server
- Commodity hardware



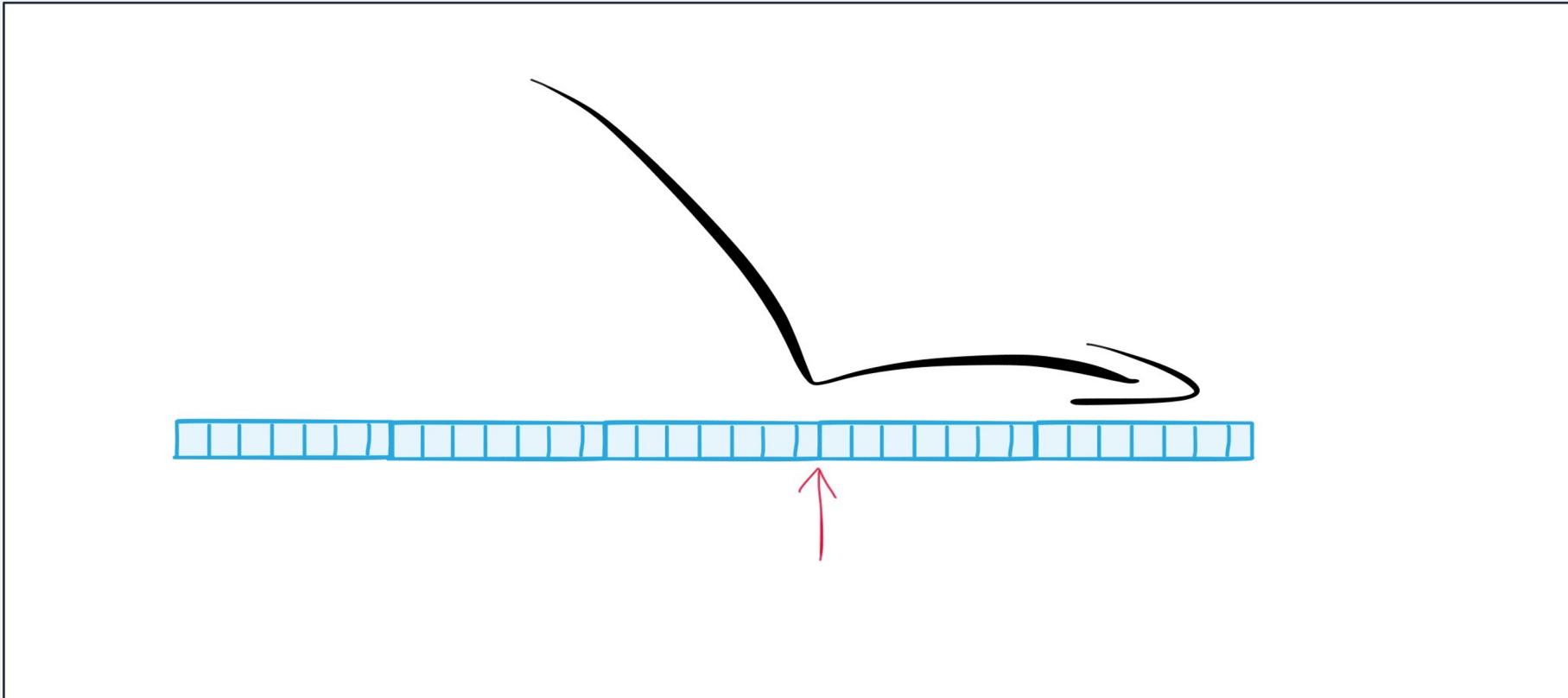
# Guarantees of a Database

- Strict ordering
- Persistence



# Rewind & Replay

Reset to any point in the shared narrative



Rewind & Replay

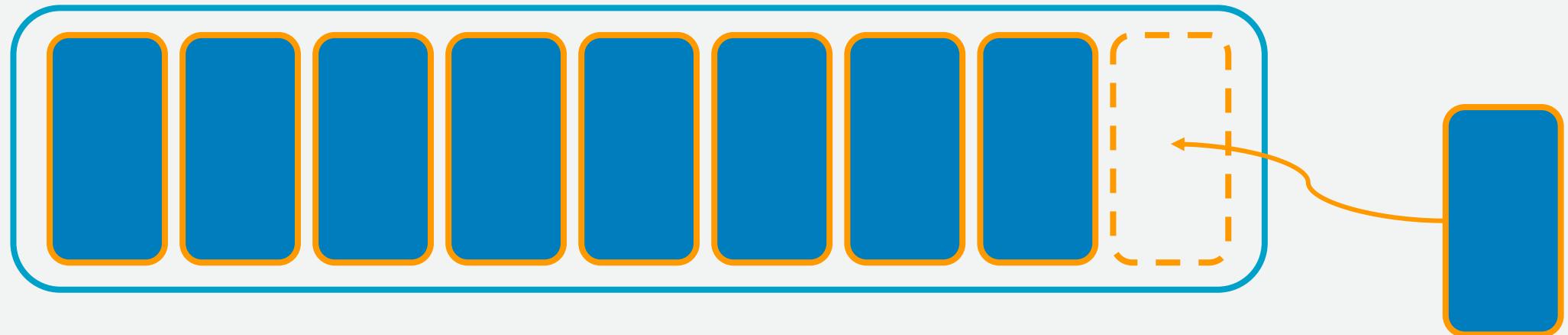


# Distributed by design

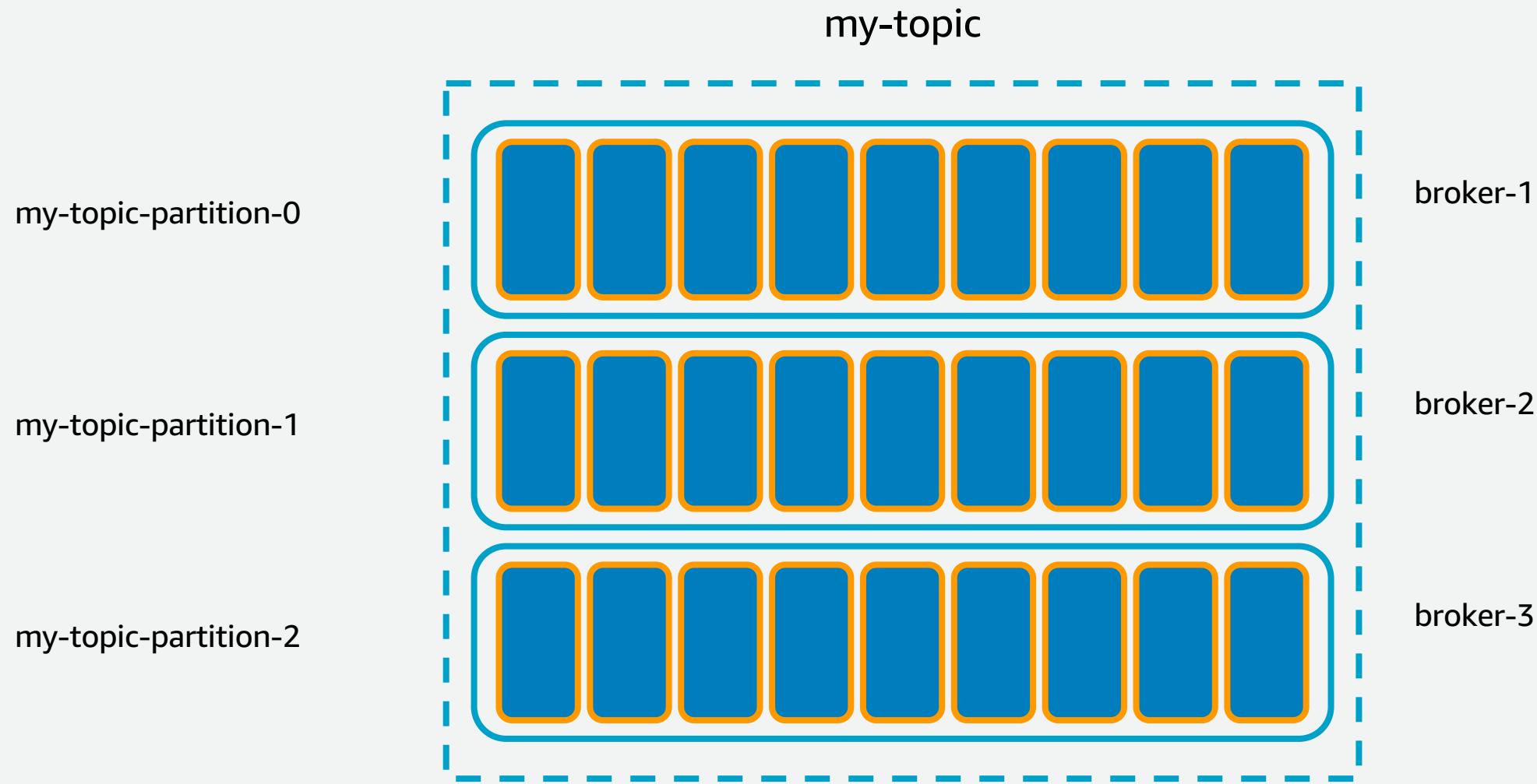
- Replication
- Fault Tolerance
- Partitioning
- Elastic Scaling

# Producing to Kafka

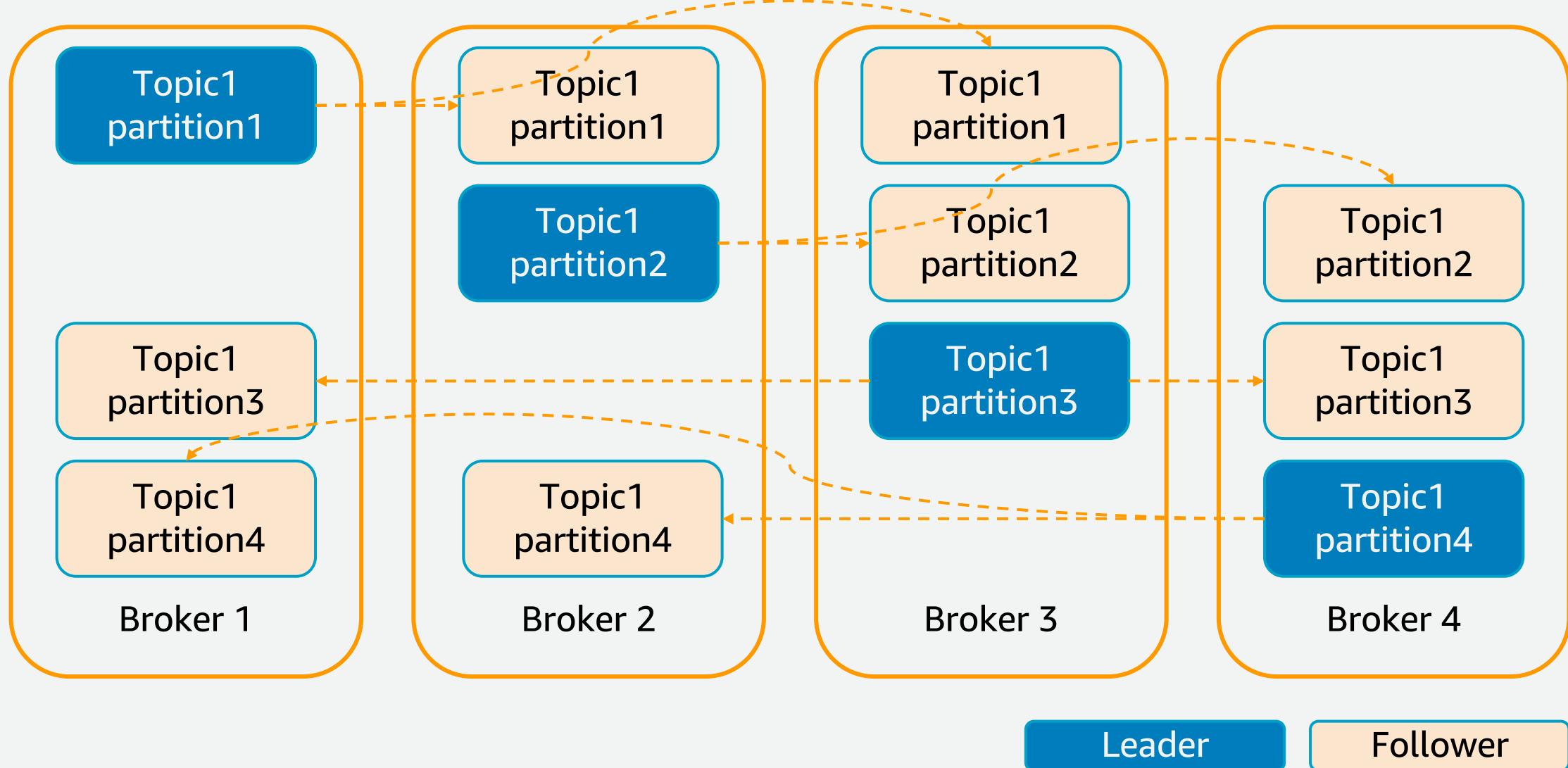
Time →



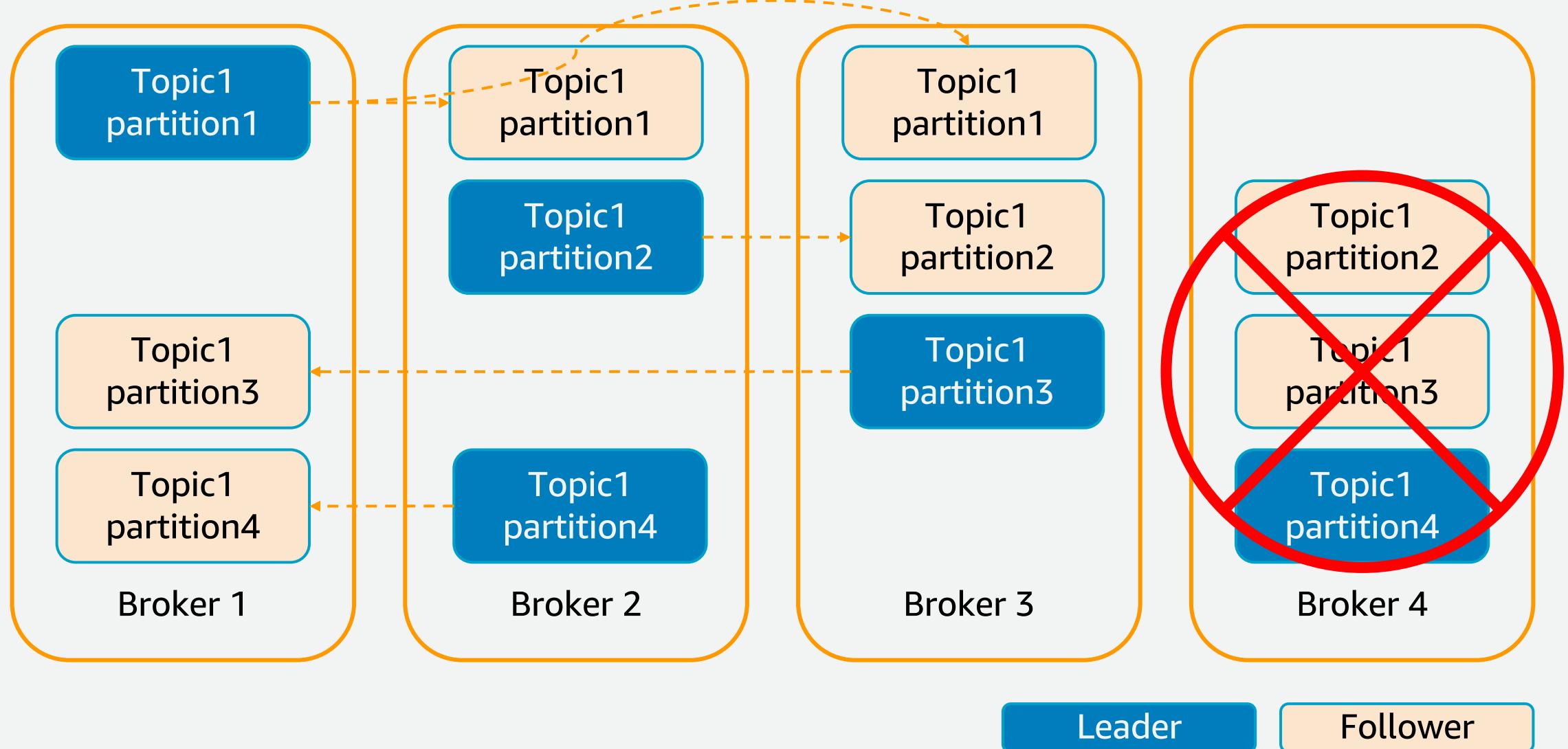
# Kafka Topics

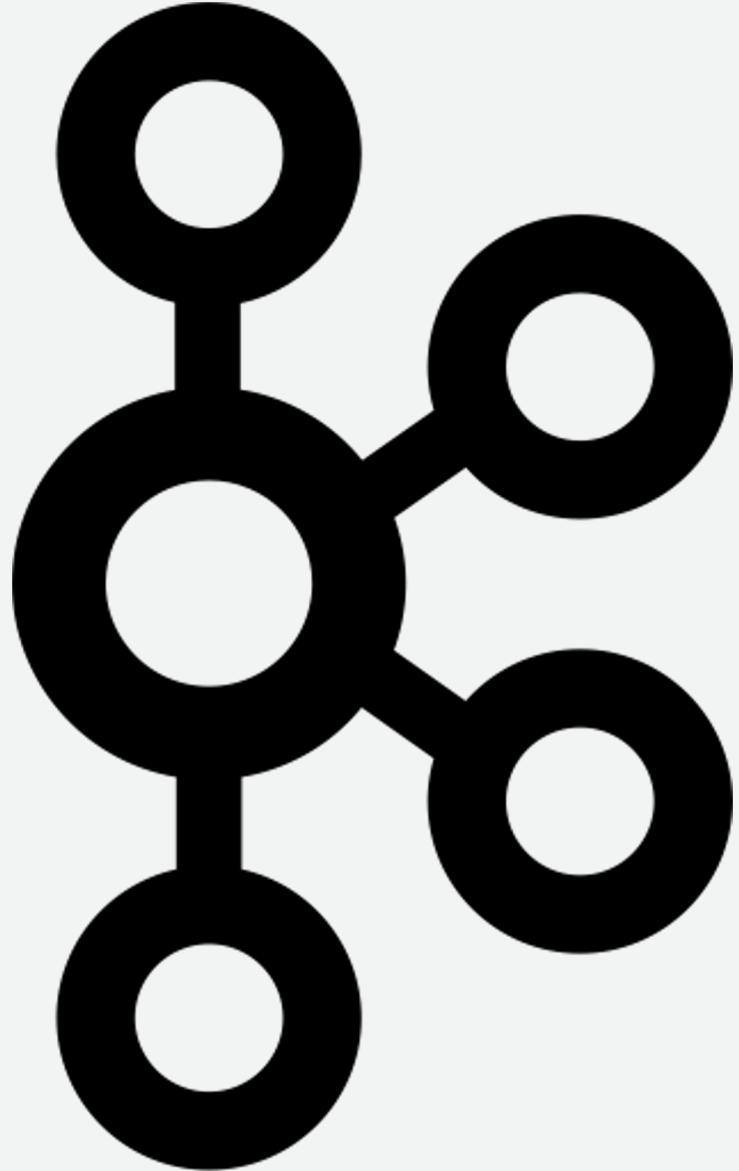


# Partition Leadership and Replication



# Partition Leadership and Replication - node failure

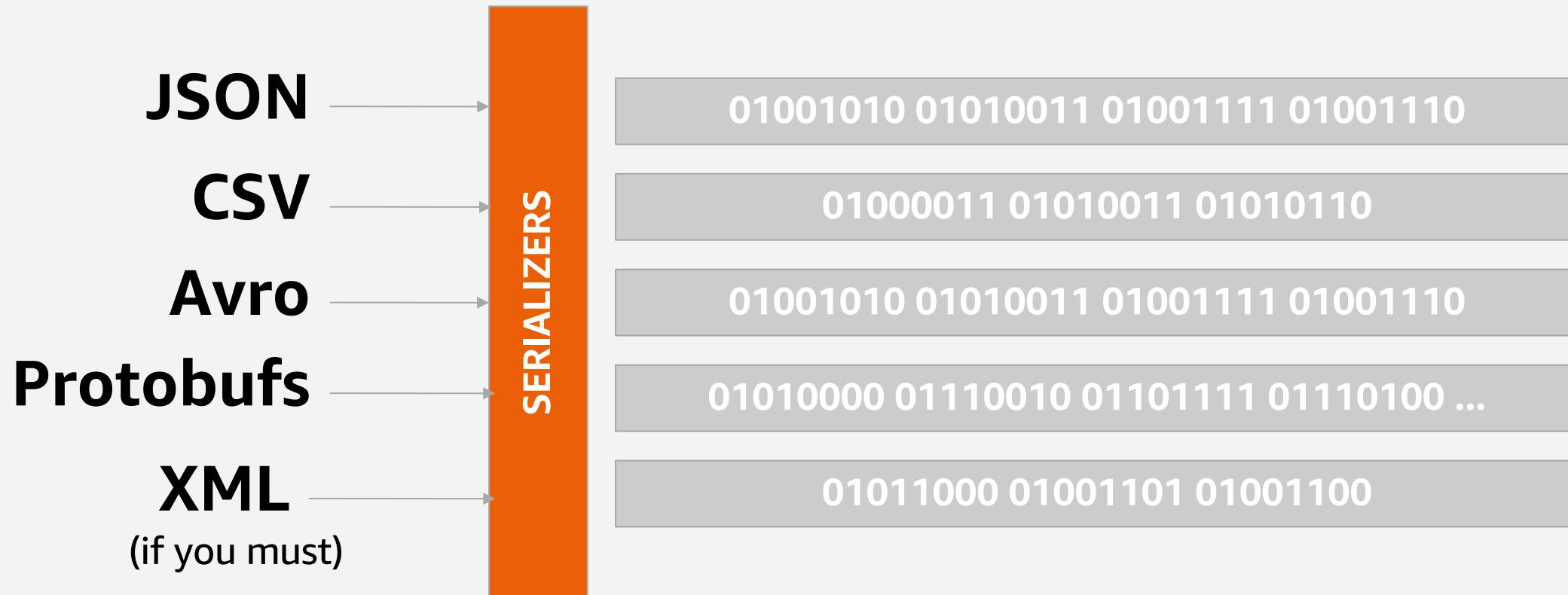




# Producing to Kafka

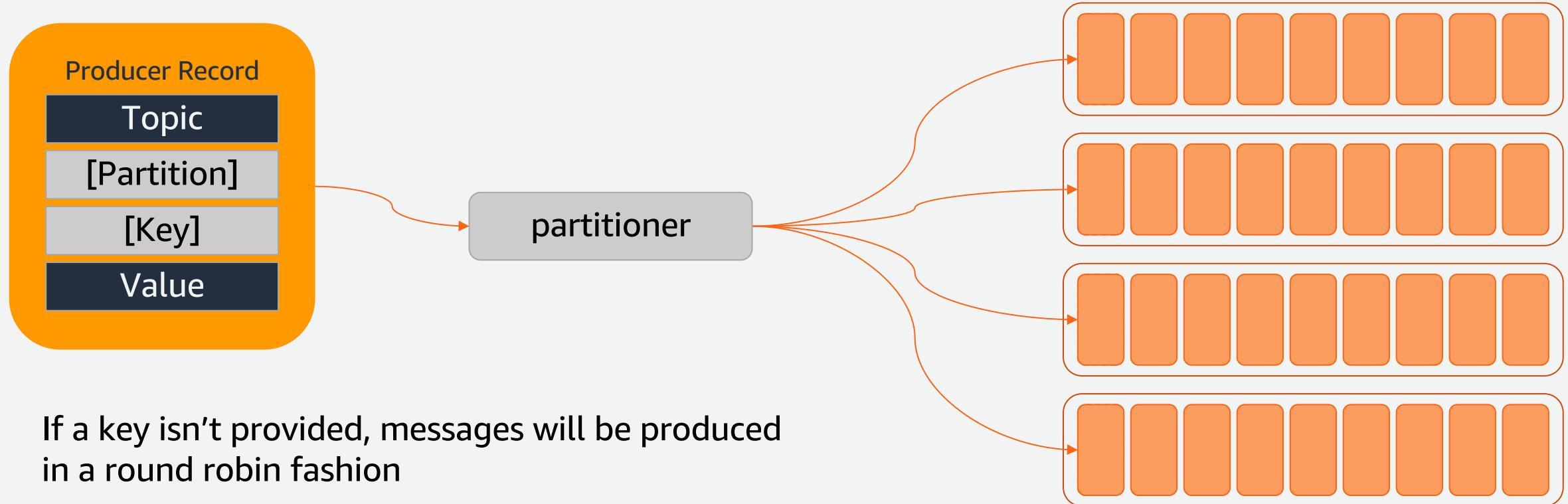
# The Serializer

Kafka doesn't care about what you send to it as long as it's been converted to a byte stream beforehand.



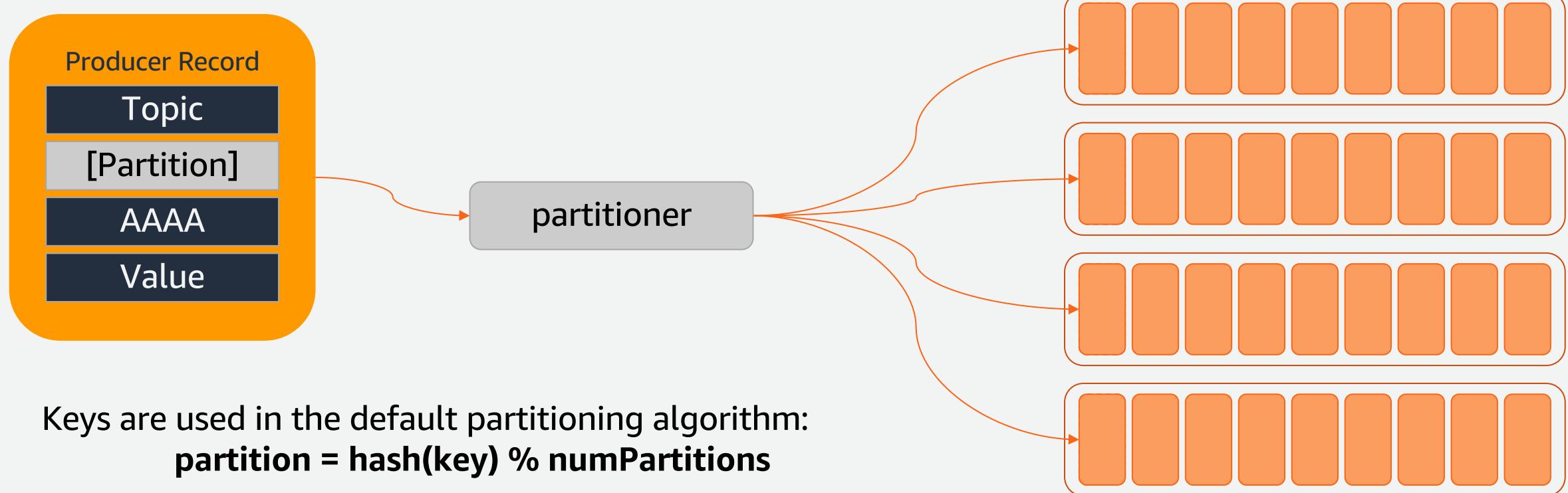
# Record Keys and why they're important - Ordering

Record keys determine the partition with the default Kafka partitioner



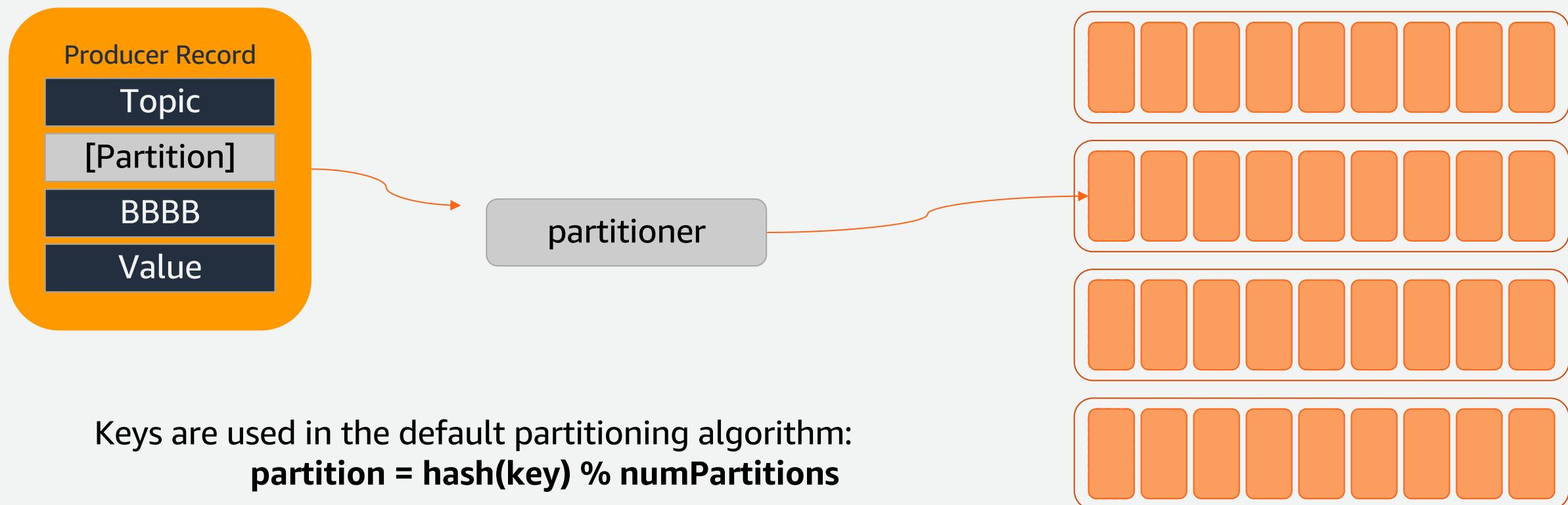
# Record Keys and why they're important - Ordering

Record keys determine the partition with the default Kafka partitioner, and therefore guarantee order for a key



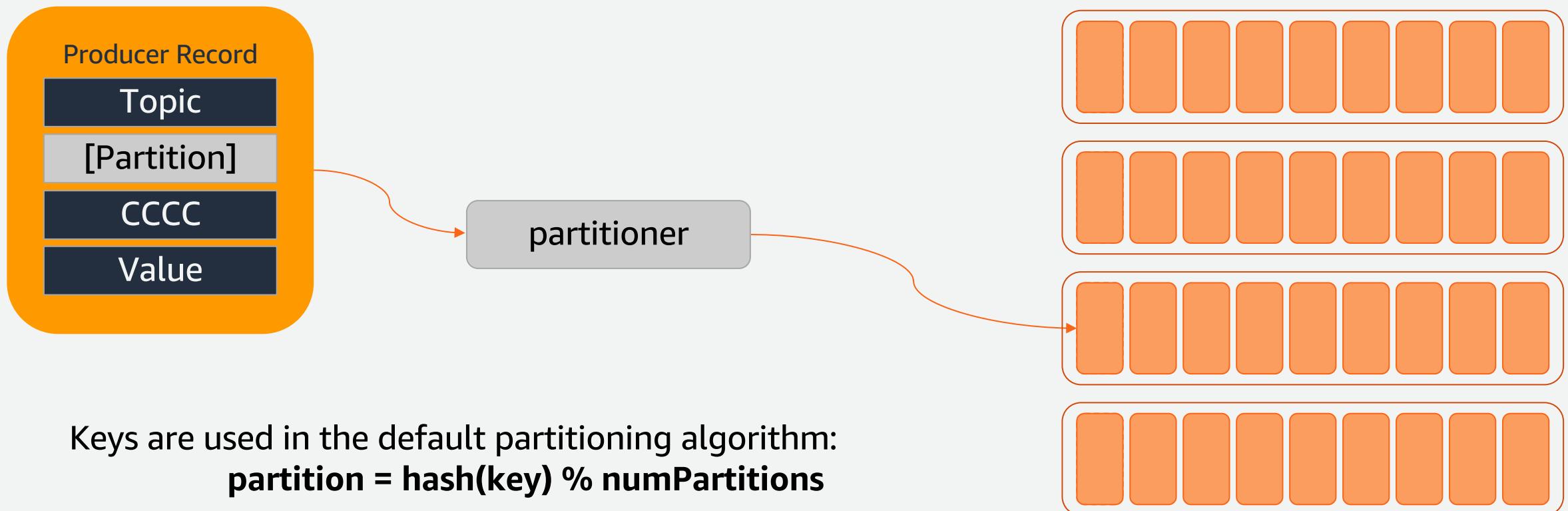
# Record Keys and why they're important - Ordering

Record keys determine the partition with the default Kafka partitioner, and therefore guarantee order for a key



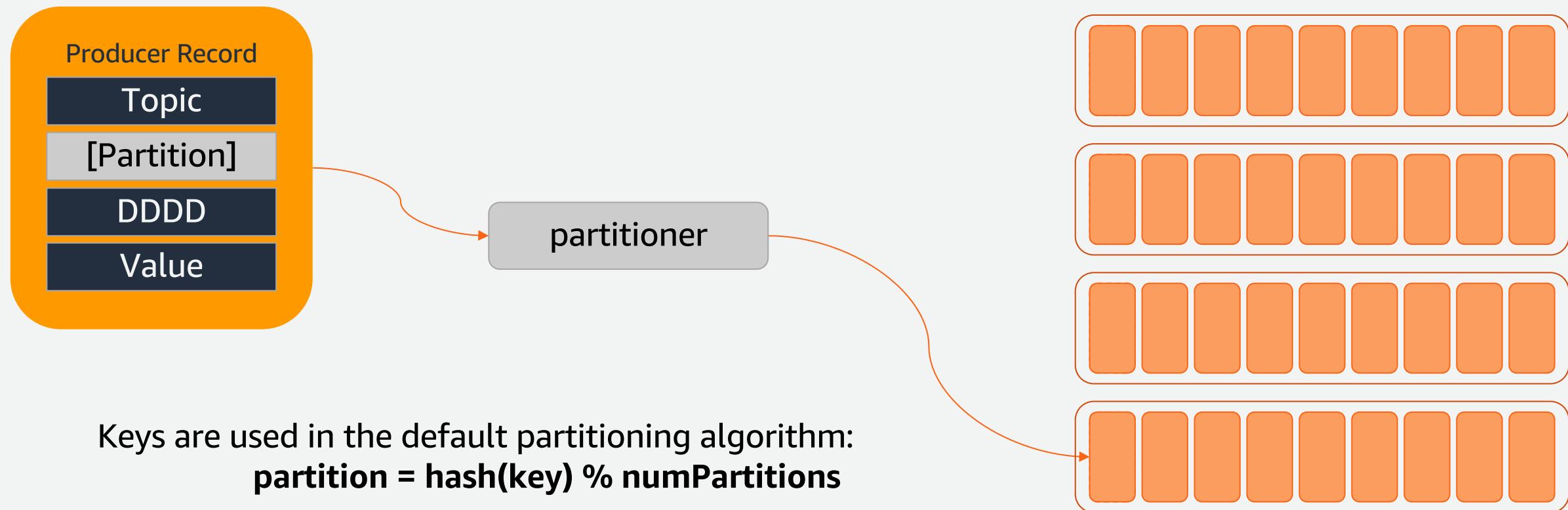
# Record Keys and why they're important - Ordering

Record keys determine the partition with the default Kafka partitioner, and therefore guarantee order for a key

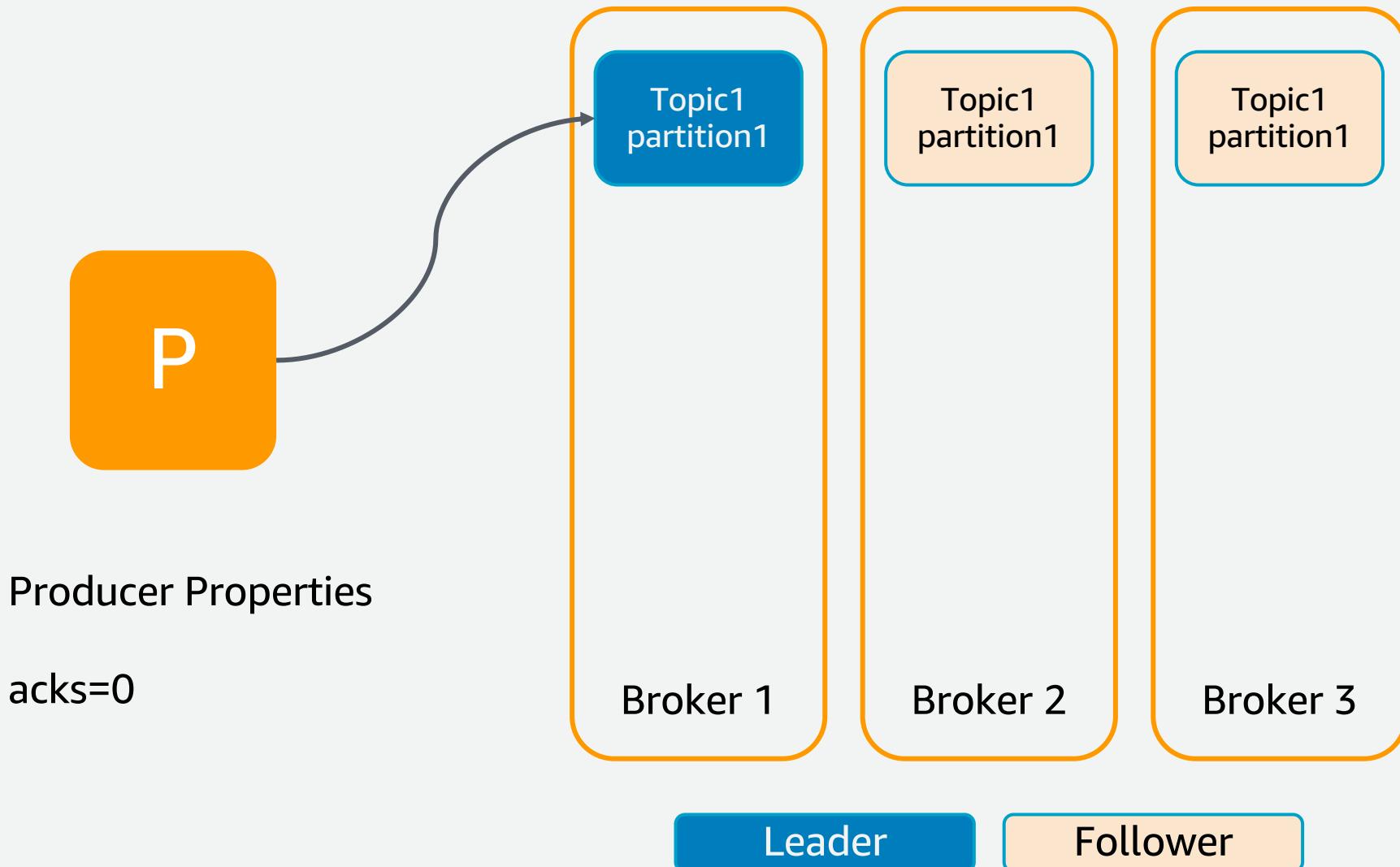


# Record Keys and why they're important - Ordering

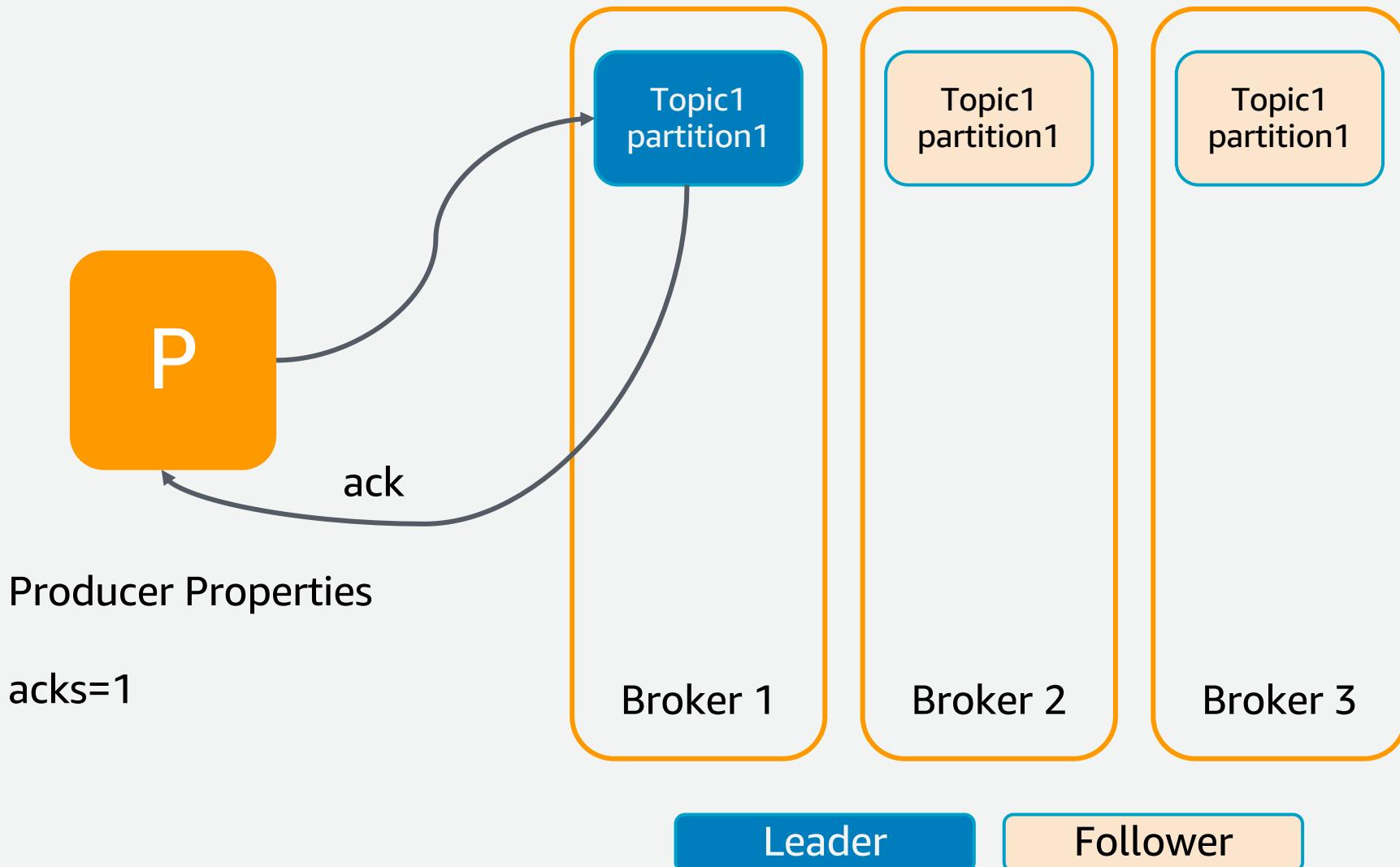
Record keys determine the partition with the default Kafka partitioner, and therefore guarantee order for a key



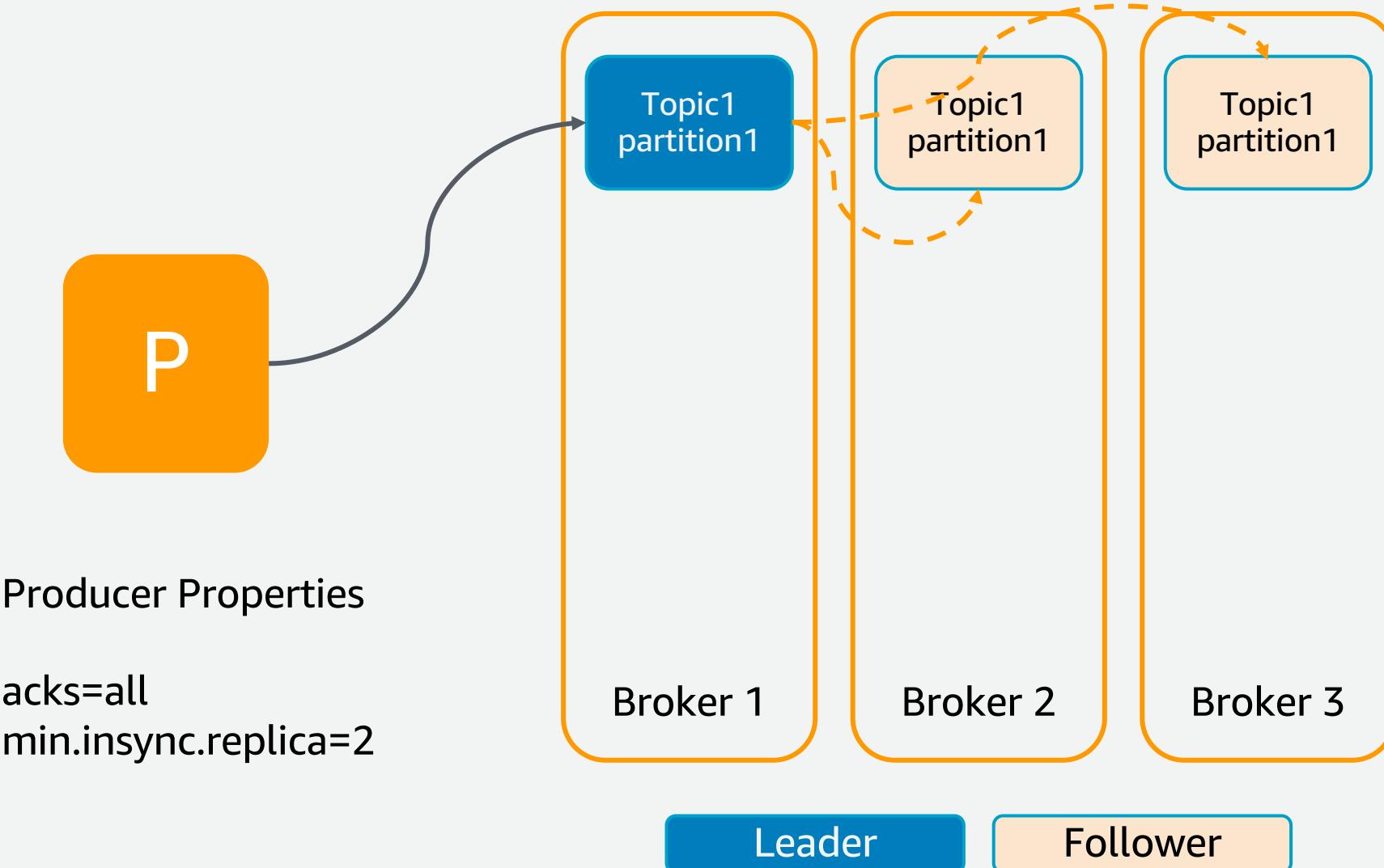
# Producer Guarantees

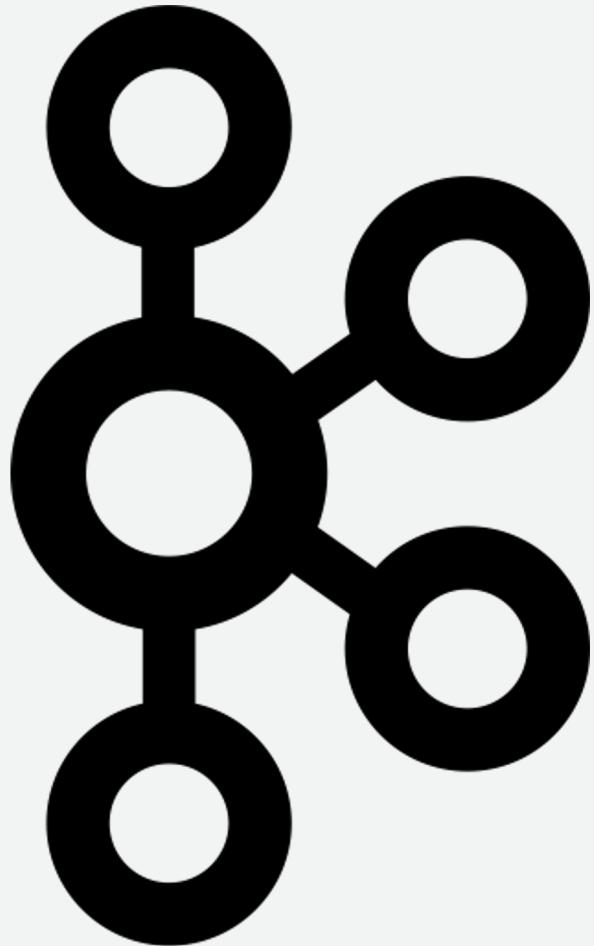


# Producer Guarantees



# Producer Guarantees

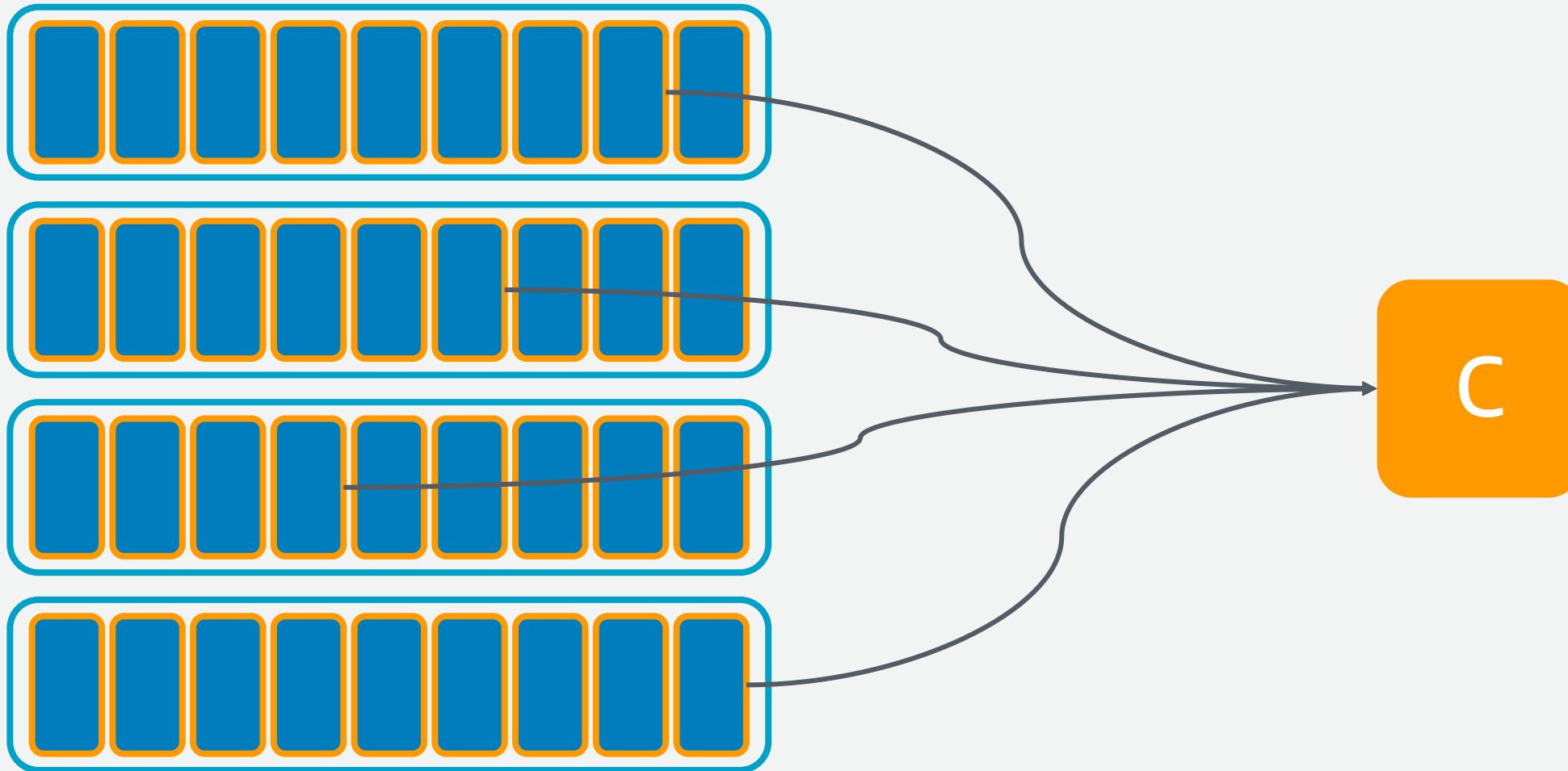




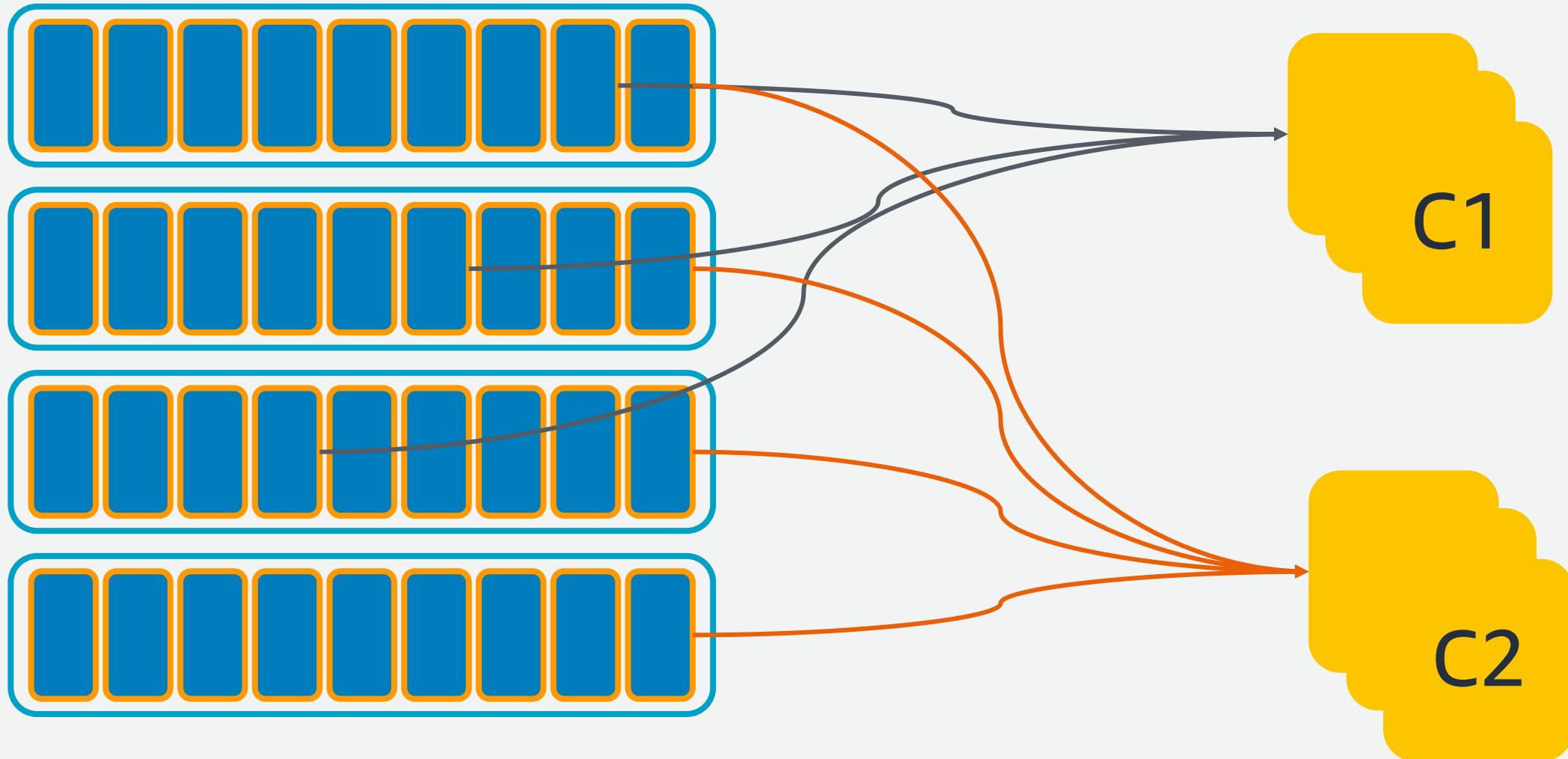
# Consuming from Kafka

Consuming together is better

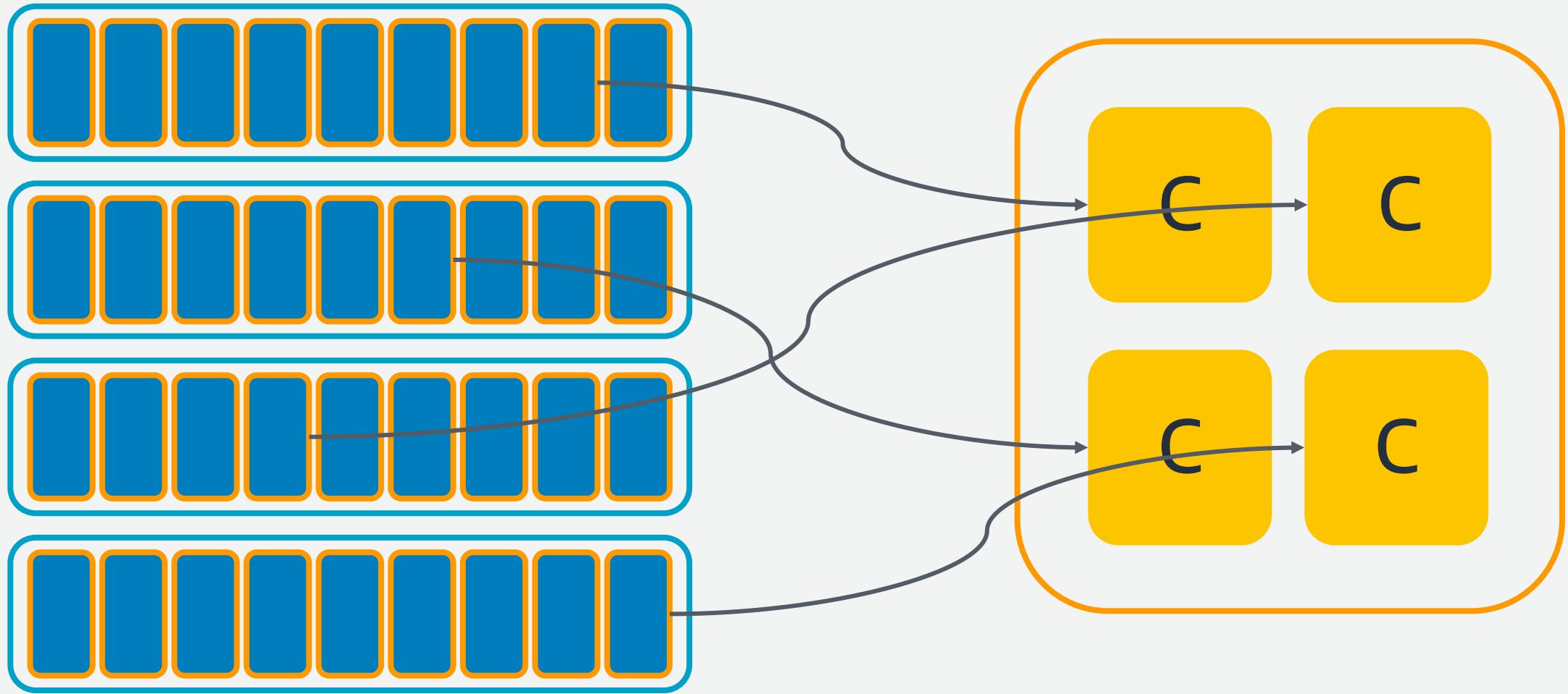
# Consuming From Kafka - Single Consumer



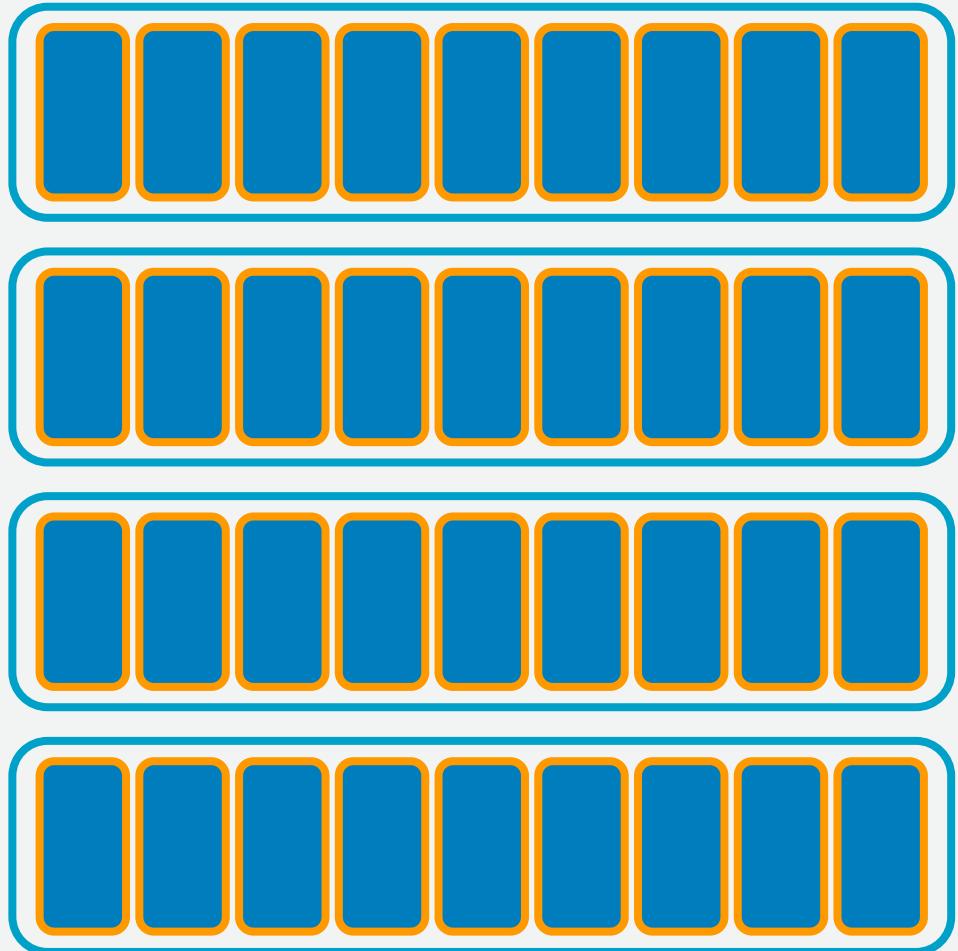
# Consuming From Kafka - Grouped Consumer



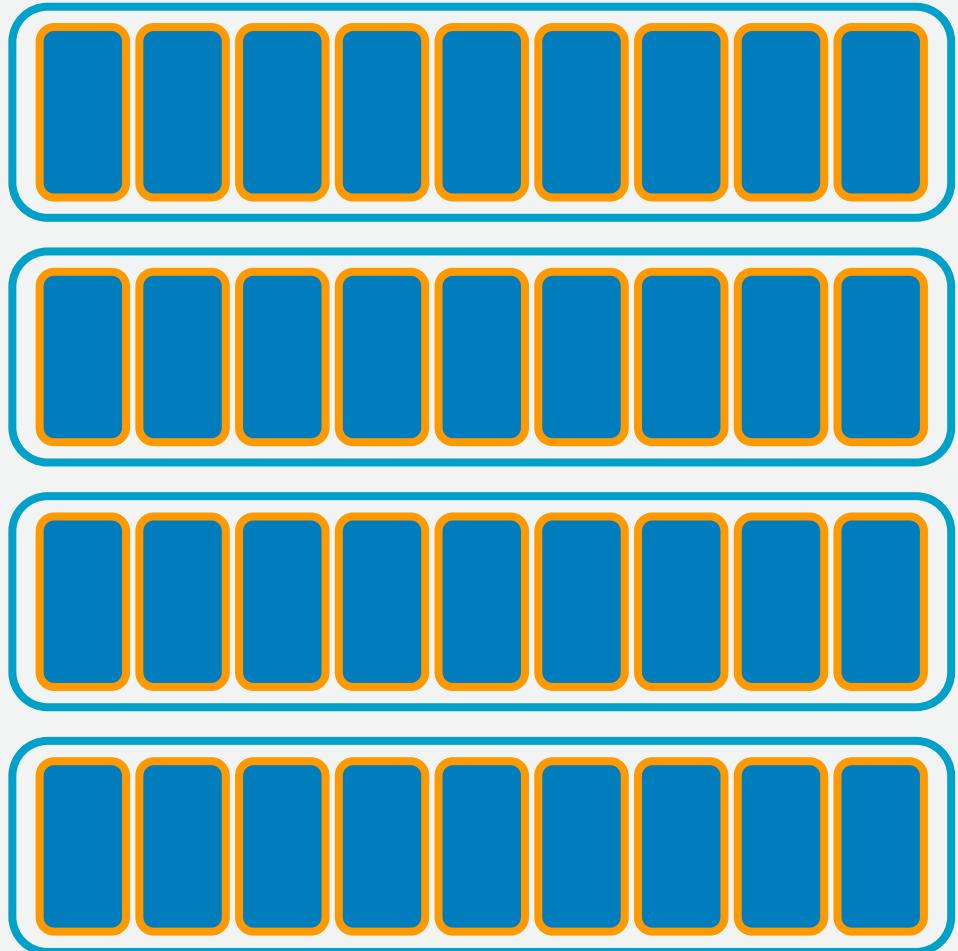
# Consuming From Kafka - Grouped Consumer



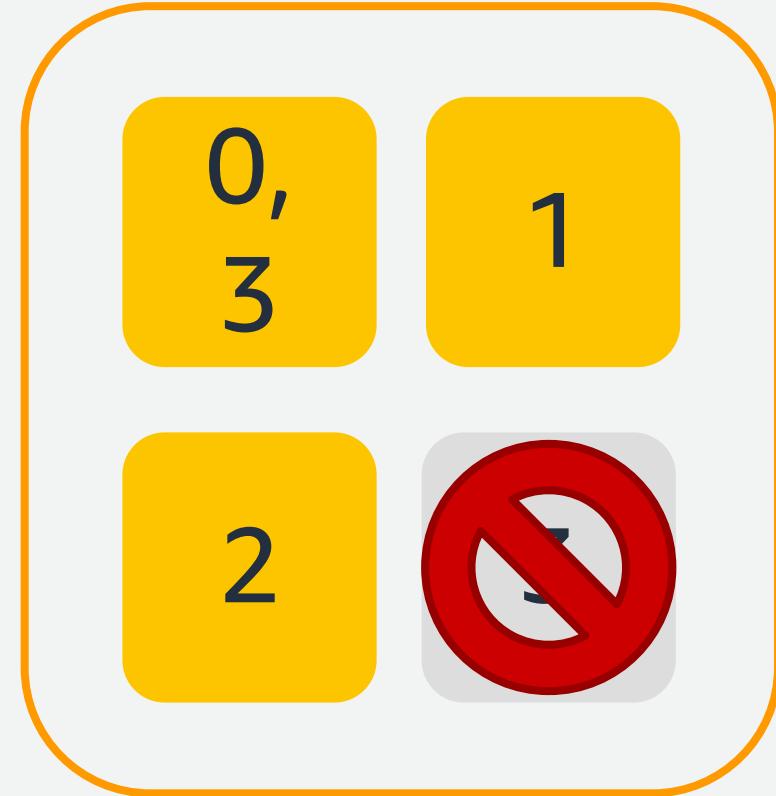
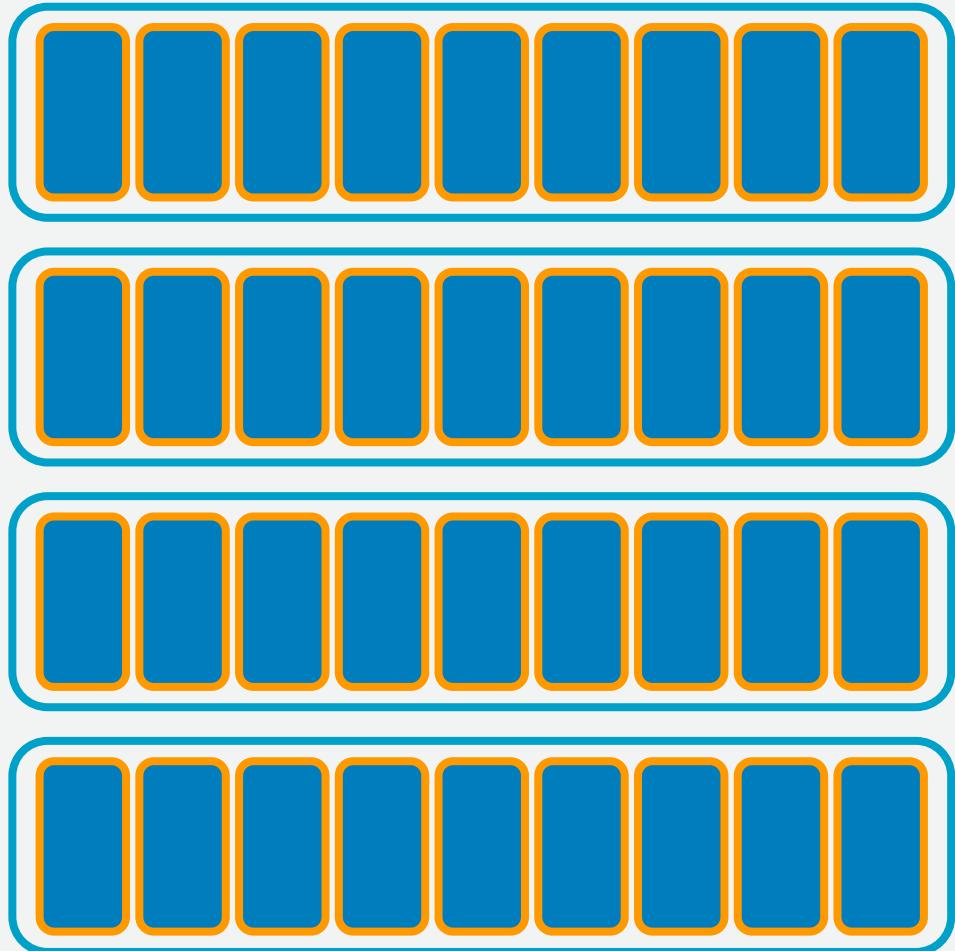
# Consuming From Kafka - Grouped Consumer



# Consuming From Kafka - Grouped Consumer



# Consuming From Kafka - Grouped Consumer

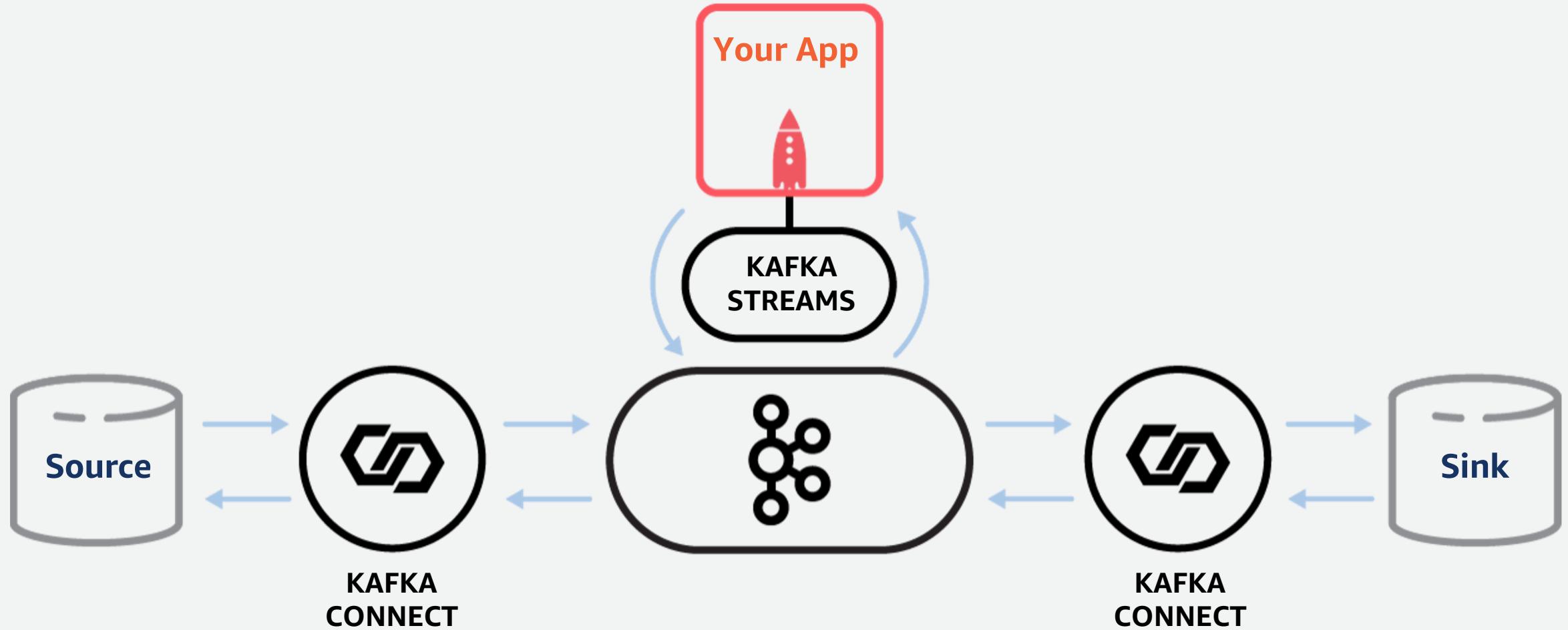


# Kafka APIs

Making Kafka an event-streaming platform



# Kafka Connect and Kafka Streams



# Kafka Connect: Reliable and scalable integration of Kafka with other systems

- Centralized management and configuration
- Support for hundreds of technologies including RDBMS, Elasticsearch, HDFS, S3
- Supports CDC ingest of events from RDBMS
- Preserves data schema
- Fault tolerant and automatically load balanced
- Extensible API
- Single Message Transforms
- Part of Apache Kafka

```
{  
  "connector.class": "io.confluent.connect.jdbc.JdbcSourceConnector",  
  "connection.url": "jdbc:mysql://localhost:3306/demo?user=rmoff&password=foo",  
  "table.whitelist": "sales,orders,customers"  
}
```

# Kafka Streams: Write standard Java apps and microservices to process your data in real-time

- No separate processing cluster required
- Develop on Mac, Linux, Windows
- Deploy to containers, VMs, bare metal, cloud
- Powered by Kafka: elastic, scalable, distributed, battle-tested
- Perfect for small, medium, large use cases
- Fully integrated with Kafka security
- Exactly-once processing semantics
- Part of Apache Kafka

```
KStream<User, PageViewEvent> pageViews = builder.stream("pageviews-topic");
KTable<Windowed<User>, Long> viewsPerUserSession = pageViews
    .groupByKey()
    .count(SessionWindows.with(TimeUnit.MINUTES.toMillis(5)), "session-views");
```



# Stream processing with Kafka

Main Logic {

```
object FraudFilteringApplication extends App {
    val builder: StreamsBuilder = new StreamsBuilder()

    val fraudulentPayments: KStream[String, Payment] = builder
        .stream[String, Payment]("payments-kafka-topic")
        .filter((_, payment) => payment.fraudProbability > 0.8)
    fraudulentPayments.to("fraudulent-payments-topic")

    val config = new java.util.Properties
    config.put(StreamsConfig.APPLICATION_ID_CONFIG, "fraud-filtering-app")
    config.put(StreamsConfig.BOOTSTRAP_SERVERS_CONFIG, "kafka-broker1:9092,kafka-broker2:9092")

    val streams: KafkaStreams = new KafkaStreams(builder.build(), config)
    streams.start()
}
```

Example: Using **Kafka's Streams API** for writing elastic, scalable, fault-tolerant Java and Scala applications

# ksqlDB to the rescue



VIDEO\_ANALYSIS\_ALERTS\_NEW

● Running

```
1 CREATE STREAM VIDEO_ANALYSIS_ALERTS_NEW WITH (KAFKA_TOPIC='pksqlc-gq2orVIDEO_ANALYSIS_ALERTS_NEW', PARTITIONS=3, REPLICAS=3) AS SELECT
2   VIDEO_ANALYSIS.START_TIME START_TIME,
3   VIDEO_ANALYSIS.END_TIME END_TIME,
4   VIDEO_ANALYSIS.PREVIOUS_STATUS PREVIOUS_STATUS
5   FROM VIDEO_ANALYSIS VIDEO_ANALYSIS
6   WHERE (VIDEO_ANALYSIS.PREVIOUS_STATUS LIKE '%ALARM%')
7   EMIT CHANGES;
8
```



**Kanchan Waikar**

Senior Partner Solutions Architect at AWS  
kanchanwaikar



# A real-world streaming use-case

## **Use-case:**

A non-compliance identification and notification system

## **Goal:**

Reduce accidents by identifying and correcting non-compliance

## **Infrastructure-scale:**

Tens of thousands of surveillance cameras, hundreds of construction sites, thousands of construction workers

## **Project Requirement:**

Design an automated PPE non-compliance reporting system that reports incidences in matter of minutes.

## **Technical requirements:**

Must scale up-and-down automatically based on loads (Cameras turned ON and OFF based on shift hours)



# Example video: construction site surveillance



Courtesy - <https://pixabay.com/videos/construction-road-excavator-worker-26239/>

## Summary Logs

(Start) HH:mm:SSS-(End) HH:mm:SSS : **Alarm**/No alarm : Status Details

00:00:000-00:00:015 : No Alarm : 1 truck(s), 1 excavator(s), 1 workers found.

00:00:015-00:00:045 : No Alarm : 2 truck(s), 1 excavator(s), 1 workers found.

00:00:045-00:00:060 : No Alarm : 1 truck(s), 1 excavator(s), 1 workers found.

00:00:060-00:00:075 : No Alarm : 1 truck(s), 1 excavator(s), no workers found.

00:00:075-00:00:090 : **ALARM** : 1 worker(s) wearing PPE but 0 wearing hard hats 1 truck(s), 1 excavator(s) found.

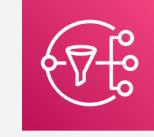
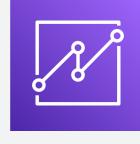
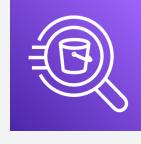
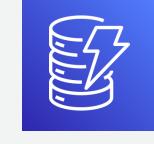
00:00:090-00:00:105 : No Alarm : 1 truck(s), 1 excavator(s), no workers found.

00:00:105-End : **ALARM** : 1 worker(s) wearing PPE but 0 wearing hard hats 1 truck(s), 1 excavator(s) found.

## Snapshots



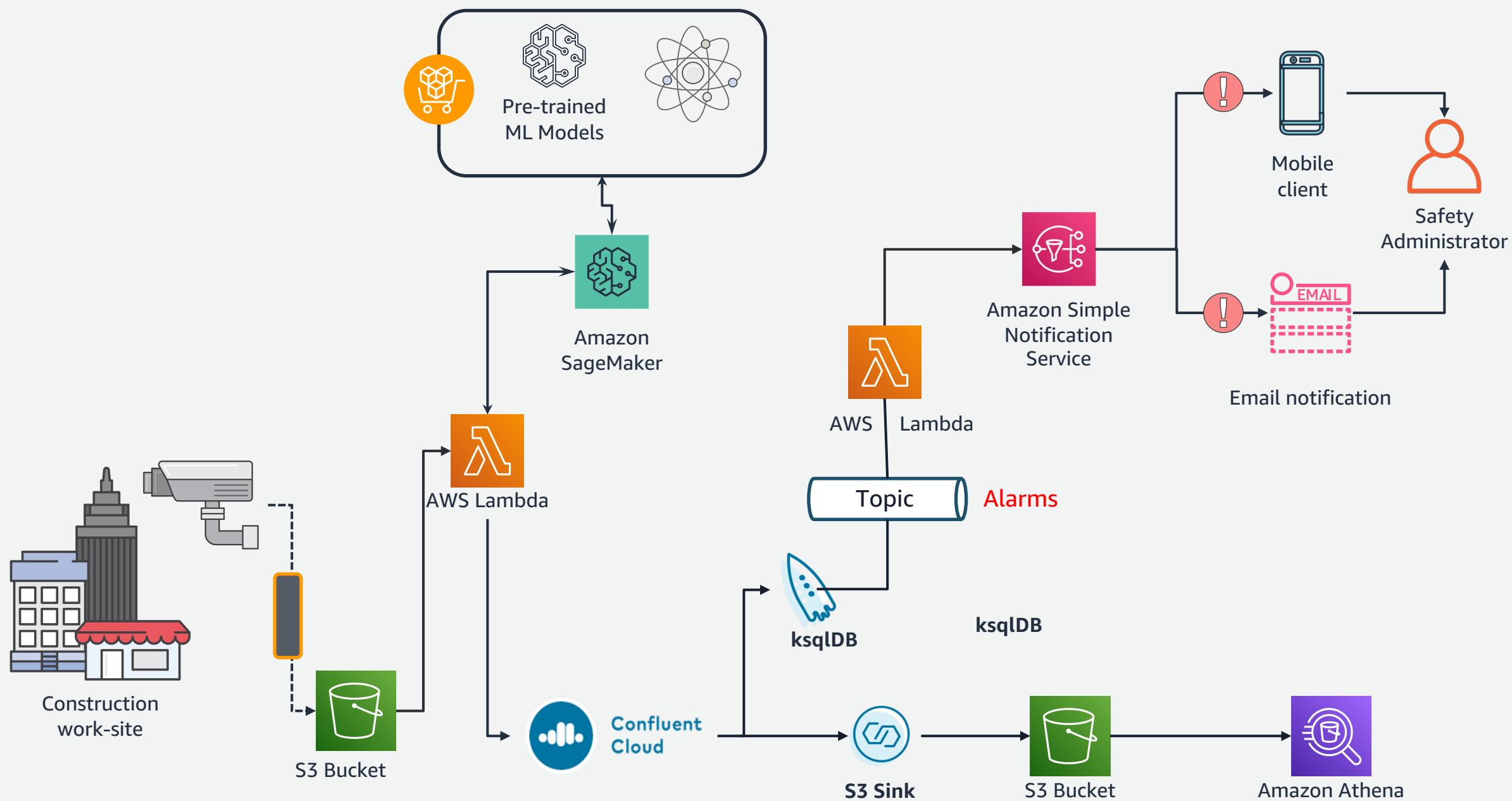
# Building an ML-driven streaming application

<b>Video/data storage</b>  Amazon Simple Storage Service (S3)	<b>ML Model</b>  Model	<b>ML Model deployment mechanism</b>  Amazon SageMaker	<b>Serverless compute</b>  AWS Lambda	<b>Notification mechanism</b>  Amazon Simple Notification Service
<b>Event Streaming platform</b>  Confluent Cloud	<b>KSQLDB</b>  ksqlDB	<b>Data Visualization tool</b>  Amazon QuickSight	<b>Interactive Query service</b>  Amazon Athena	<b>Metadata storage</b>  Amazon DynamoDB



# Demo





# ML Models



[AKTE - Forklift Detector](#)  
Pre-trained Model



[GluonCV SSD Object Detector](#)  
Pre-trained Model



[Construction Worker Detection](#)  
Pre-trained Model



[Social Distancing Detector](#)  
Pre-trained Model



[Helmet & Vest Detector for Worker Safety](#)  
Pre-trained Model



[PPE Detector for Laboratory Safety](#)  
Pre-trained Model



[Construction Machines Detector](#)  
Pre-trained Model



[TensorloT CV PPE Mask Detection](#)  
Pre-trained Model



[Hard Hat Detector for Worker Safety](#)  
Pre-trained Model

# AWS Marketplace for Machine Learning



Curated and trusted catalog of over **400** ML model packages and algorithms

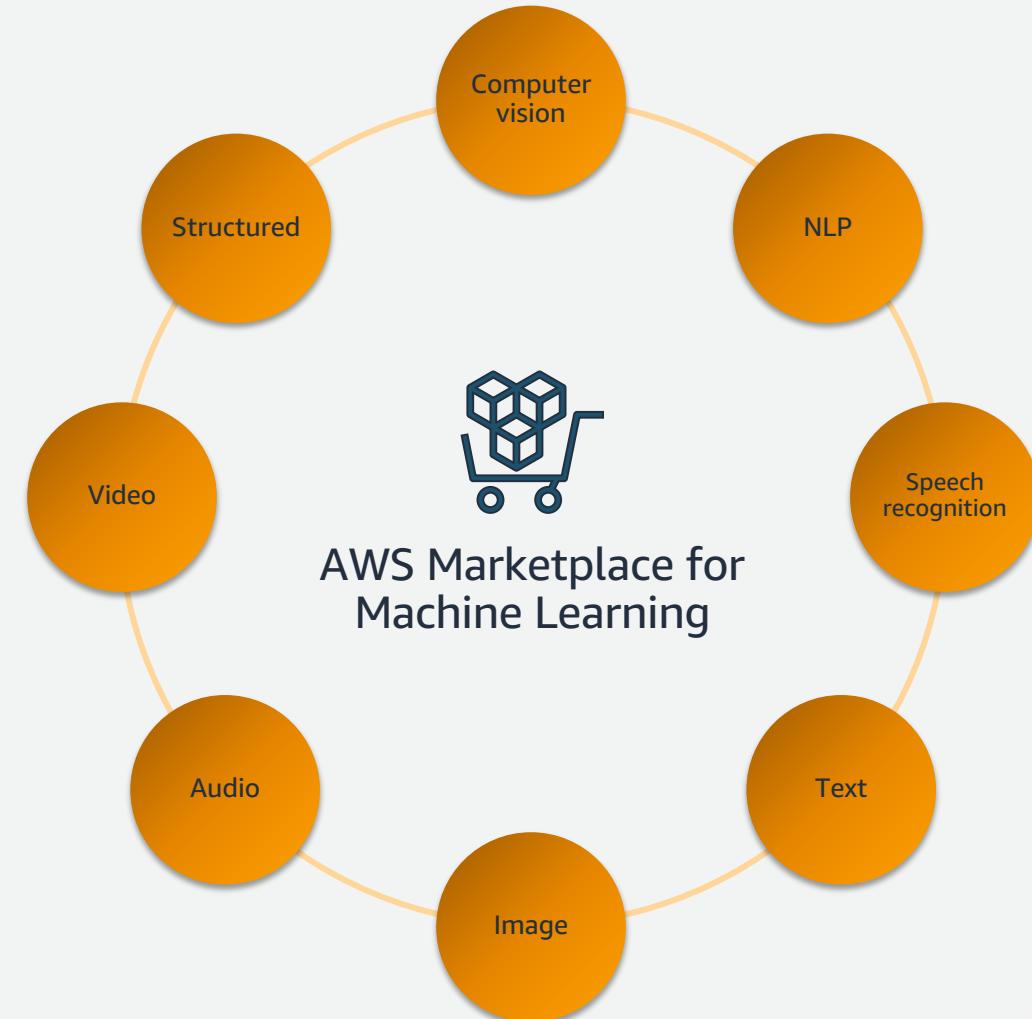
→ 14 industry segments

→ 61 sellers

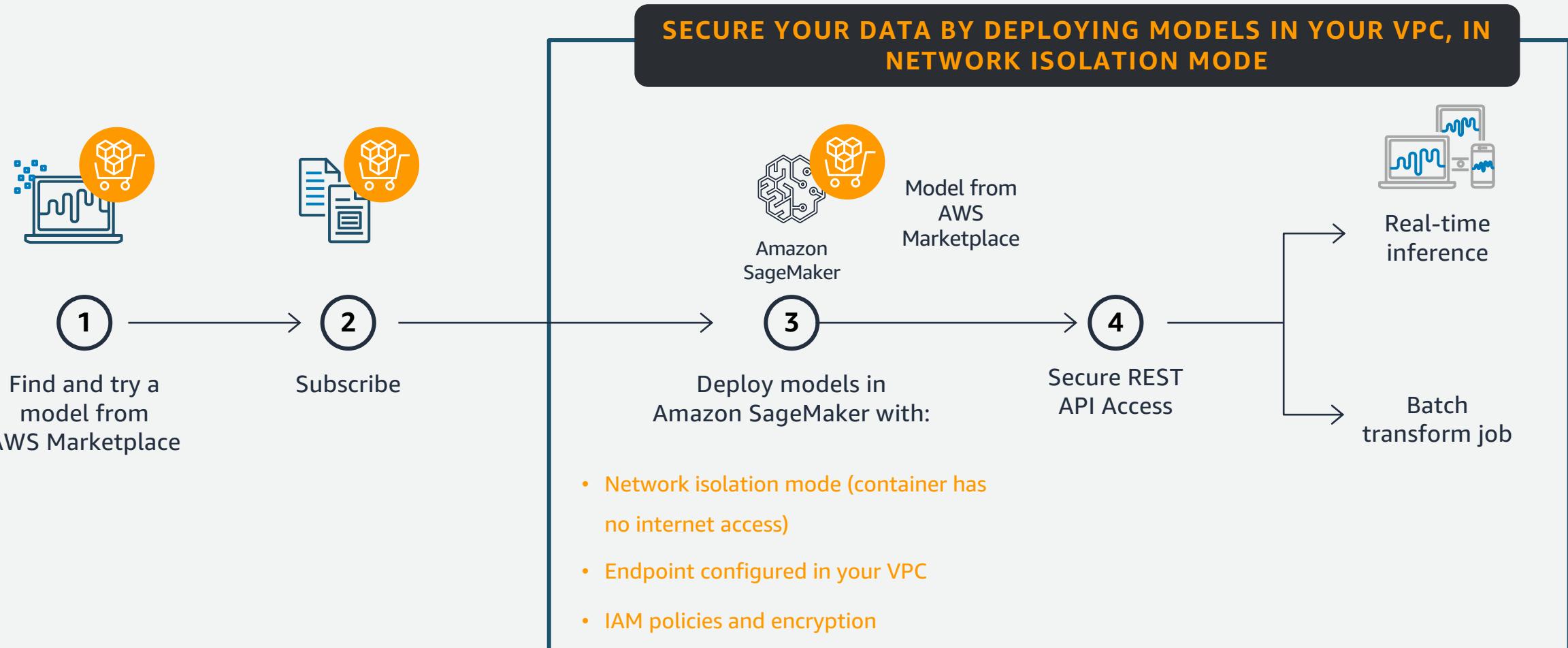
Simplified provisioning

→ Consolidated into AWS billing

→ Free | Free trial | Paid subscriptions



# Deploying ML models from AWS Marketplace



<https://aws.amazon.com/marketplace/solutions/machine-learning/pre-trained-models>



# **How to deploy ML models with Amazon Sagemaker**



# AWS Marketplace

**8,000+** • **1,600+** ISVs  
listings

**24** regions

• **290,000+** customers

• **1.5M+** subscriptions



**Flexible consumption  
and contract models**

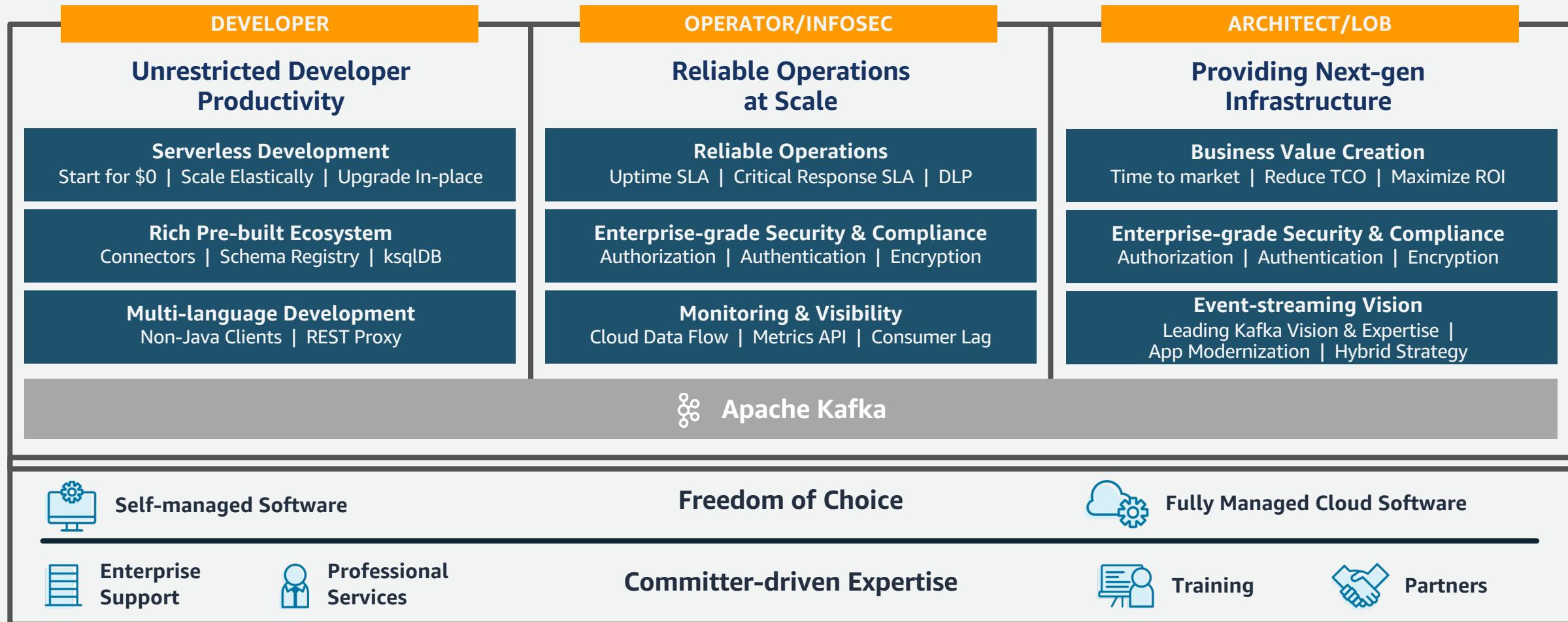


**Quick and  
easy deployment**



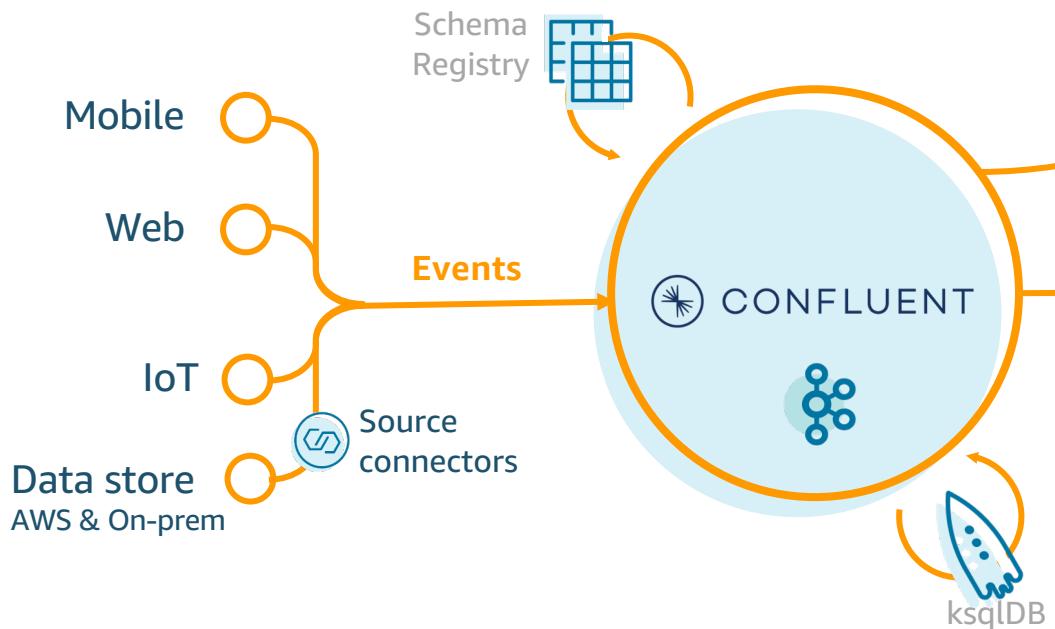
**Helpful humans  
to support you**

# Confluent Cloud



# Real-time analysis

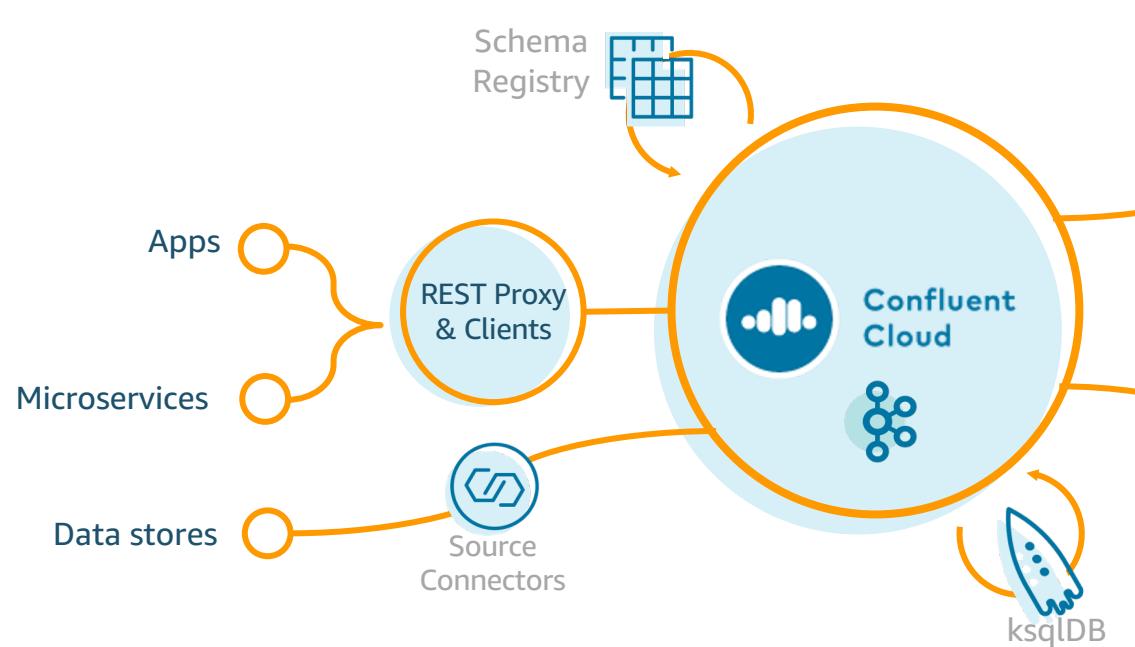
## Ingest & Process



**Store & Analyze**  
Stream data into your AWS data lake or data warehouse to execute queries on streaming data for real-time and batch analytics

# Serverless integrations

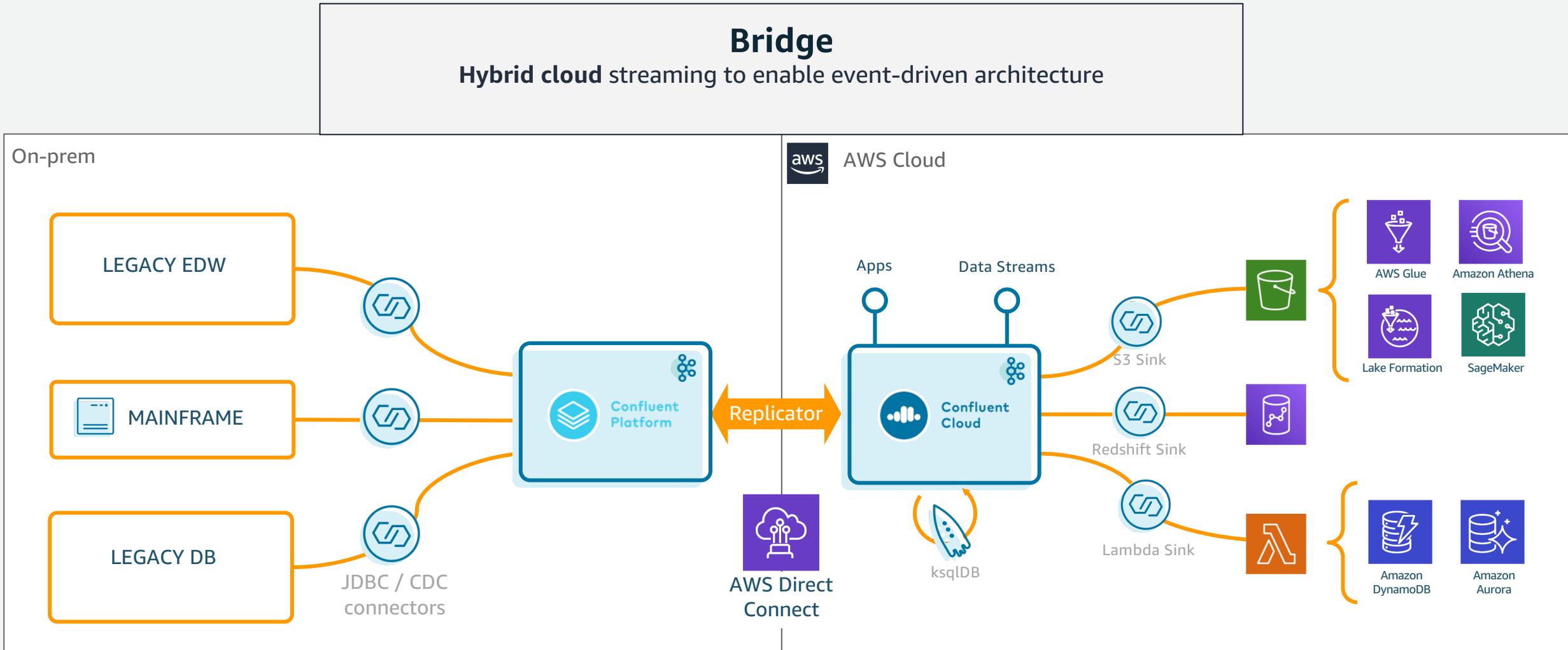
## Serverless integration



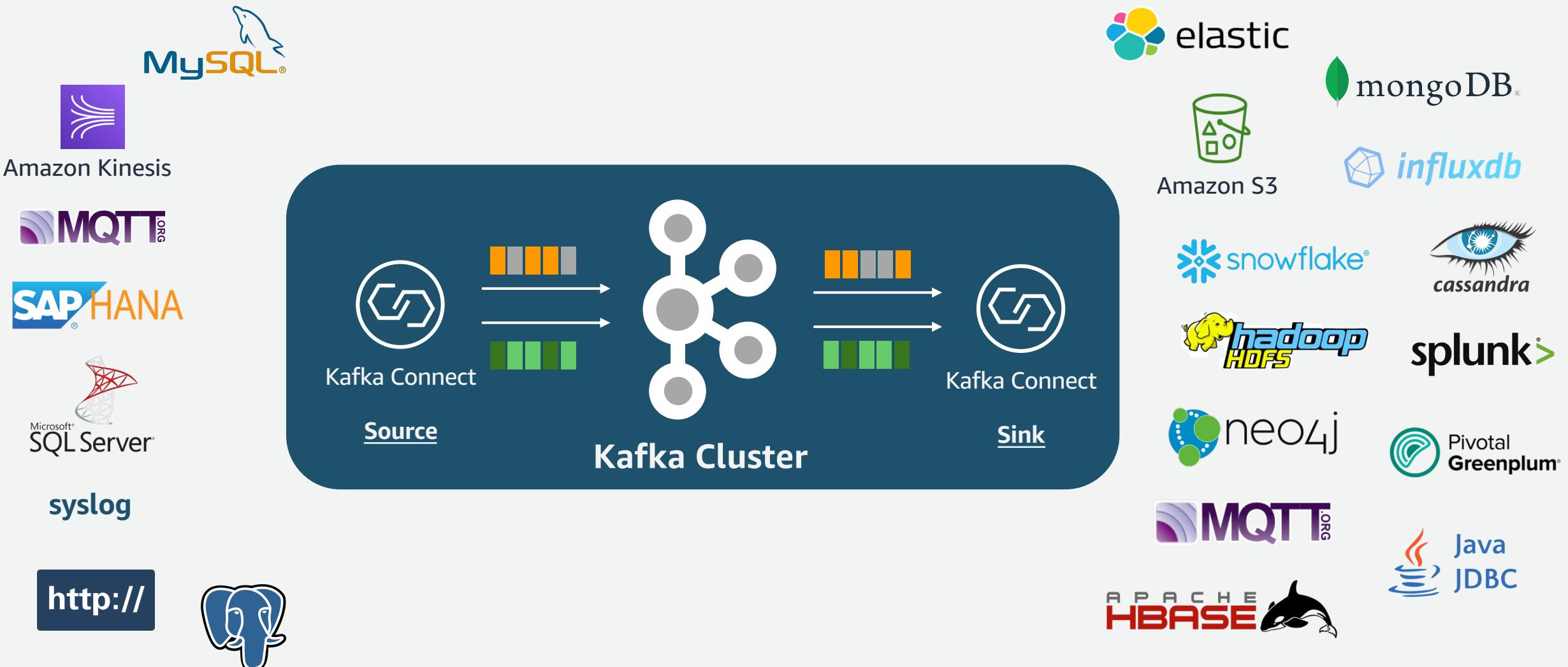
## AWS serverless platform



# Modernization



# Integrations via connectors available in aws marketplace



# Disney+hotstar powers explosive growth

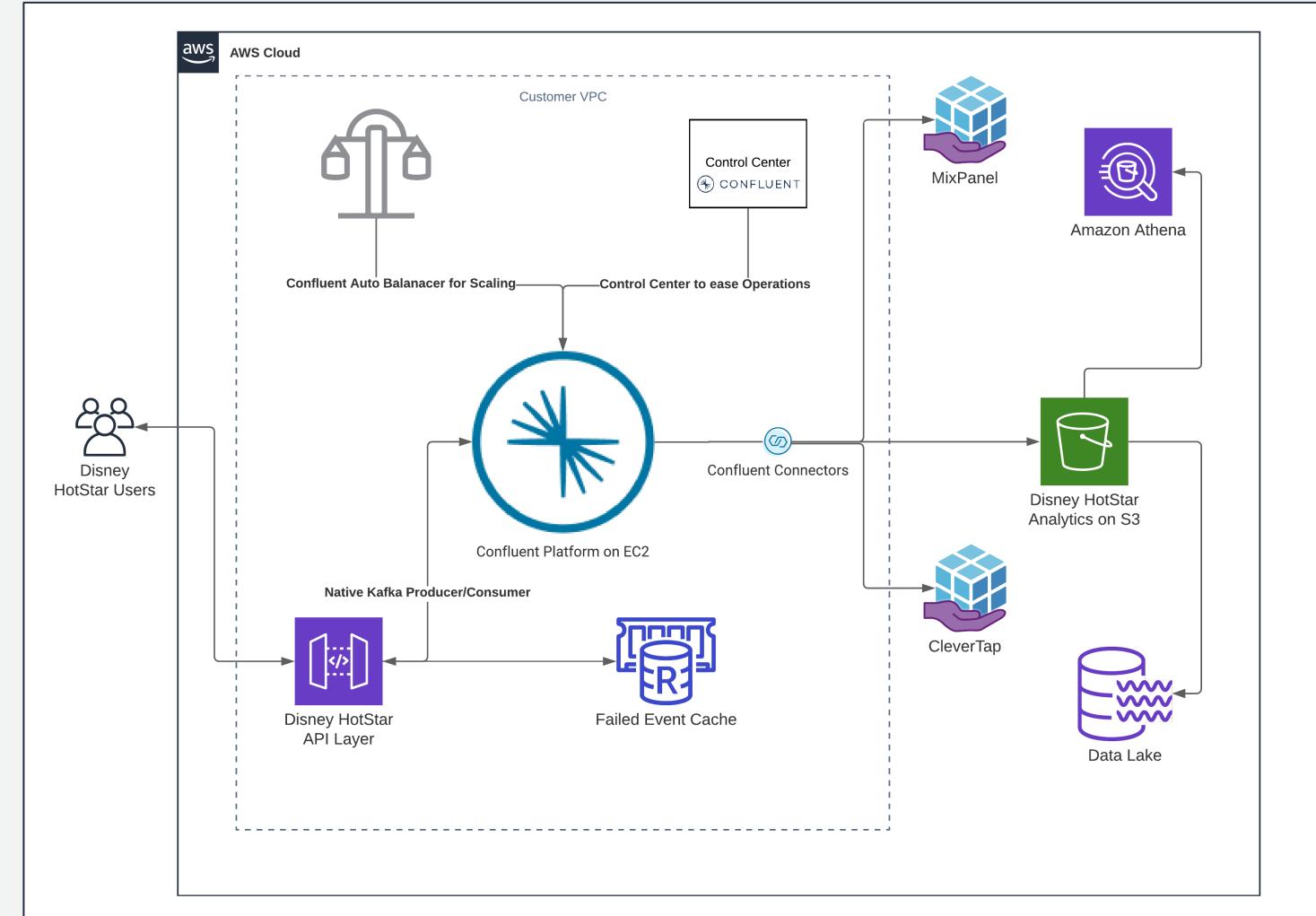
with real-time event streaming on Confluent Platform



**5 times** higher real-time event streaming (10 to 15 billion messages)

Increased number of users on the platform by **20 times**

Increased number of simultaneous users to over **25 million**



# Confluent on the AWS Marketplace (free trial)

The screenshot shows the AWS Marketplace homepage. At the top, there's a search bar with a magnifying glass icon. Below it, a banner highlights Confluent as a 'Leader in SaaS marketplaces' in the Forrester New Wave report. A large orange button labeled 'View all products' is prominent. The main content area features a search interface with dropdown menus for 'Categories', 'Vendors', 'Pricing Plans', and 'Delivery Methods'. Below this is a section titled 'Popular Categories' with icons for Operating Systems, Security, Networking, Storage, Analytics, Tools, and Machine Learning.

<https://aws.amazon.com/marketplace>

The screenshot shows the product page for 'Apache Kafka® on Confluent Cloud™ - Pay As You Go'. At the top right, a green arrow points to the 'Continue to Subscribe' button. The page includes a Confluent Cloud logo, a brief description, and tabs for Overview, Pricing, Usage, Support, and Reviews. The 'Product Overview' section contains text about starting a 3-month trial with \$200 off, the benefits of Confluent Cloud, and how to get started. A 'Highlights' sidebar lists several key features. At the bottom, there's information about hybrid deployment options.

# AWS Marketplace can help you get started

## Find



A breadth  
of development solutions:



## Buy



Through flexible  
pricing options:

Free trial

Pay-as-you-go

Hourly | Monthly | Annual  
| Multi-Year

Bring Your Own License (BYOL)

Seller Private Offers

Channel Partner Private Offers

## Deploy



With multiple  
deployment options:

AWS Control Tower

AWS Service Catalog

AWS CloudFormation  
(Infrastructure as Code)

Software as a Service (SaaS)

Amazon Machine Image (AMI)

Amazon Elastic Container Service  
(ECS)

Amazon Elastic Kubernetes Service  
(EKS)

# Webinar summary

-  Think about Kafka for your event-based architectures and consider pre-build ML models with Amazon Sagemaker to get started quickly
-  Look to AWS partners like Confluent to help you scale and provide support
-  Use AWS Marketplace to help you discovery, experiment and innovate with new tools