

An Introduction to Apache Kafka[®] and IBM Event Streams

Andrew Dunnings

andrew.dunnings1@ibm.com



87%

of companies are transforming to be more customer-centric



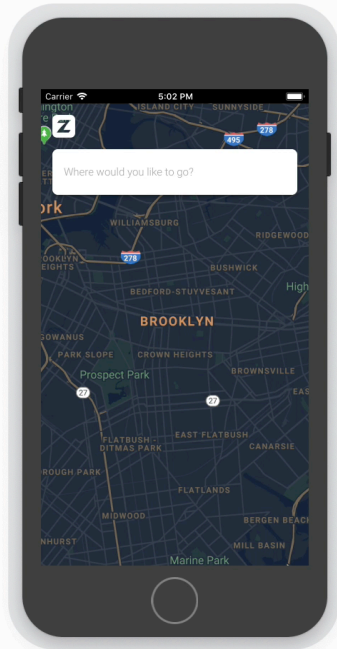
Typical Event-driven Use Case

Customer Satisfaction

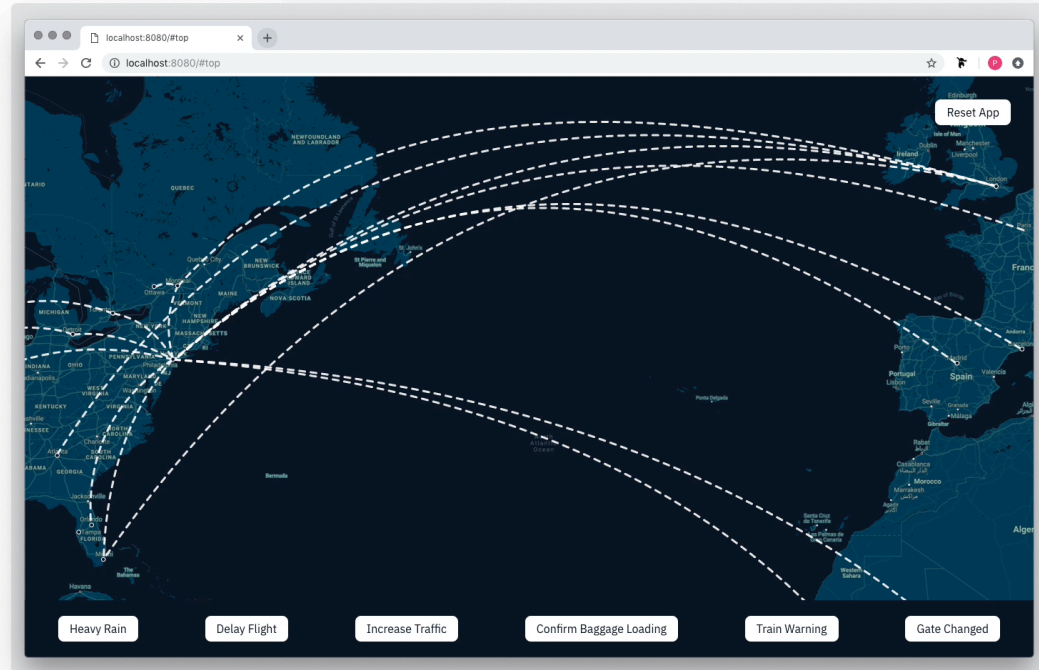


‘Zoom Air’ is a commercial airline

Re-accommodate passengers
before they realize their journey
has been disrupted



Z ZoomAir Phone Application
Connect with your passengers...



ZoomAir Data Controller
... turn disruptive events into helpful actions.

Event-Driven delivered engaging customer experiences

*Getting data to where it's needed, **before** it's needed*



React to events in real time, as they happen



Deliver **responsive** customer experiences

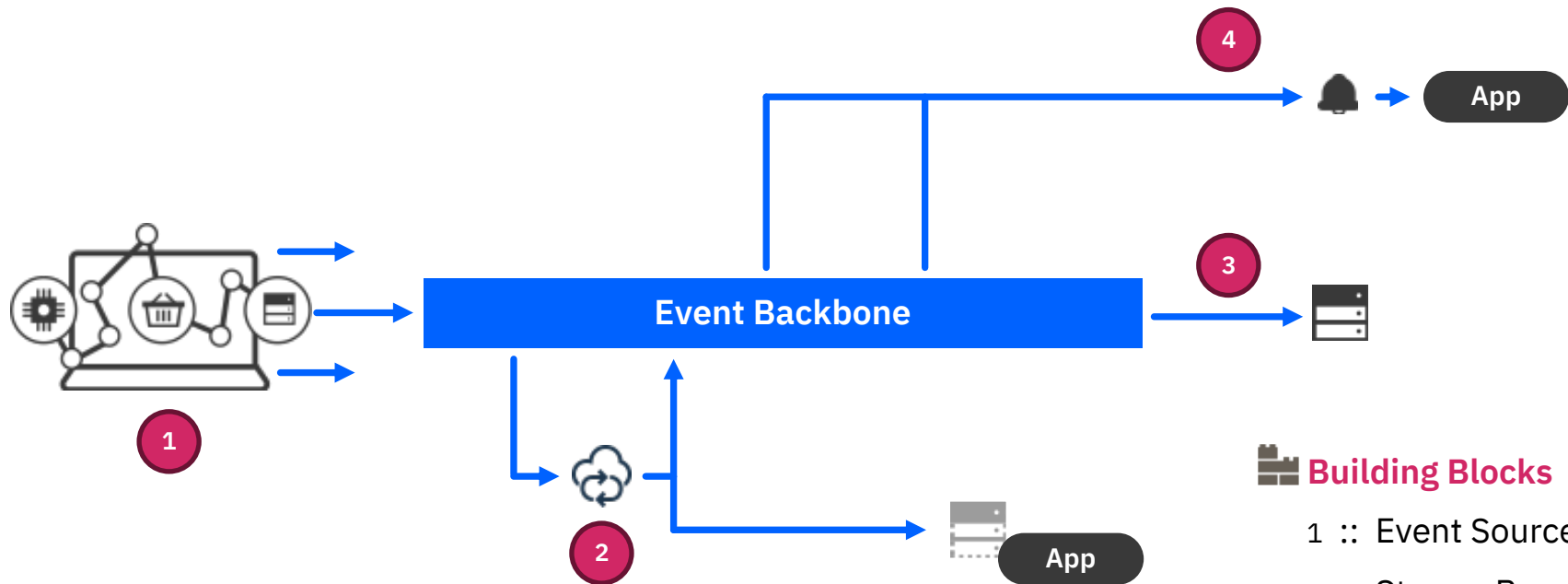


Bring real-time **intelligence** to your apps

*Creating 'Event-Driven' applications requires new **tools** and a different **approach***

Event-driven in practice

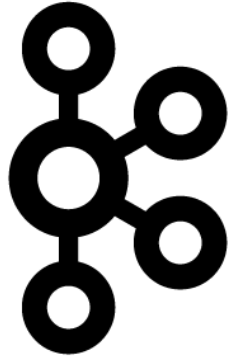
Components of an event-streaming application



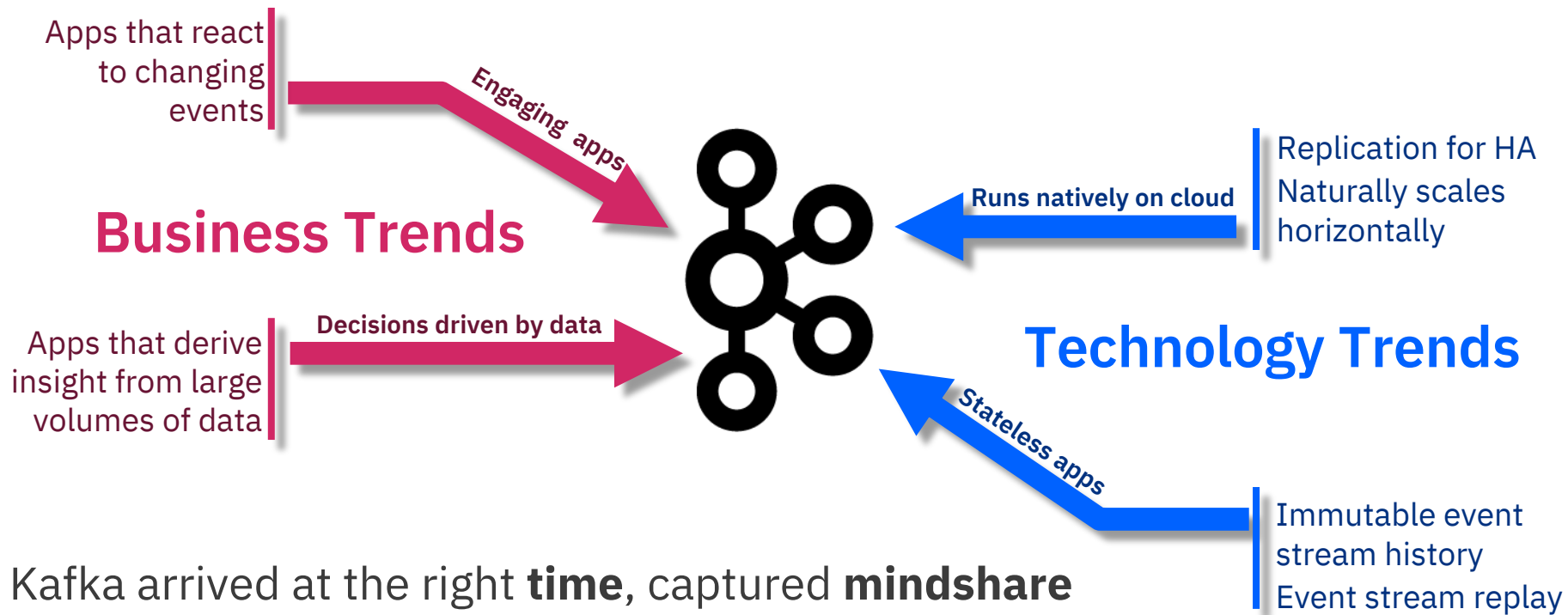
Building Blocks

- 1 :: Event Sources
- 2 :: Stream Processing
- 3 :: Event Archive
- 4 :: Notifications

Why is Apache Kafka so popular?



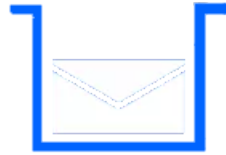
Why is Apache Kafka so popular?



Kafka arrived at the right **time**, captured **mindshare** among **developers** and exploded in popularity

Two Styles of Messaging

MESSAGE QUEUING



Transient data
persistence



Request/reply

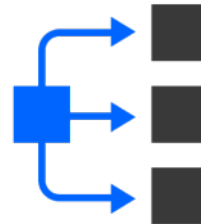


Targeted
reliable delivery

EVENT STREAMING



Stream history

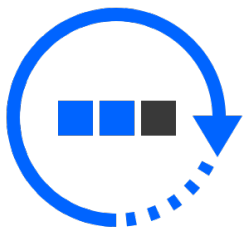


Scalable
consumption

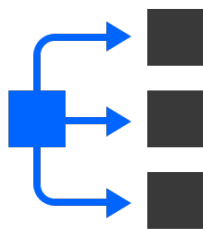


Immutable data

Properties of the Event Backbone



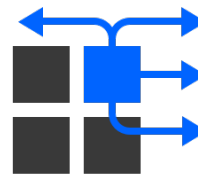
Stream history



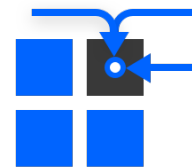
Scalable consumption



Immutable data



Scalable



Highly available

Apache Kafka

Apache Kafka is an **open source, distributed streaming platform**



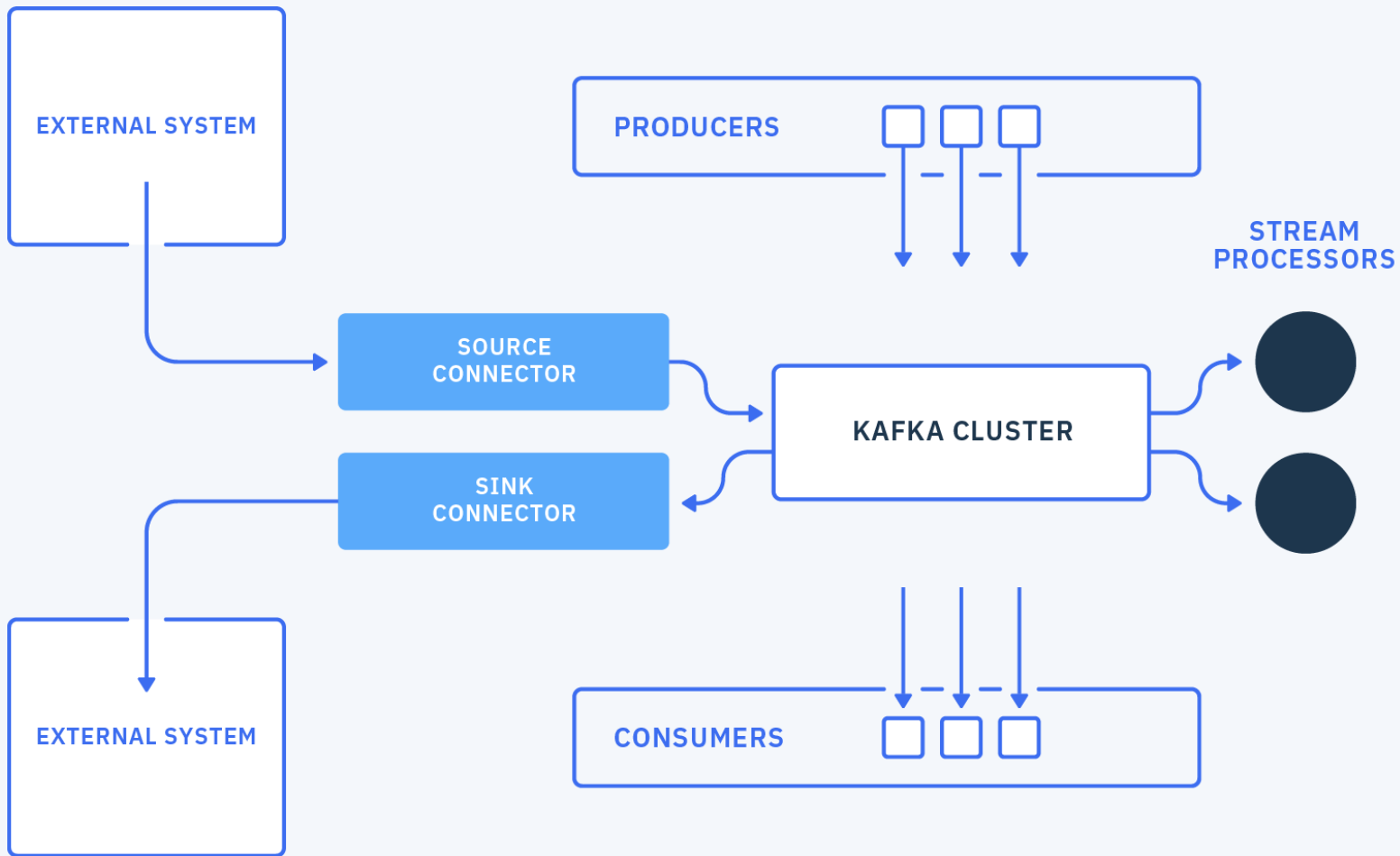
Publish and subscribe to streams of events

Store events in durable way

Process streams of events as they occur

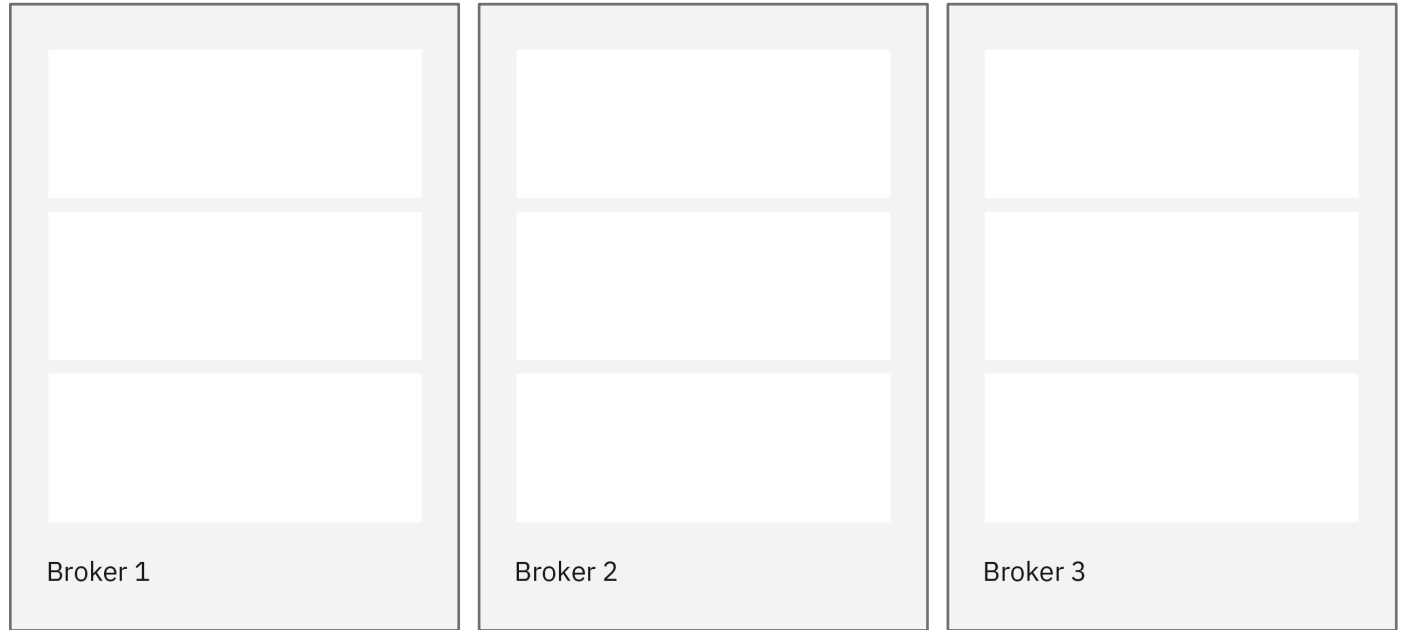
A rapidly growing community...





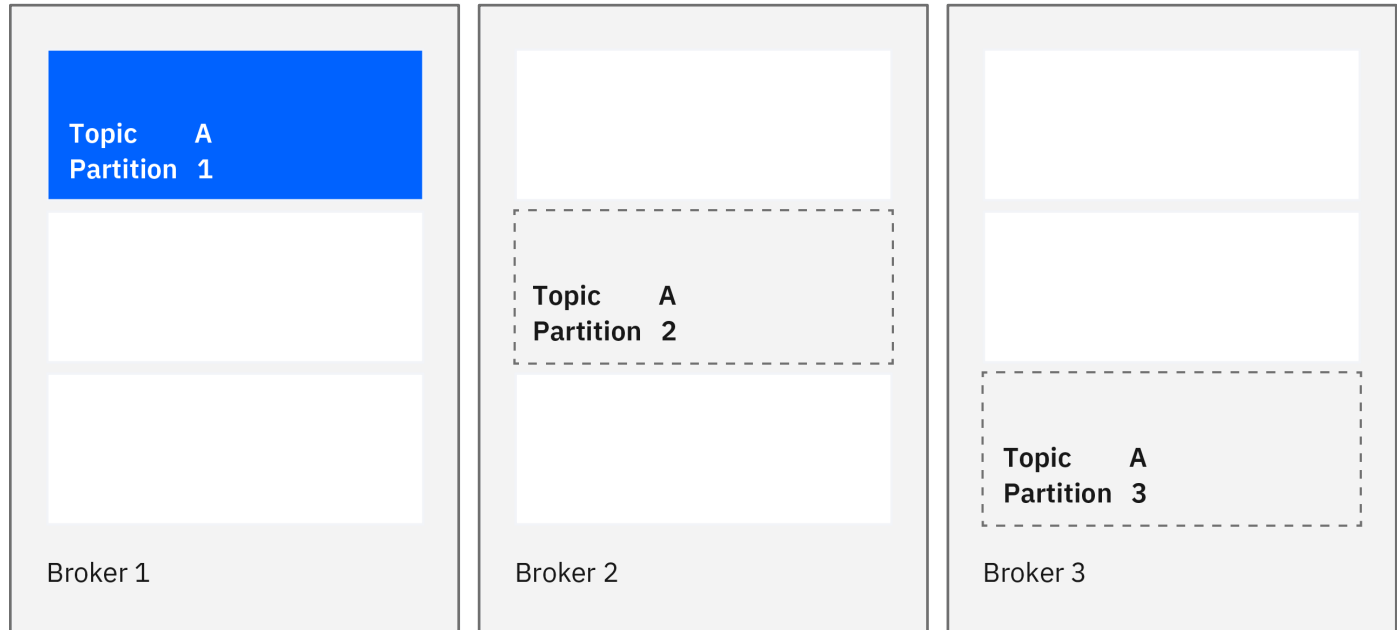


Brokers



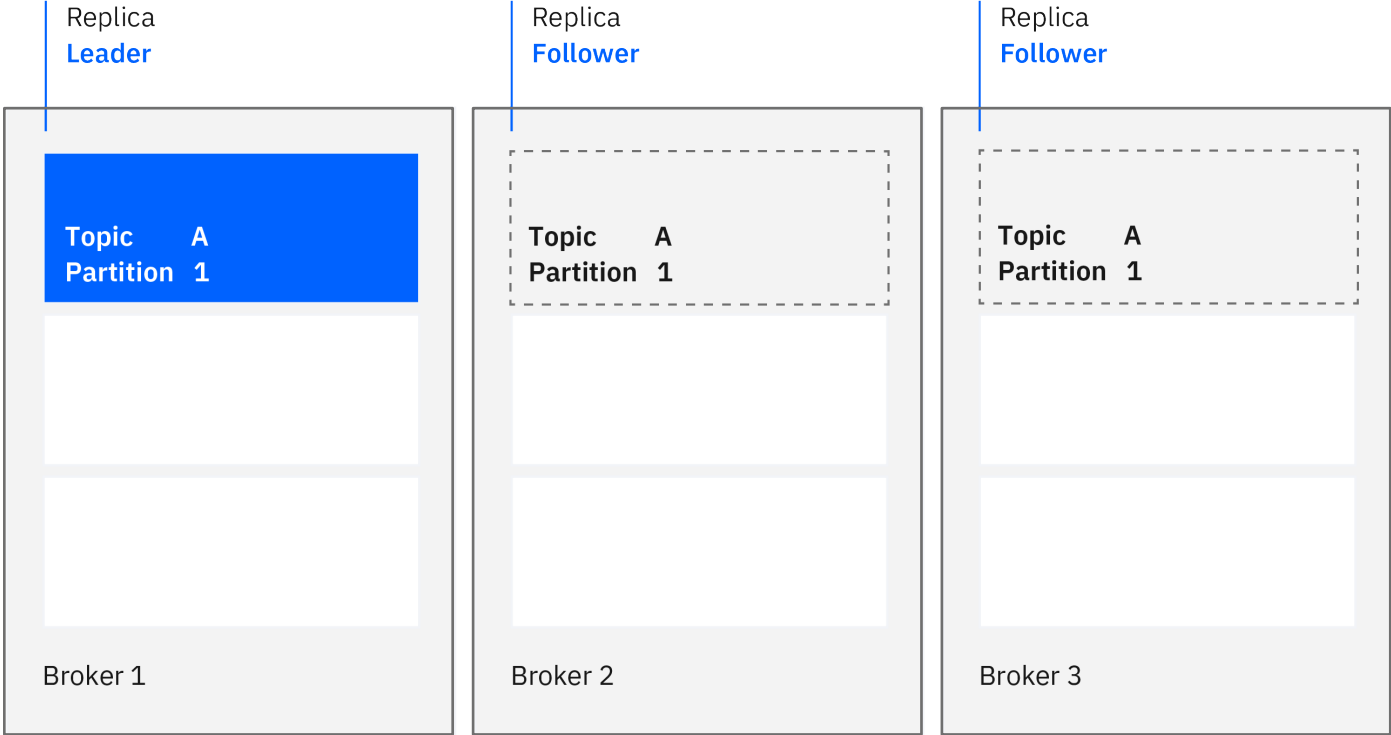
Kafka Cluster

Partitions



Kafka Cluster

Replication



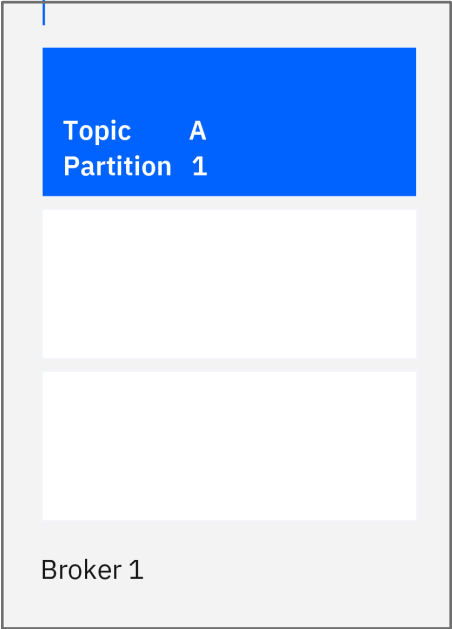
Kafka Cluster

Replication

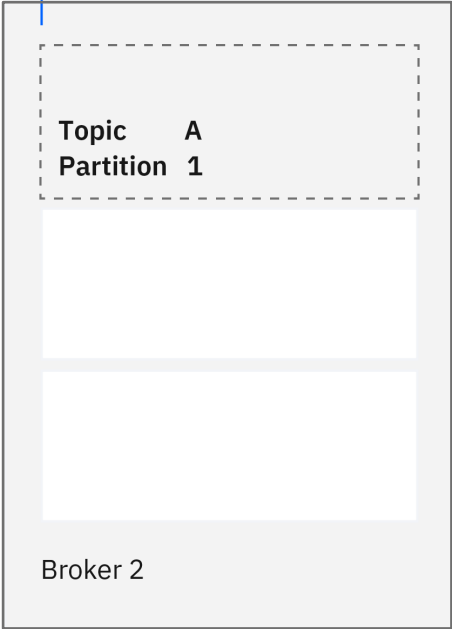
KAFKA CLIENTS



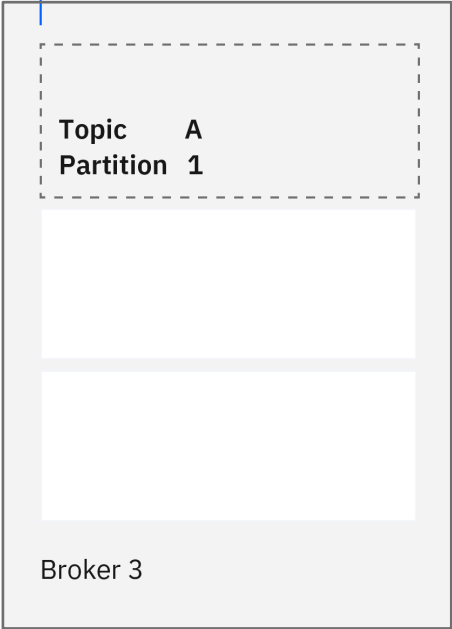
Replica
Leader



Replica
Follower

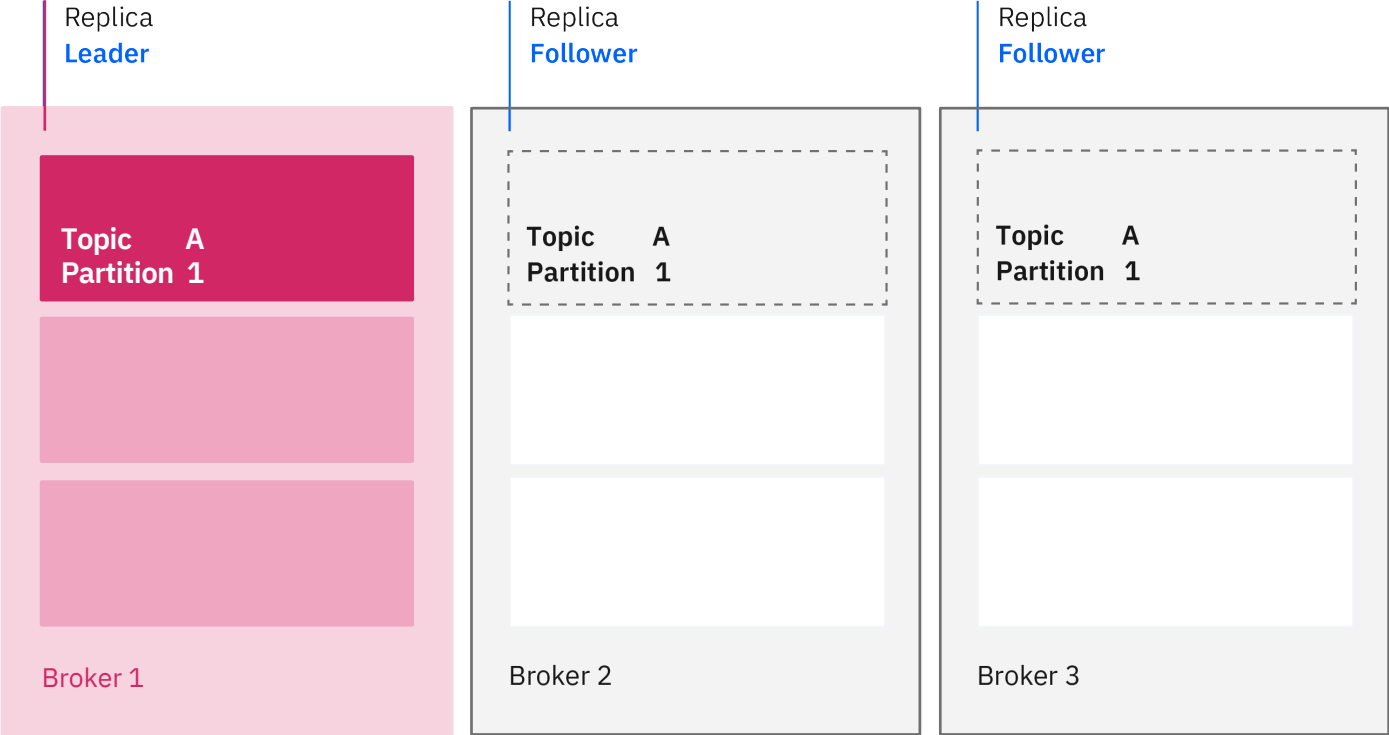


Replica
Follower



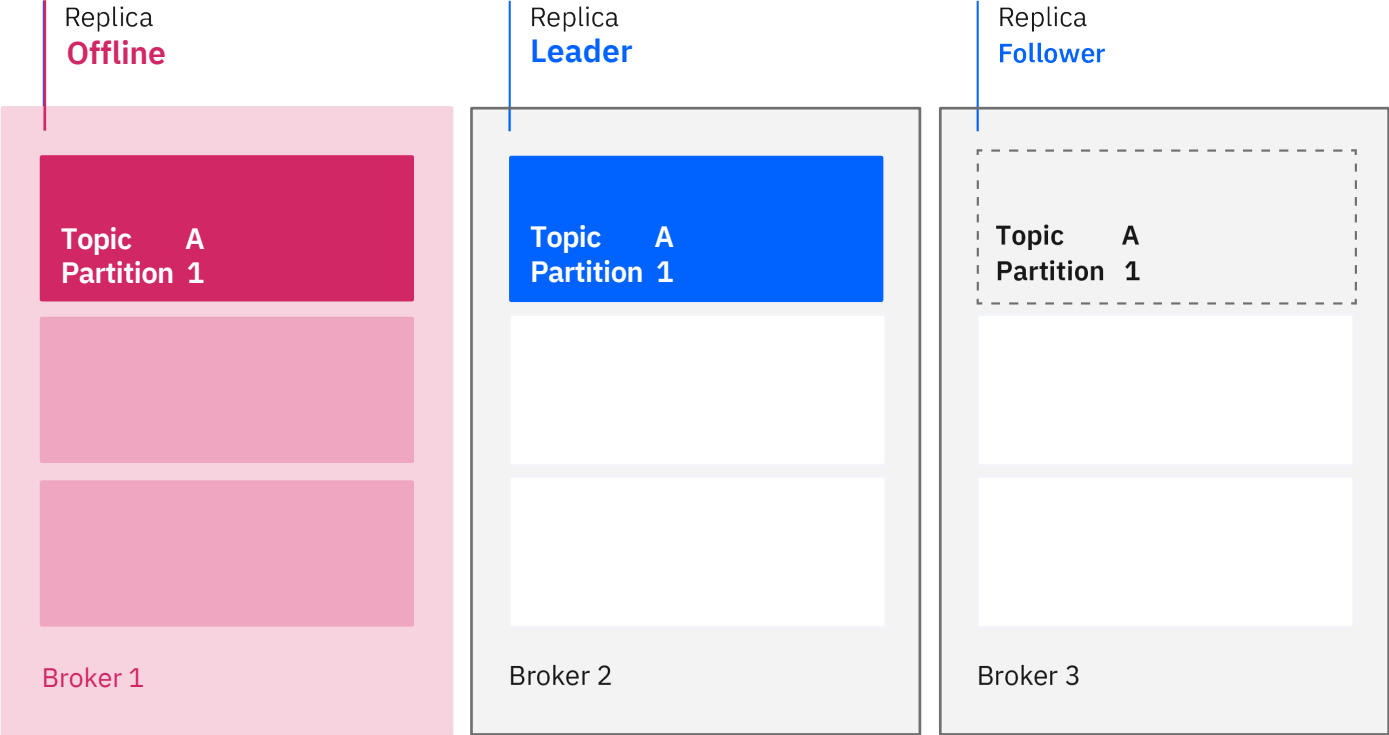
Kafka Cluster

Replication



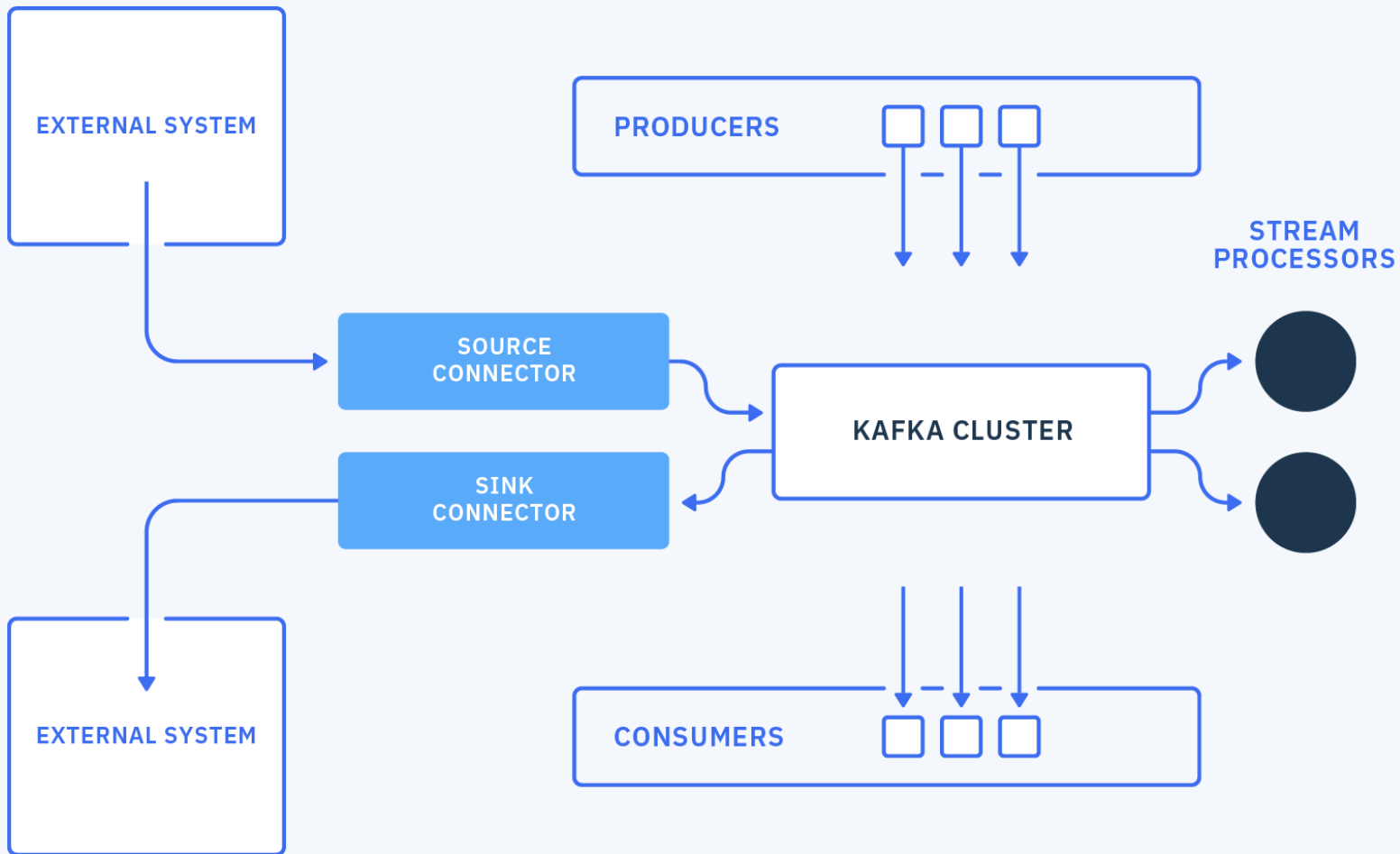
Kafka Cluster

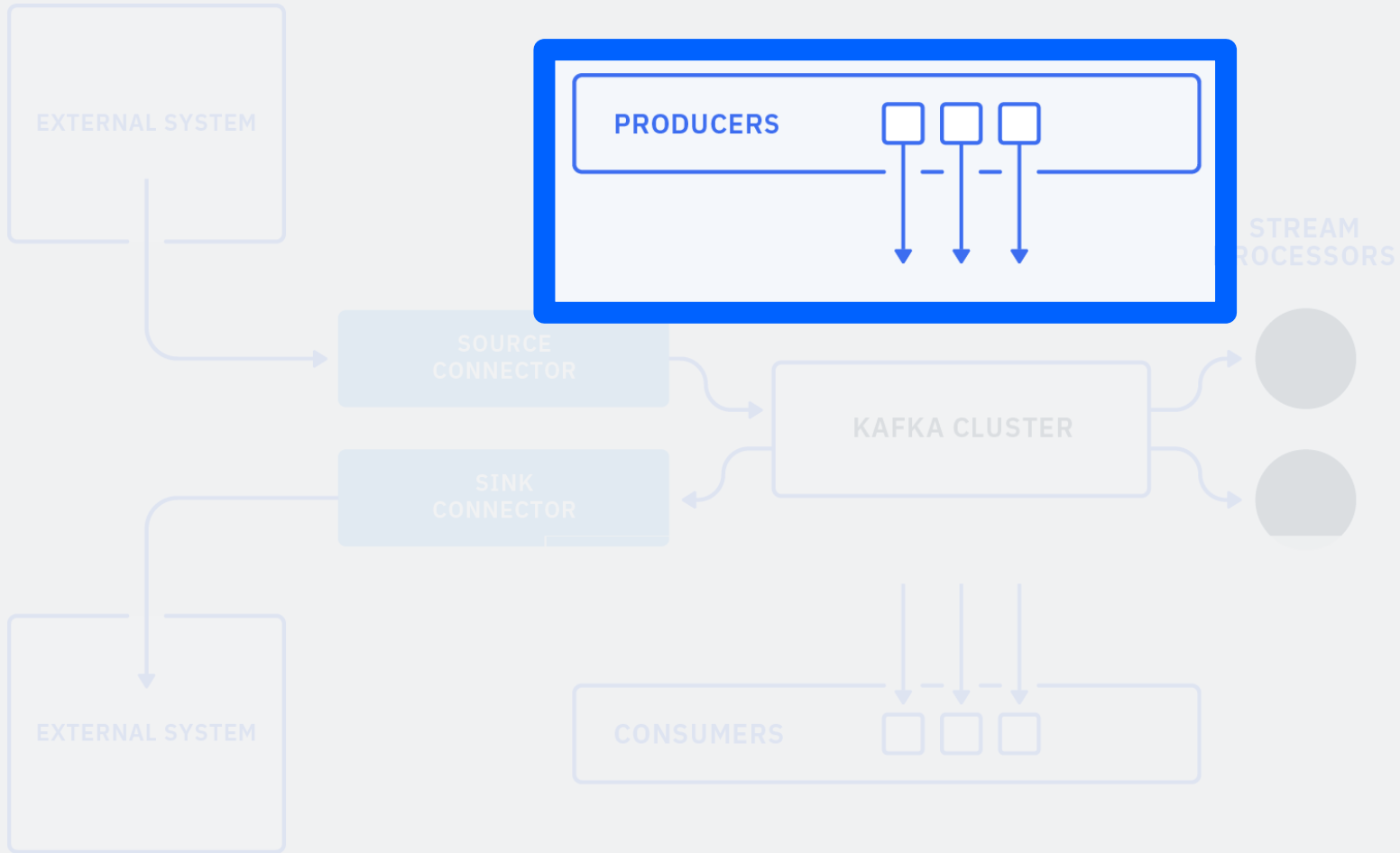
Replication



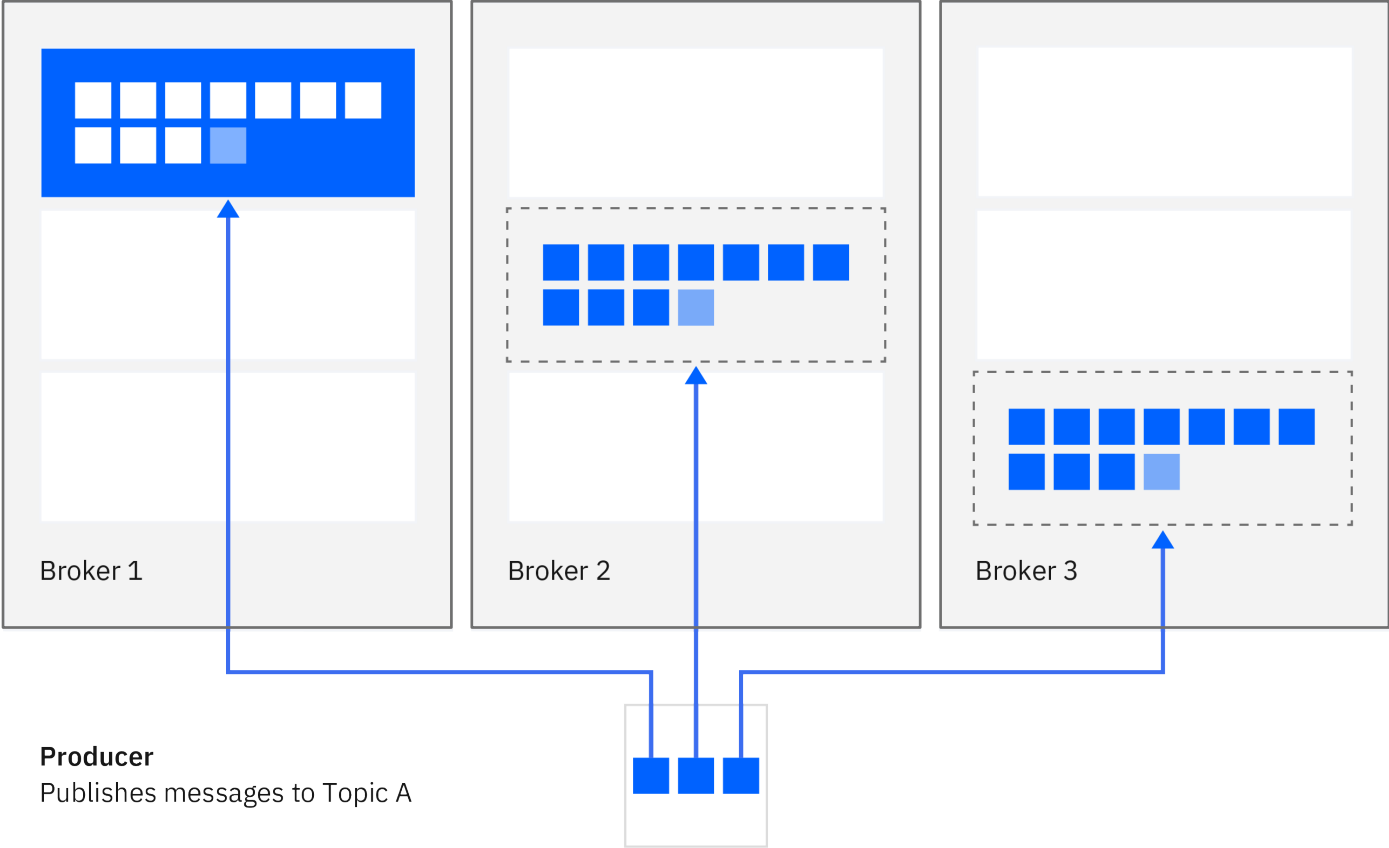
Kafka Cluster

How to I get data into Kafka?

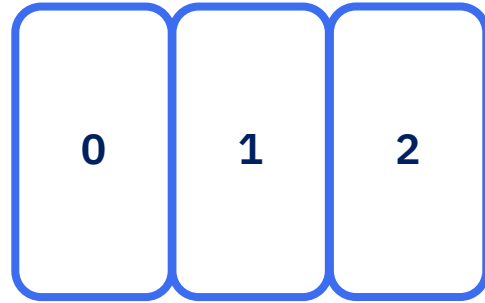




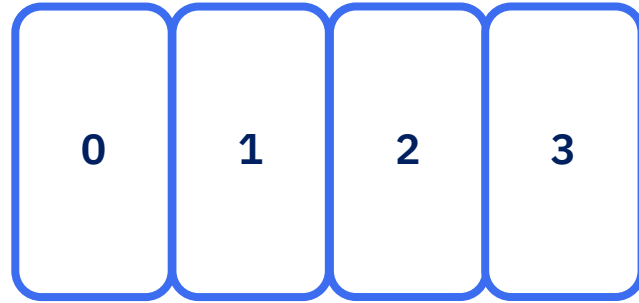
Producers



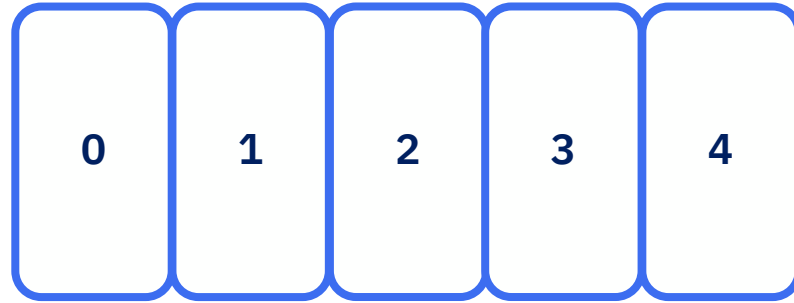
Topics



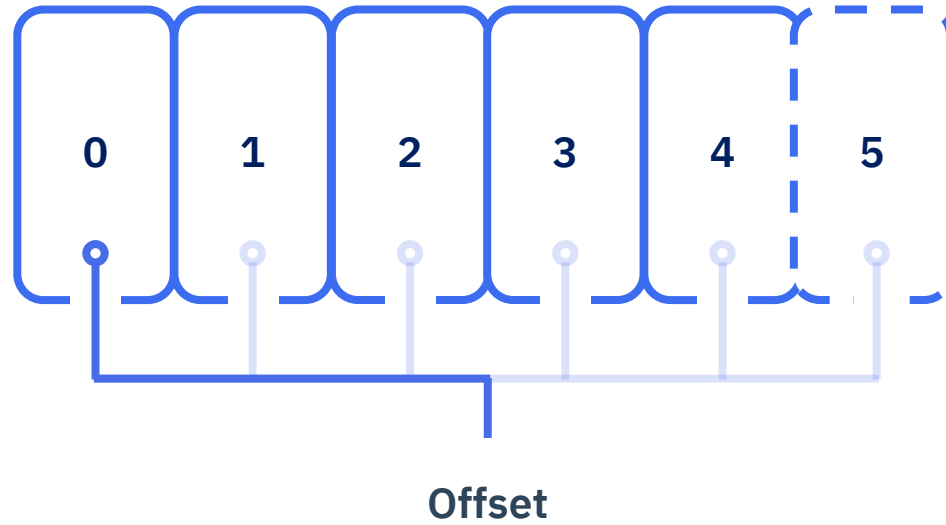
Topics



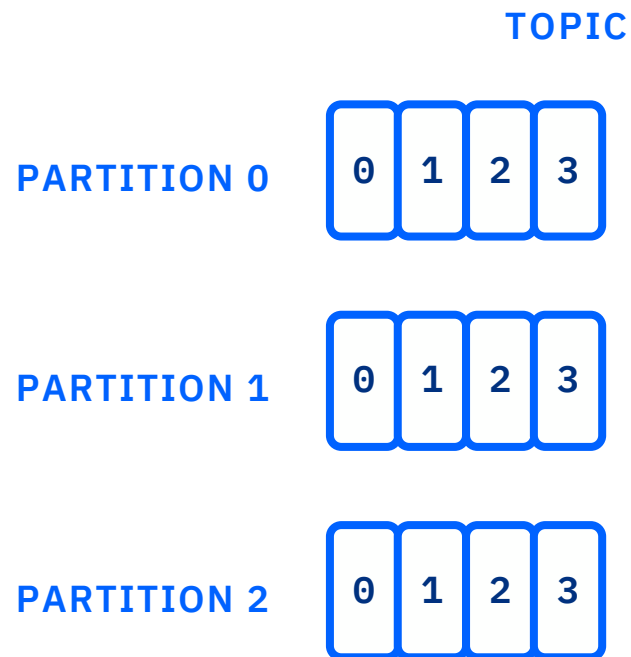
Topics



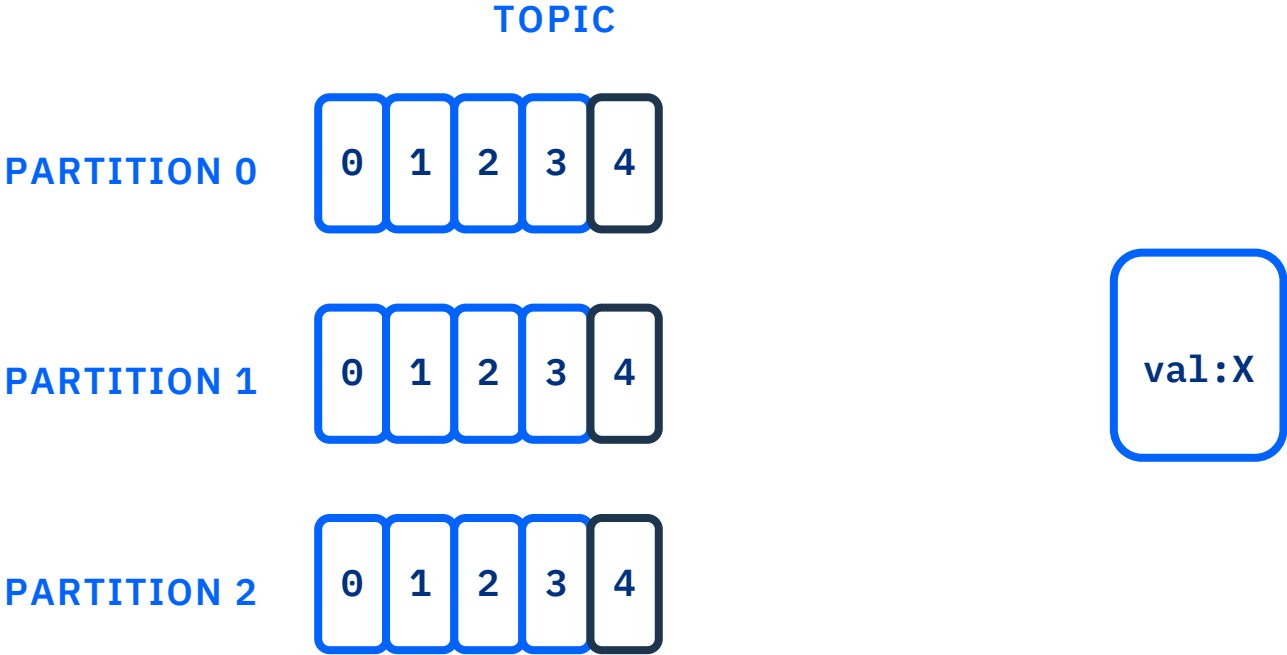
Topics



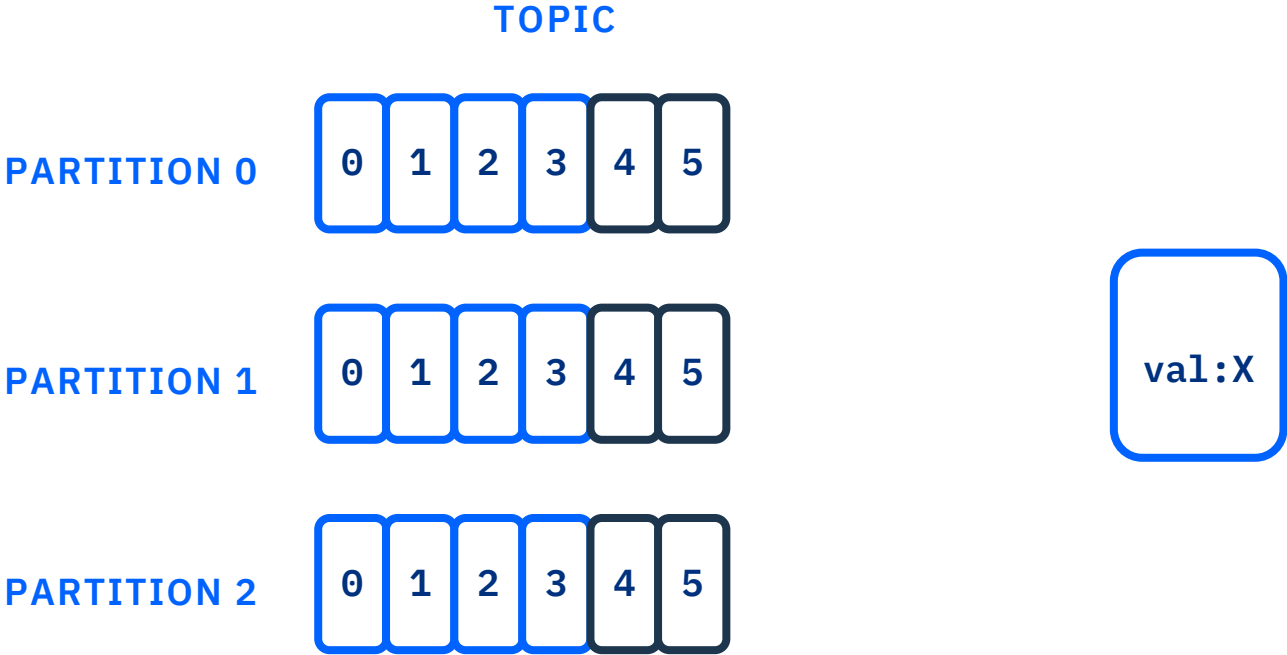
Topics and Keys



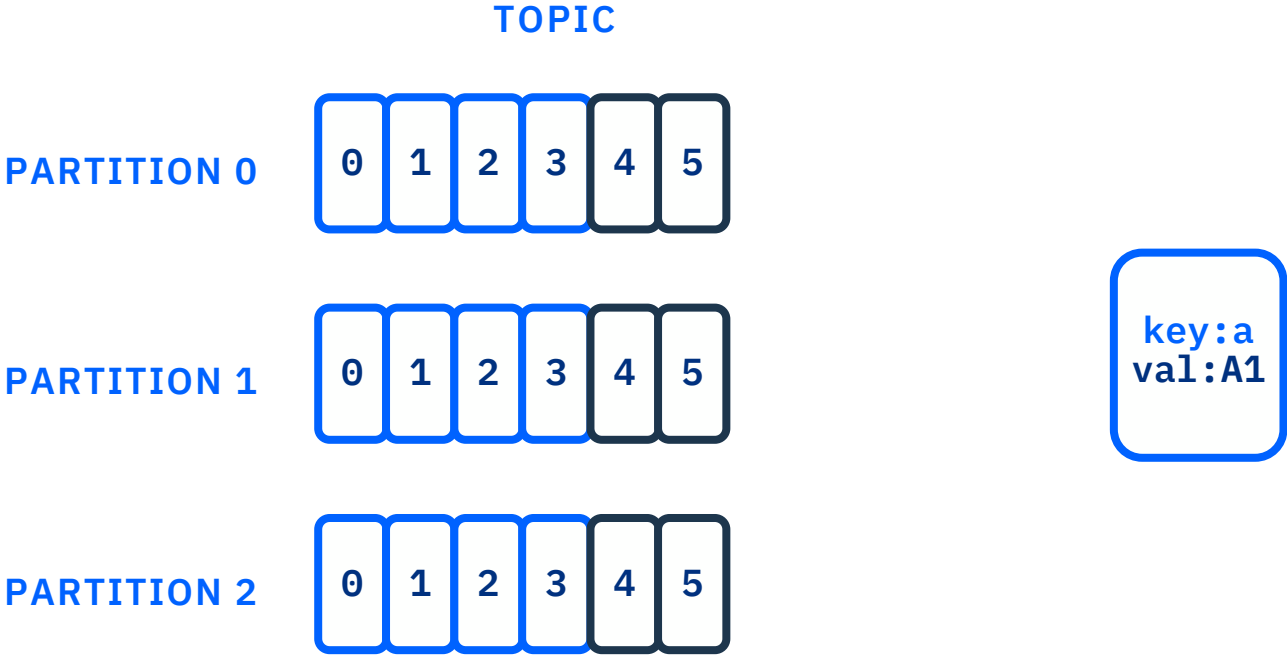
Producers and Keys



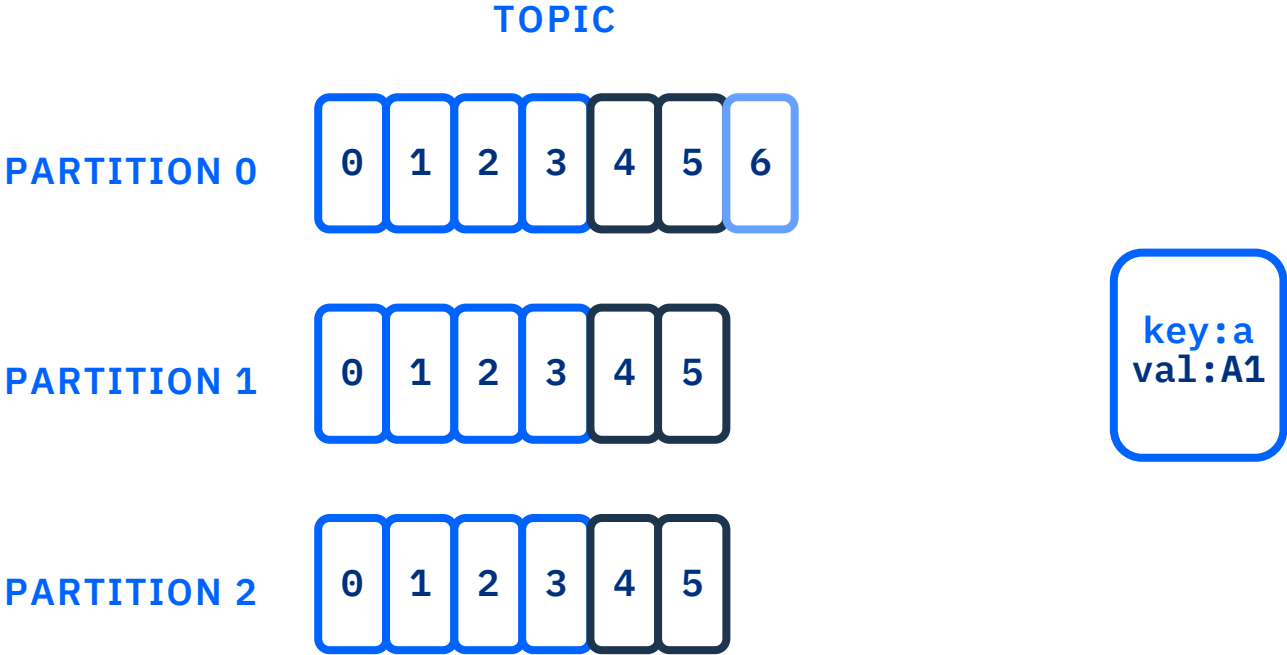
Producers and Keys



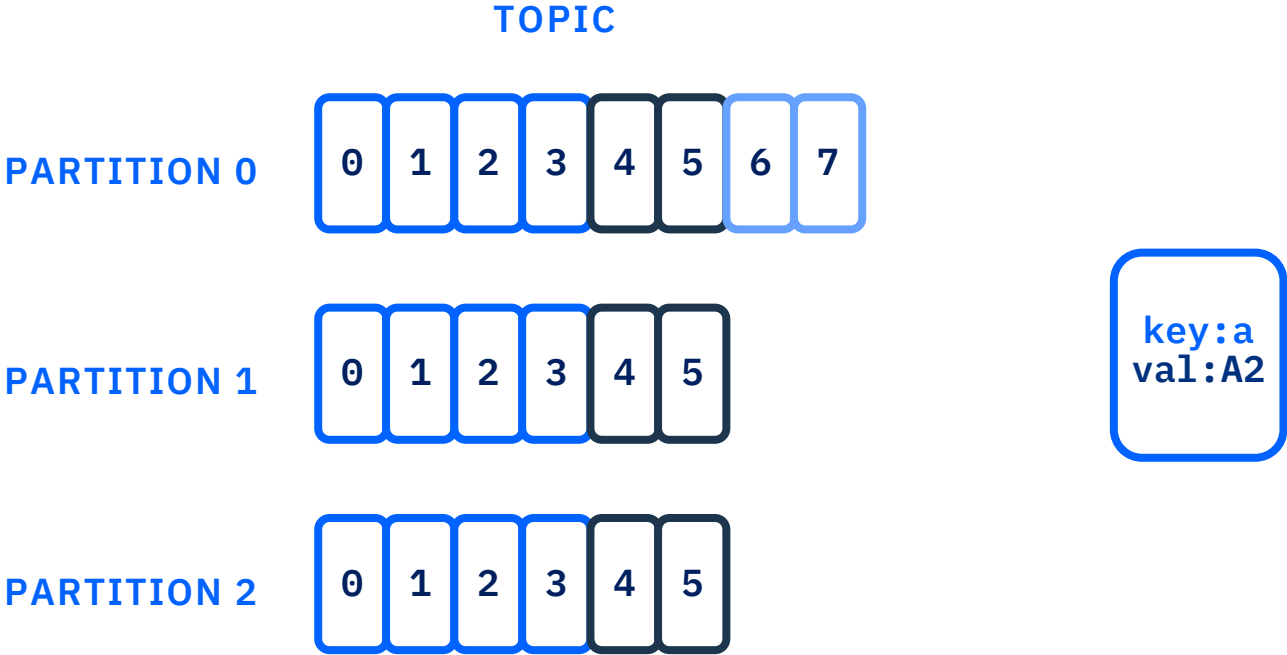
Producers and Keys



Producers and Keys



Producers and Keys



Partitions – Retention

Retention set in time or size

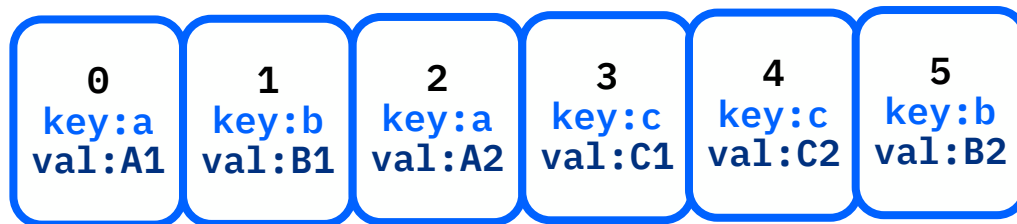
```
log.retention.minutes = 3000  
log.retention.bytes   = 1024
```



Partitions - Compaction

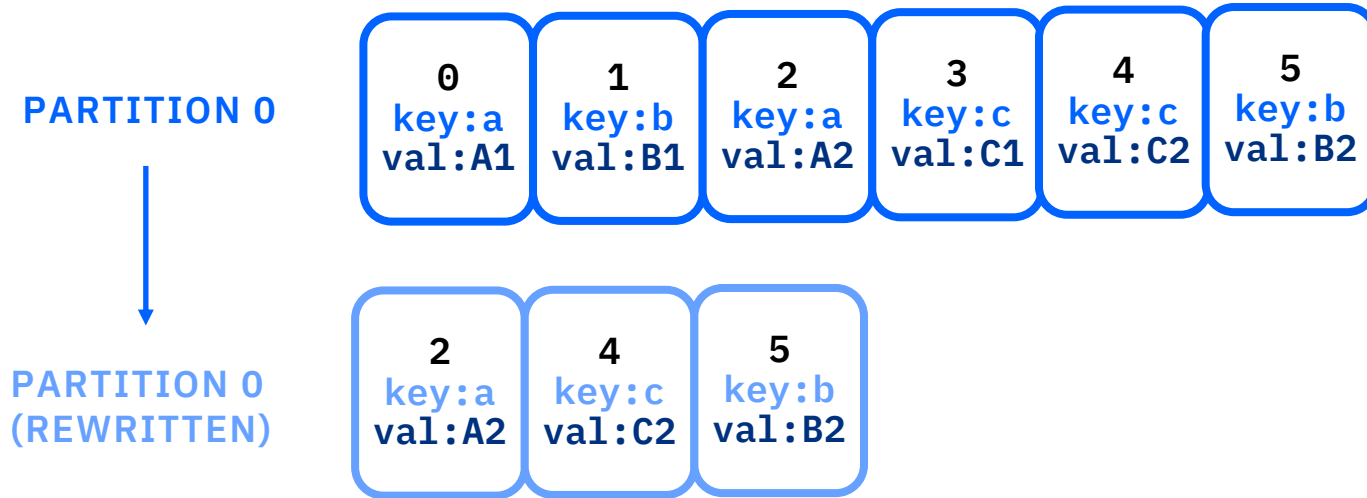
Compacted topics are evolving data stores

PARTITION 0



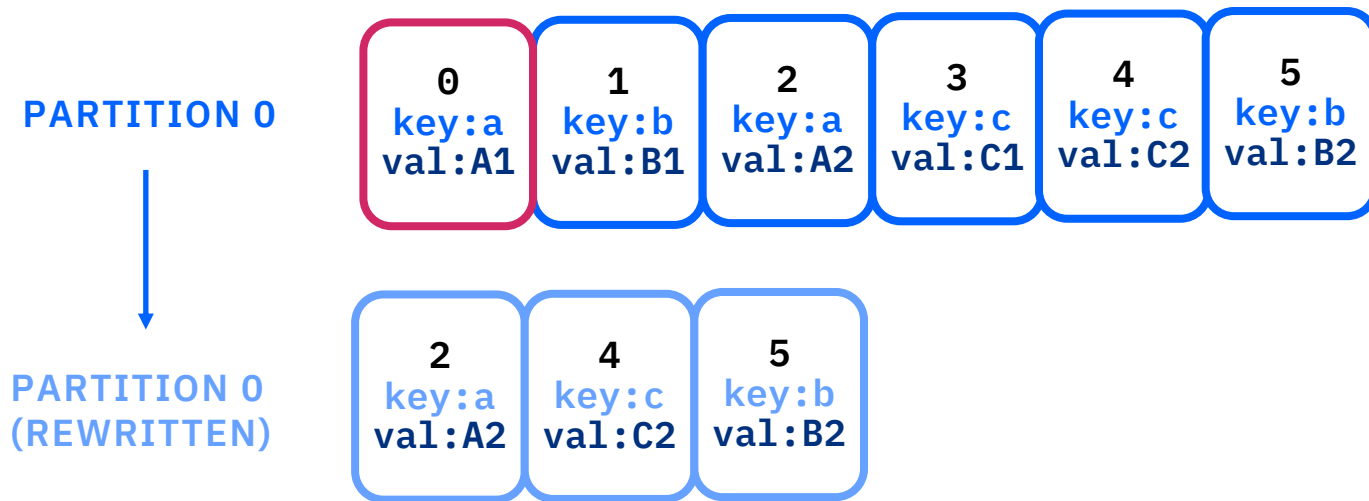
Partitions - Compaction

Compacted topics are evolving data stores



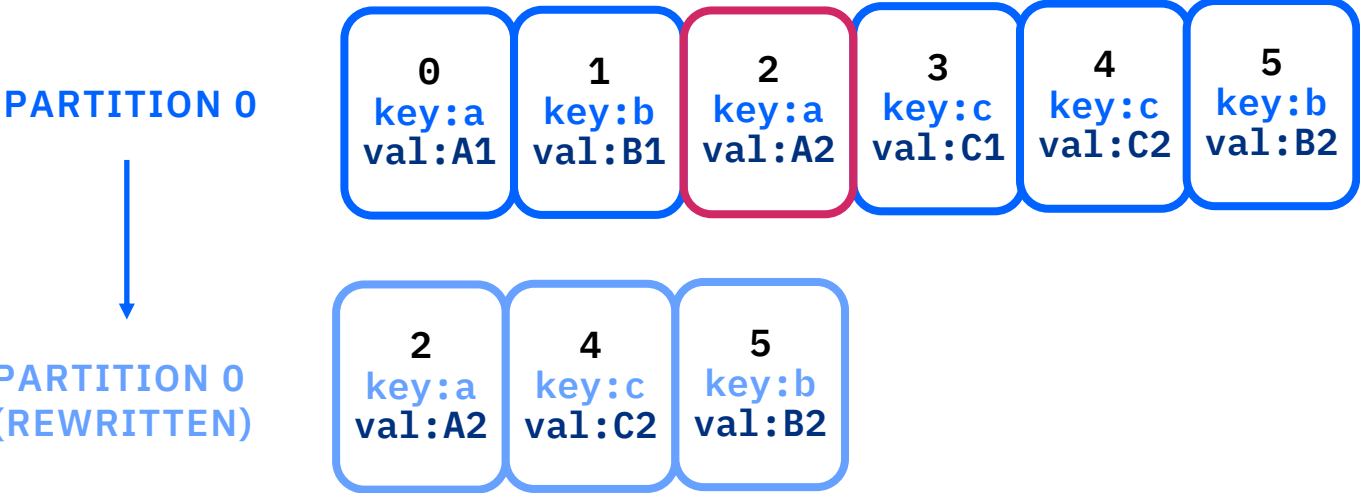
Partitions - Compaction

Compacted topics are evolving data stores



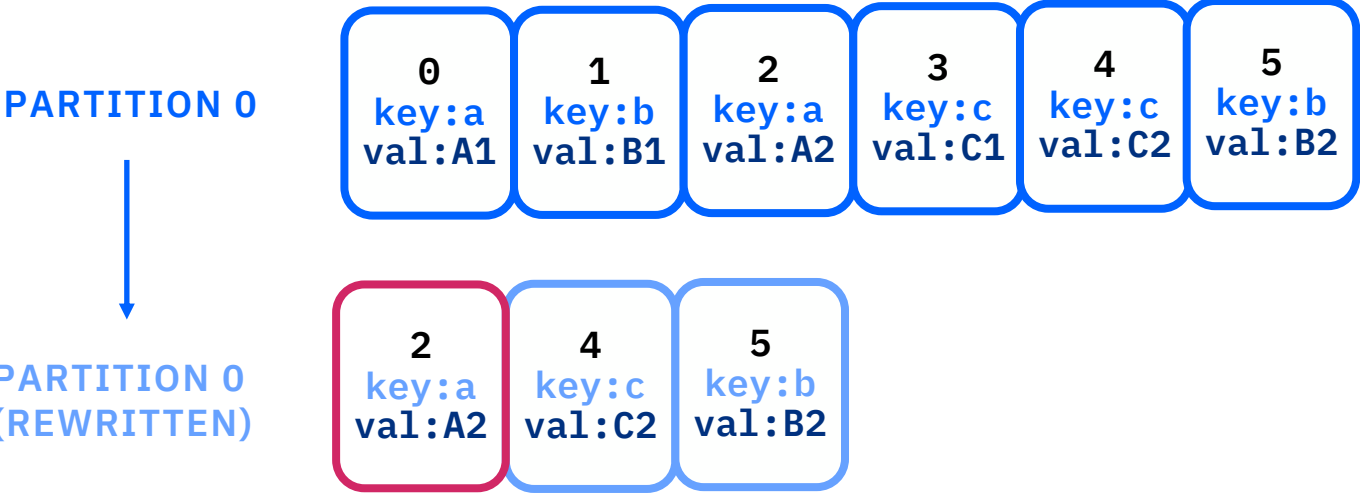
Partitions - Compaction

Compacted topics are evolving data stores



Partitions - Compaction

Compacted topics are evolving data stores



Configuring producers

Producer can choose acknowledgement level:

- 0** Fire-and-forget
Fast, but risky
- 1** Waits for 1 broker to acknowledge
- ALL** Waits for all replica brokers to acknowledge

Configuring producers

Producer can choose whether to retry:



Do not retry

Loses messages on error



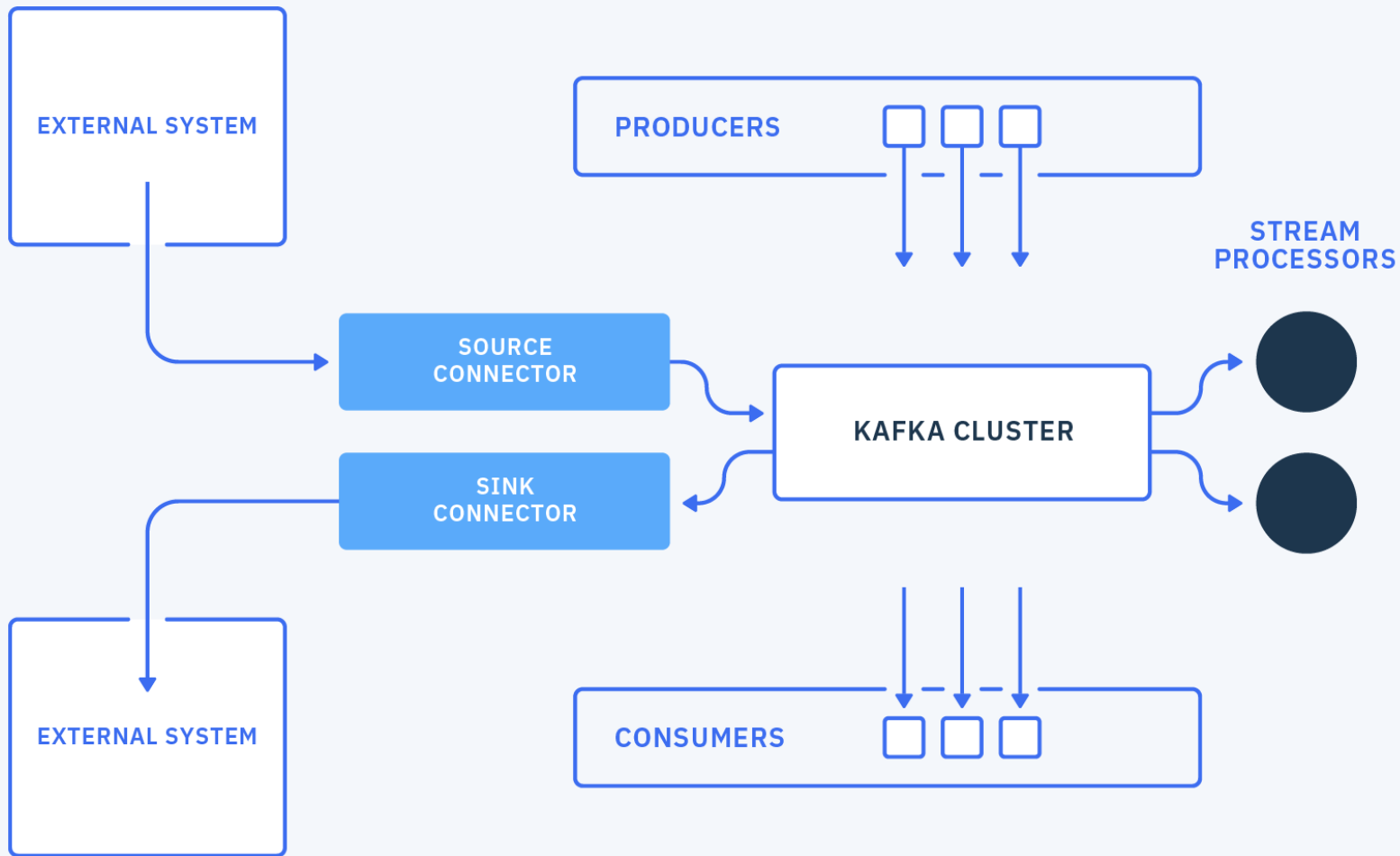
Retry

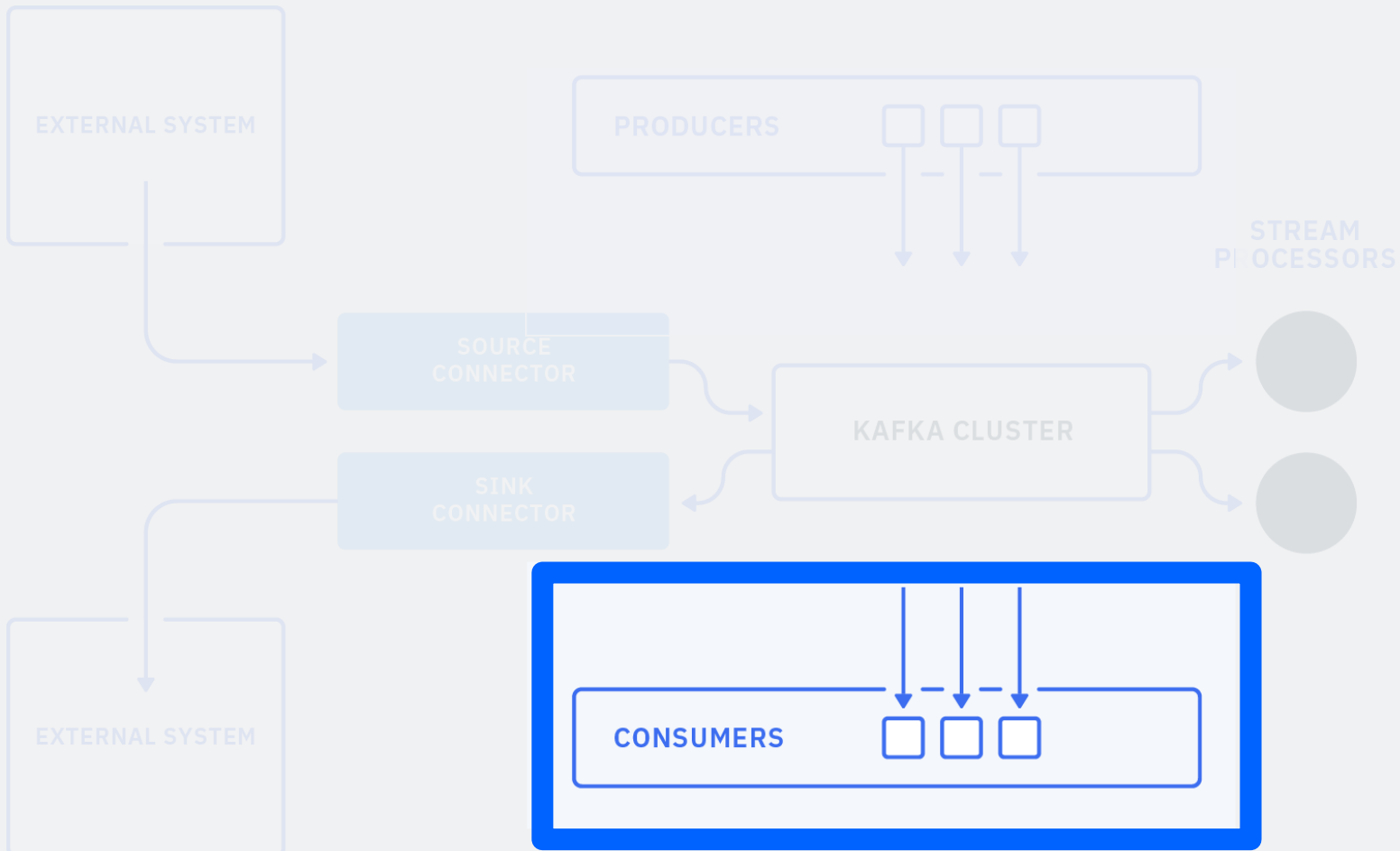
Retry, might result in duplicates on error

Producer can also choose idempotence

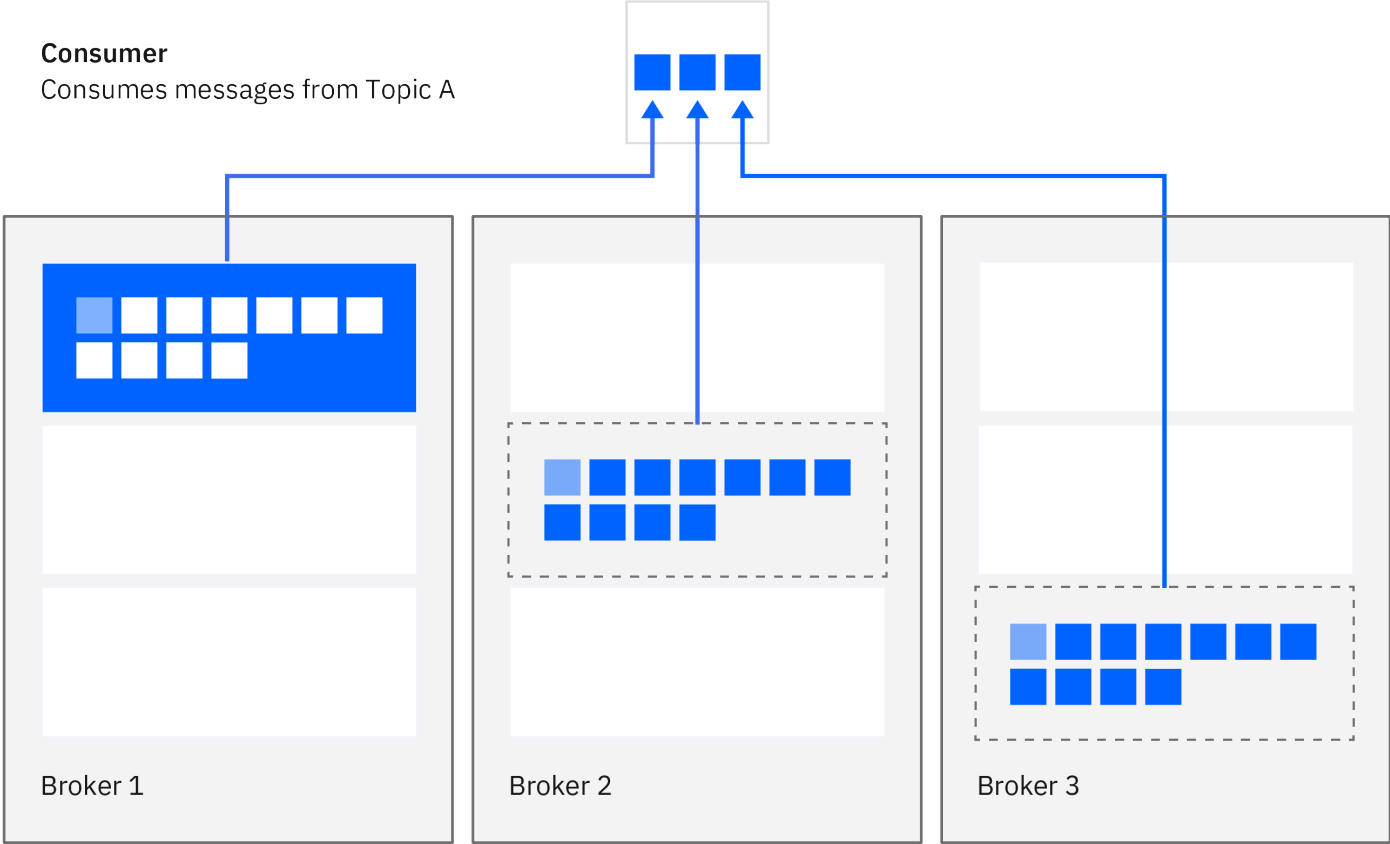
Can retry without risking duplicates

How to I get data out of Kafka?





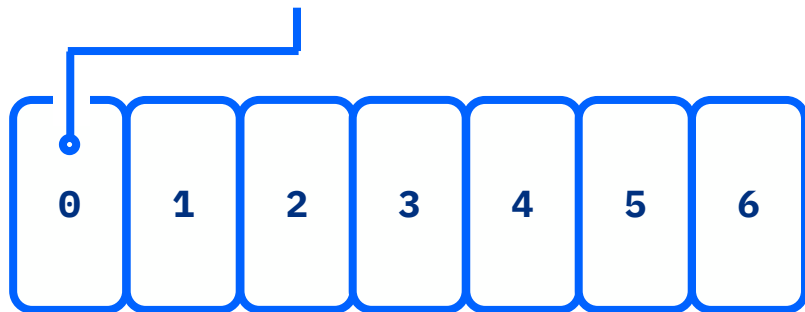
Consumers



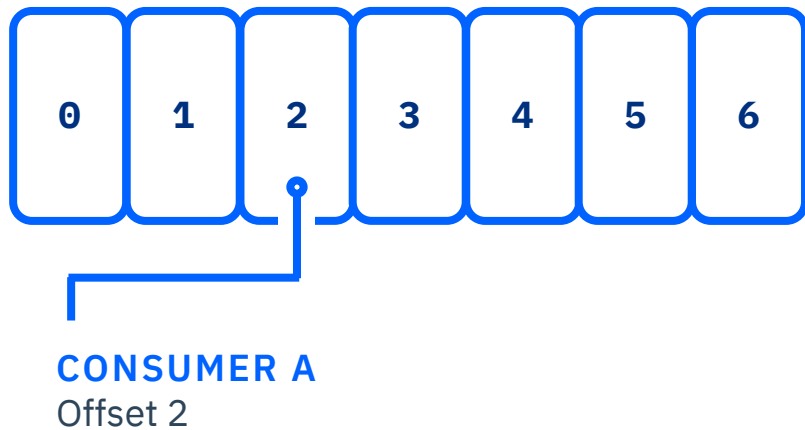
Consumers

CONSUMER A

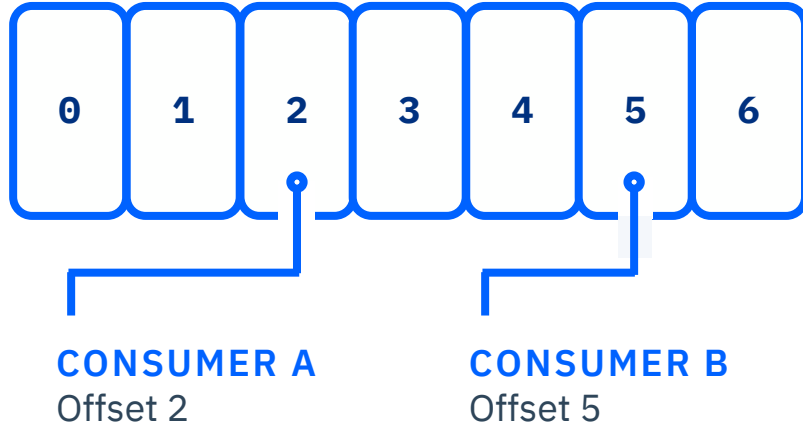
Offset 0



Consumers



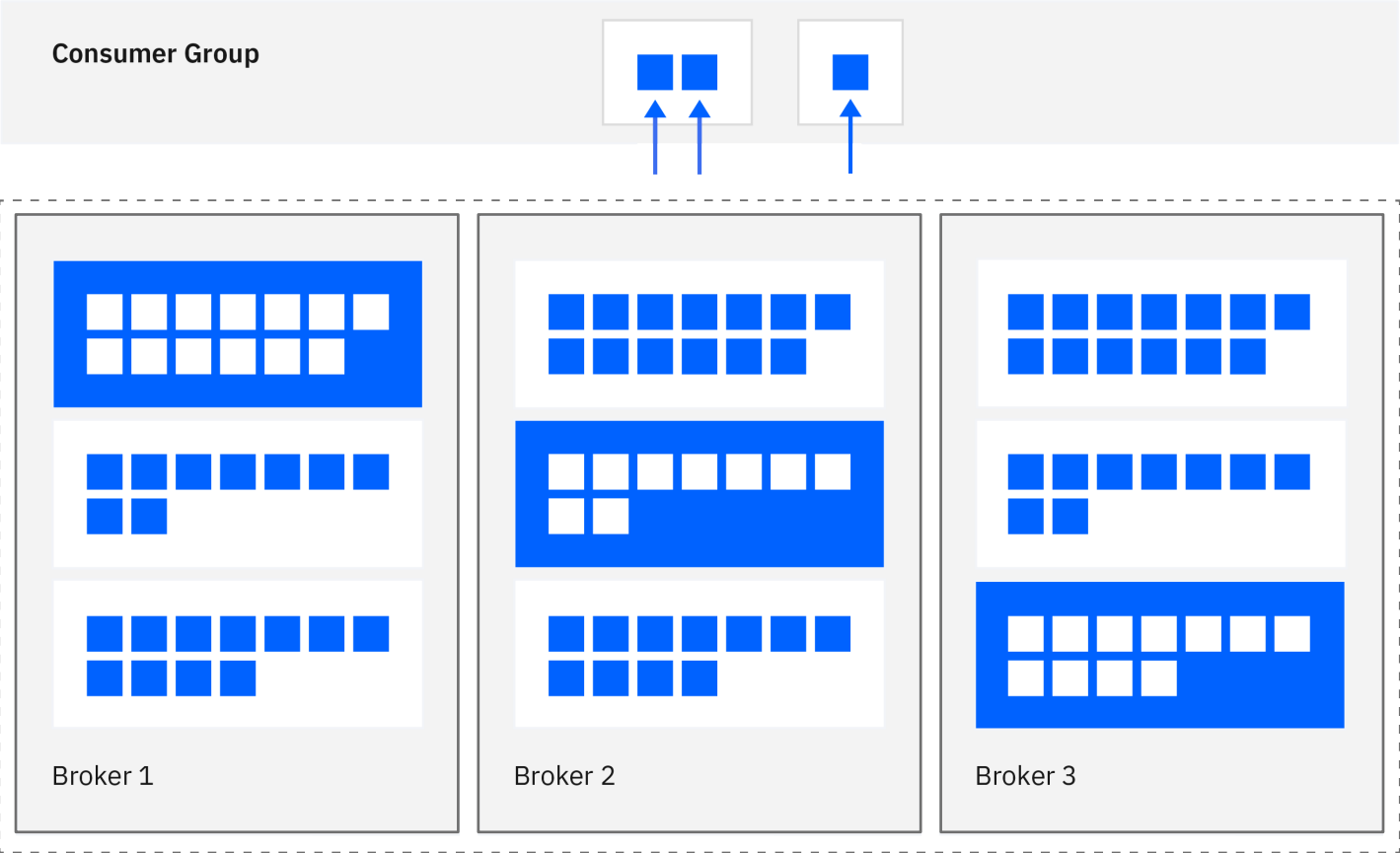
Consumers



Records remain on the topic

Allows for parallel, scalable consumption

Consumer Groups



Consumer Groups

TOPIC

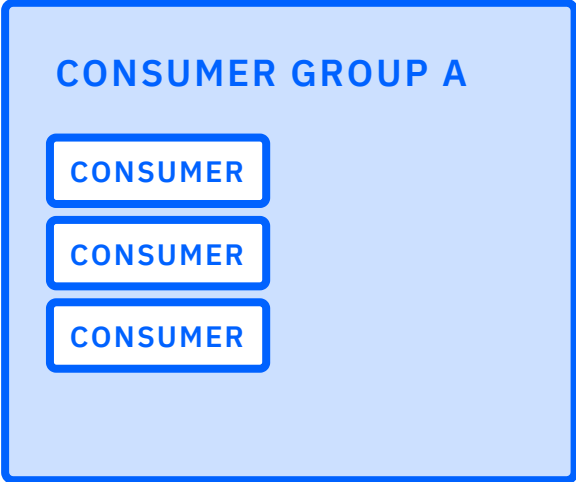
PARTITION 0



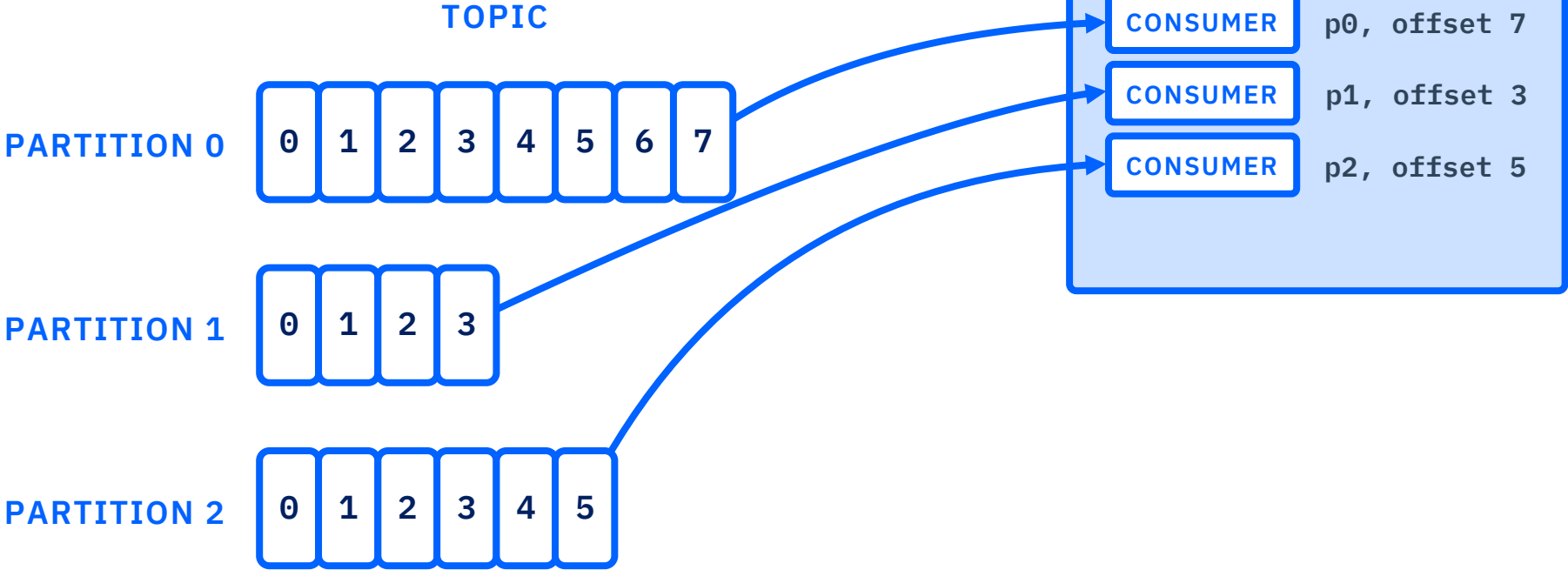
PARTITION 1



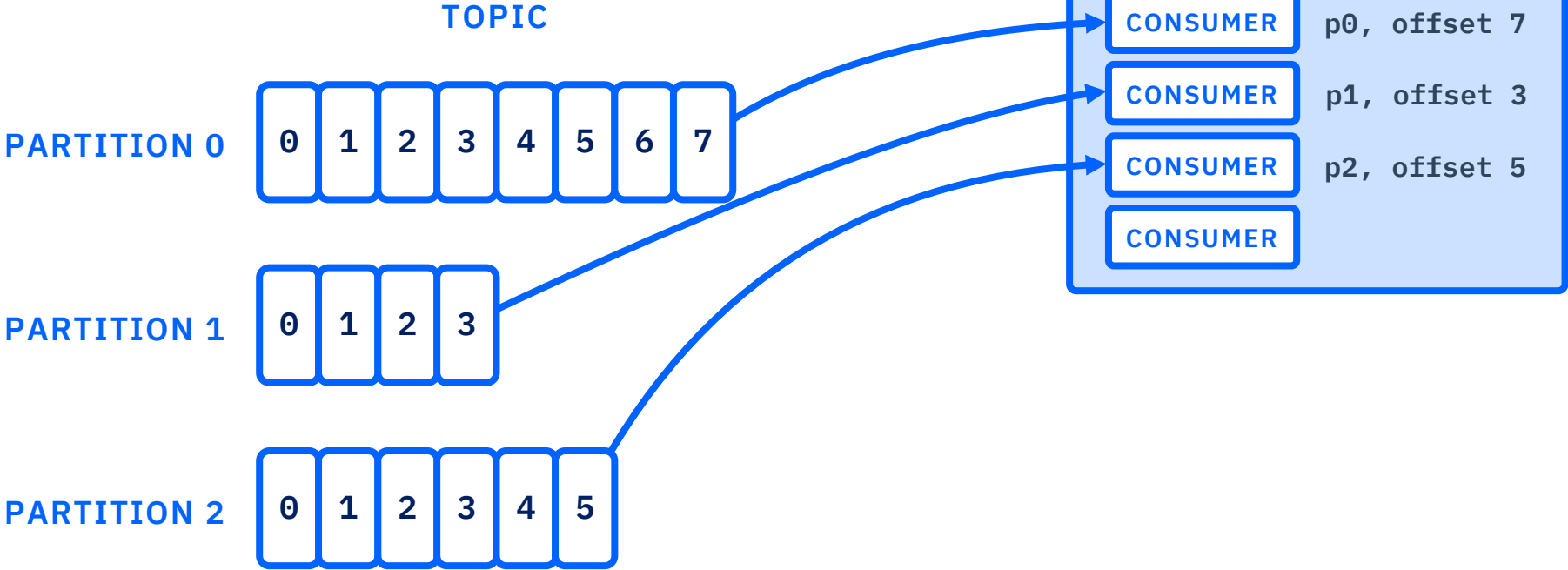
PARTITION 2



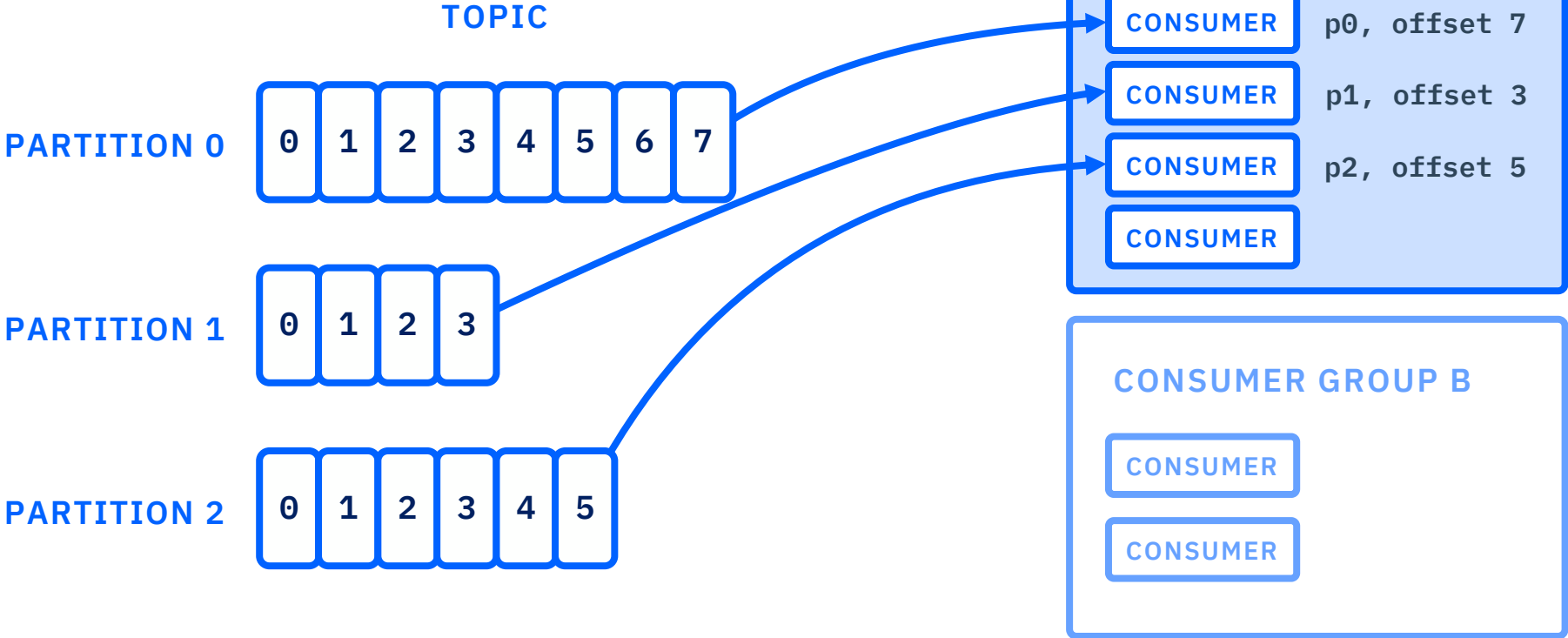
Consumer Groups



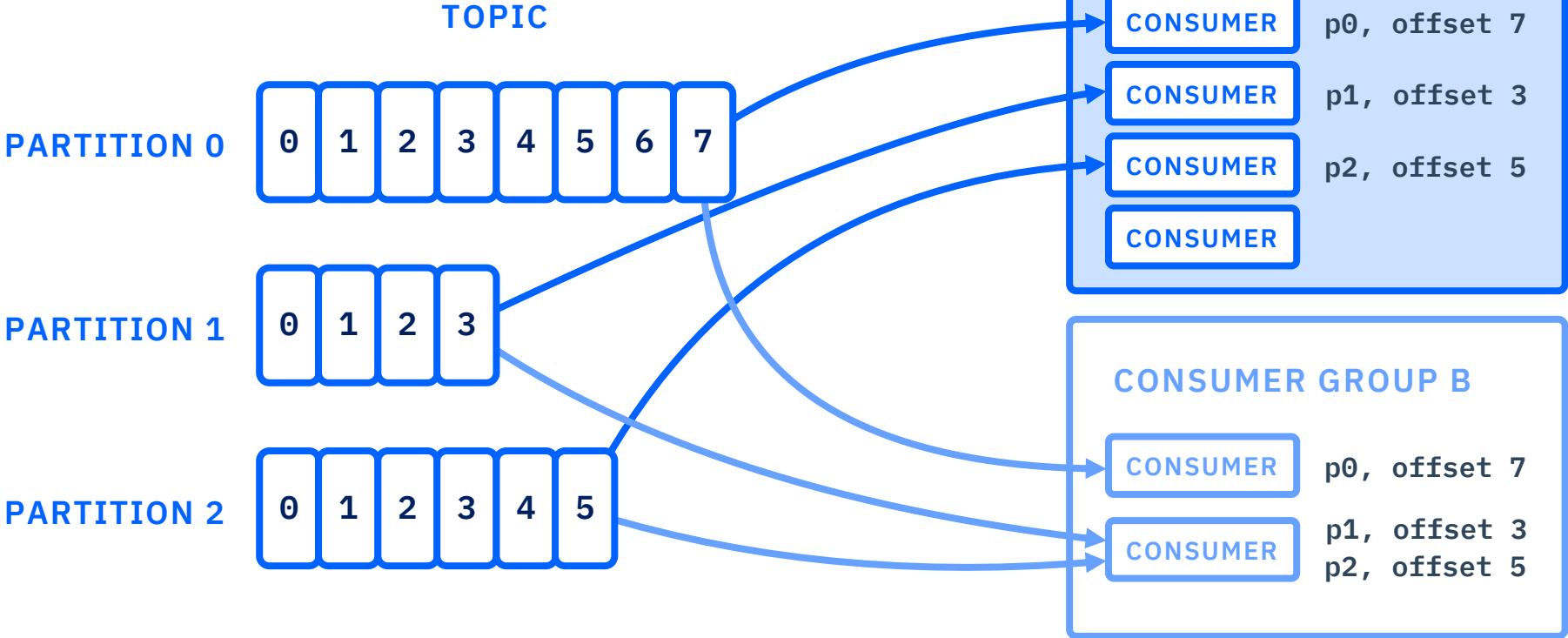
Consumer Groups



Consumer Groups



Consumer Groups



Configuring consumers

Consumer can choose how to commit offsets:

Automatic

Commits might go faster than processing

Manual,
asynchronous

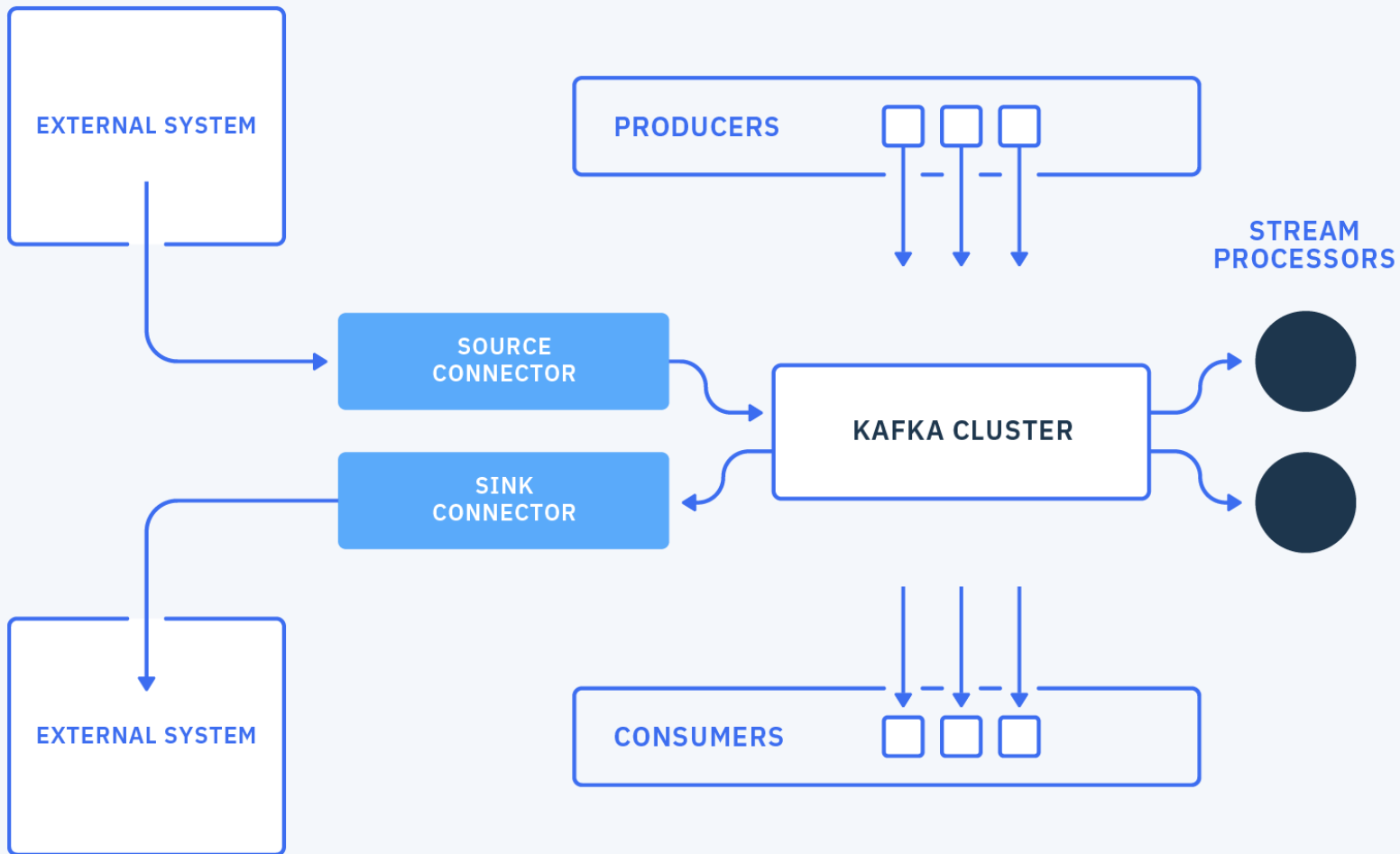
Fairly safe, but could re-process messages

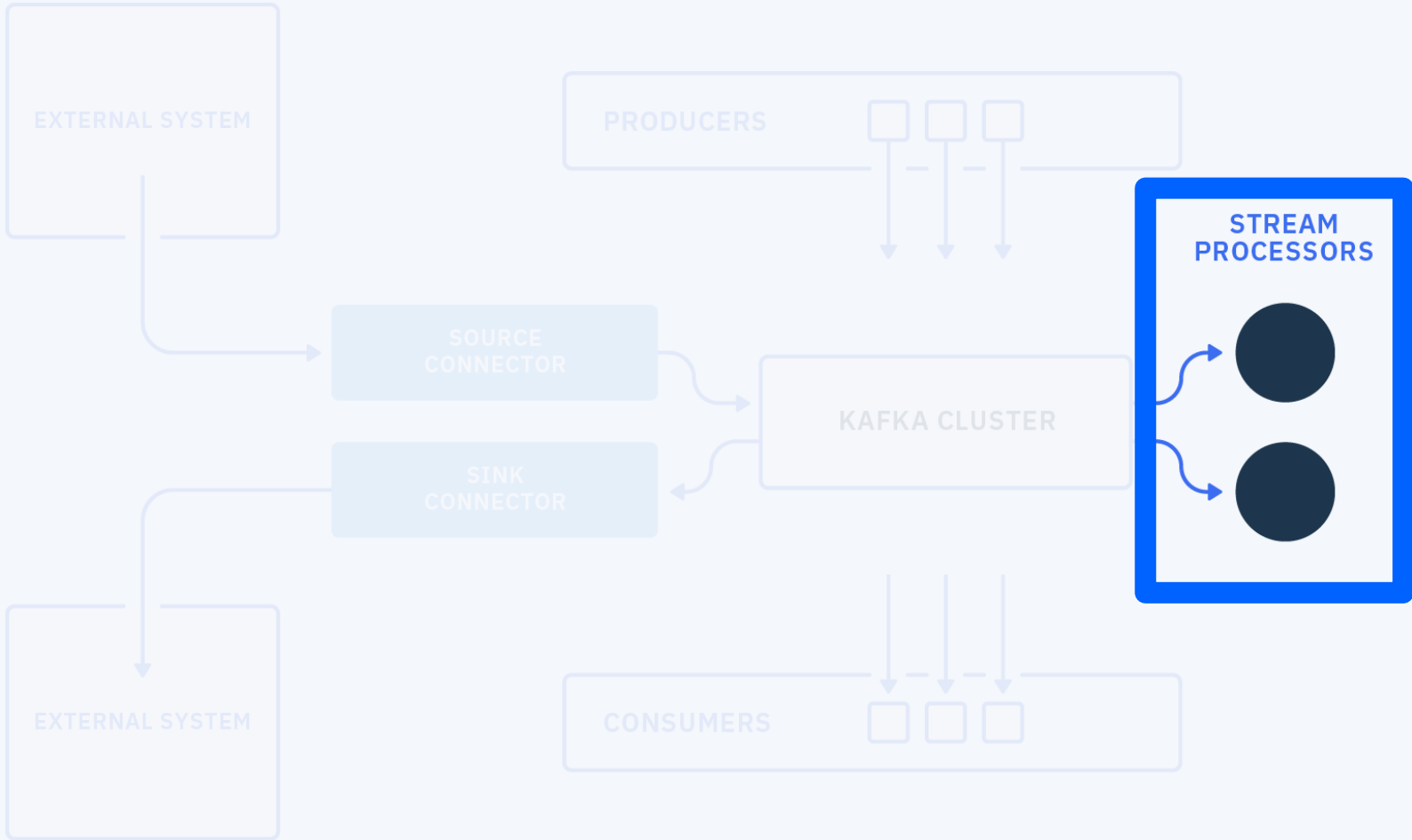
Manual,
synchronous

Safe, but slows down processing

A common pattern is to commit offsets on a timer

Connecting Kafka to External Systems



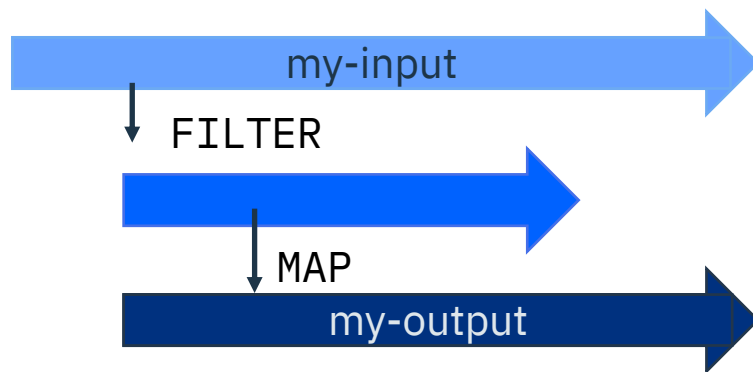


Kafka Streams

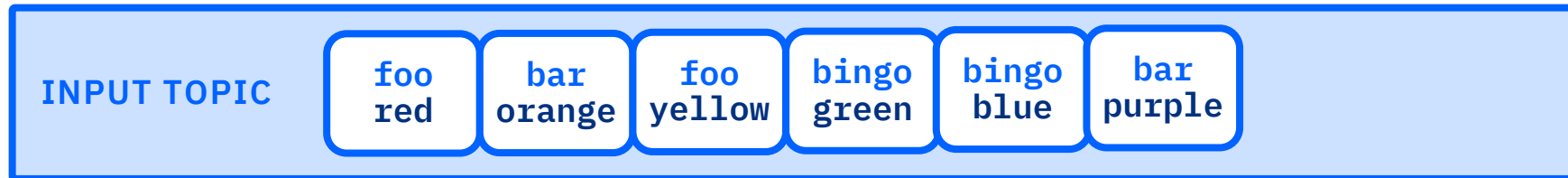
Client library for processing and analyzing data stored in Kafka

Processing happens in the app

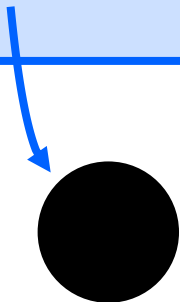
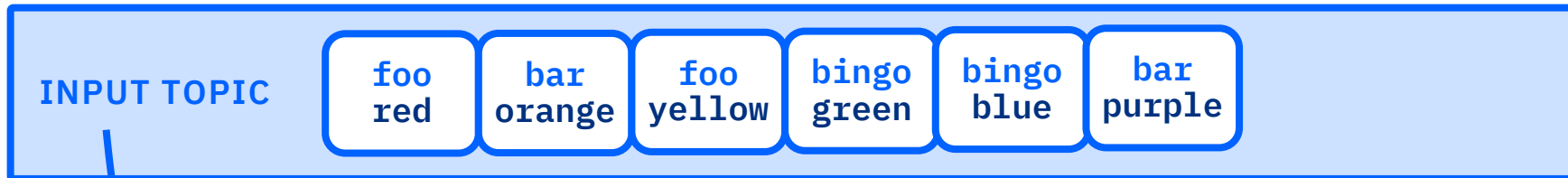
Supports per-record processing – no batching



Kafka Streams

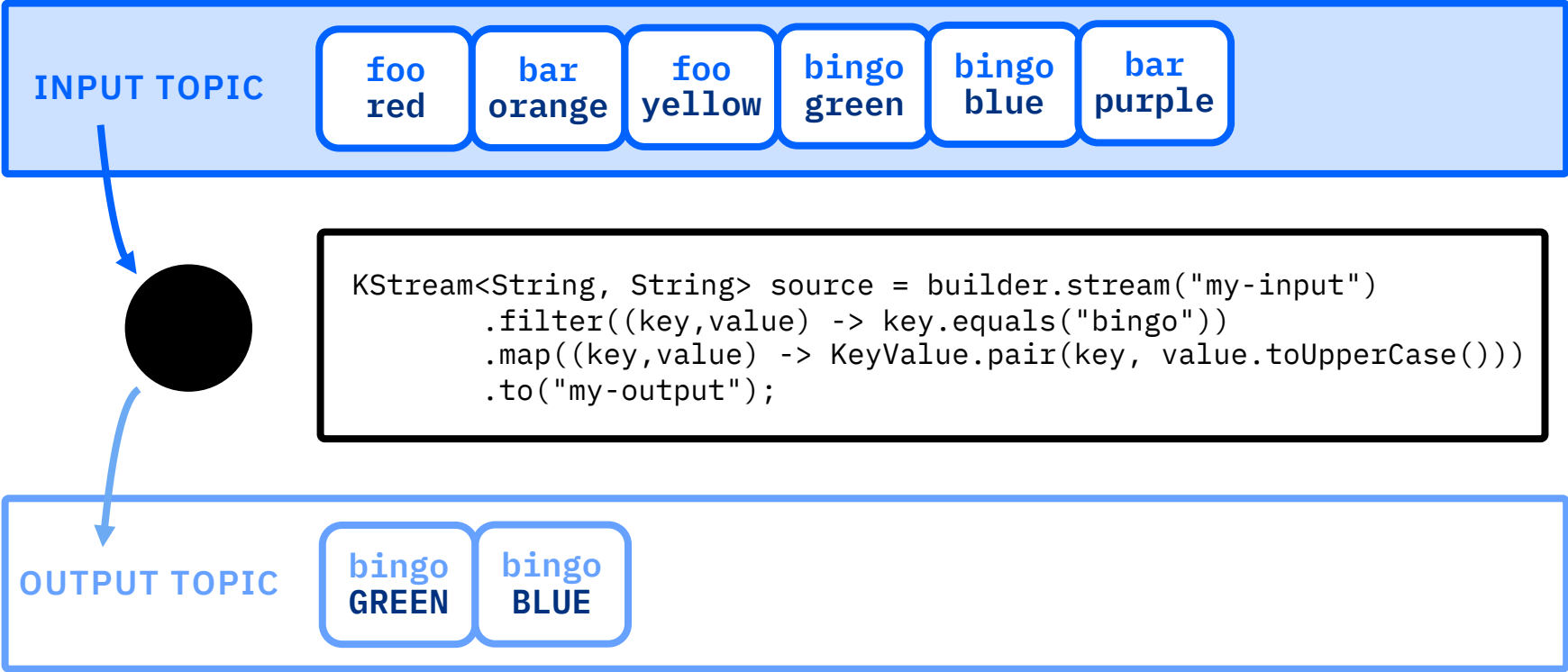


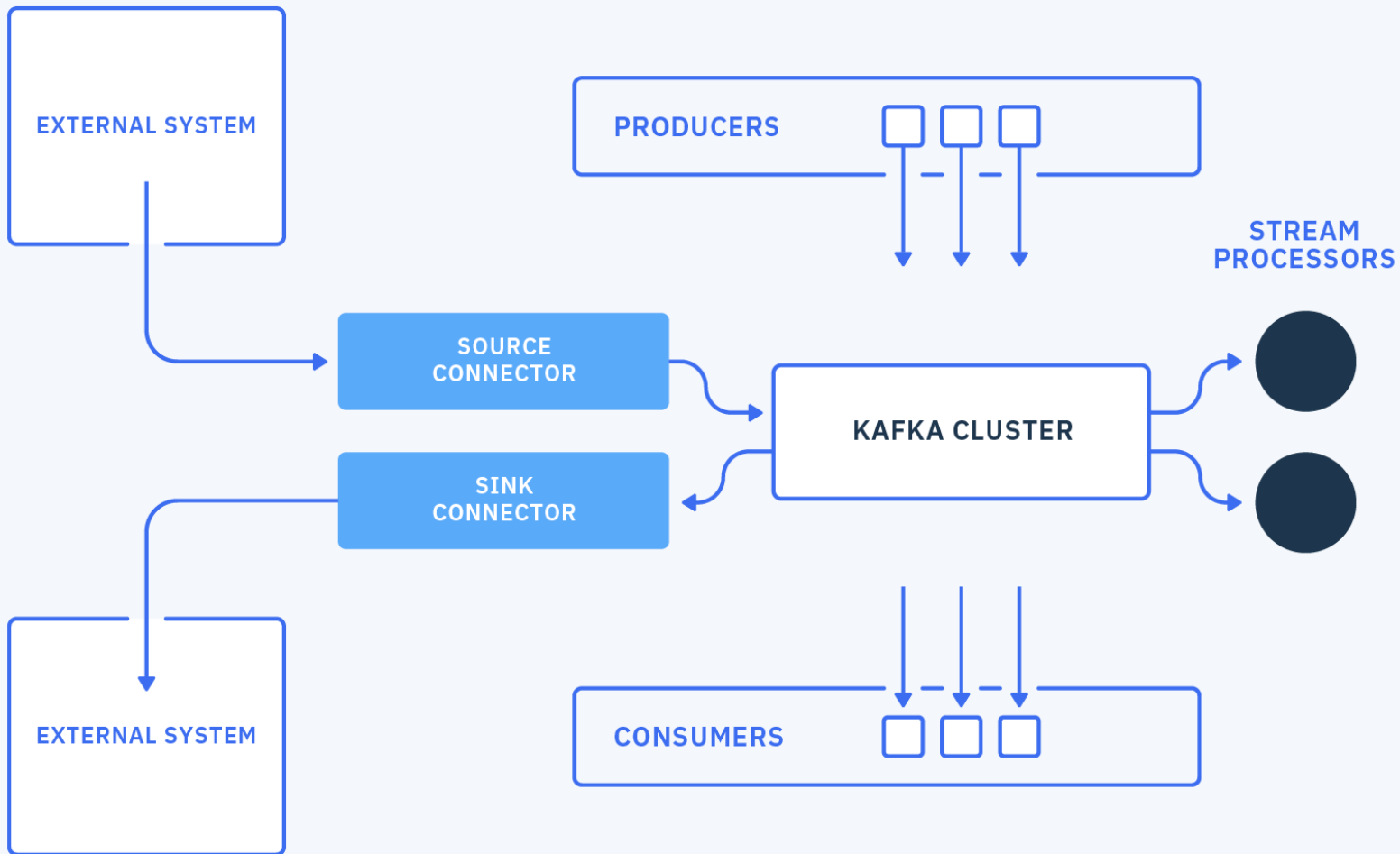
Kafka Streams

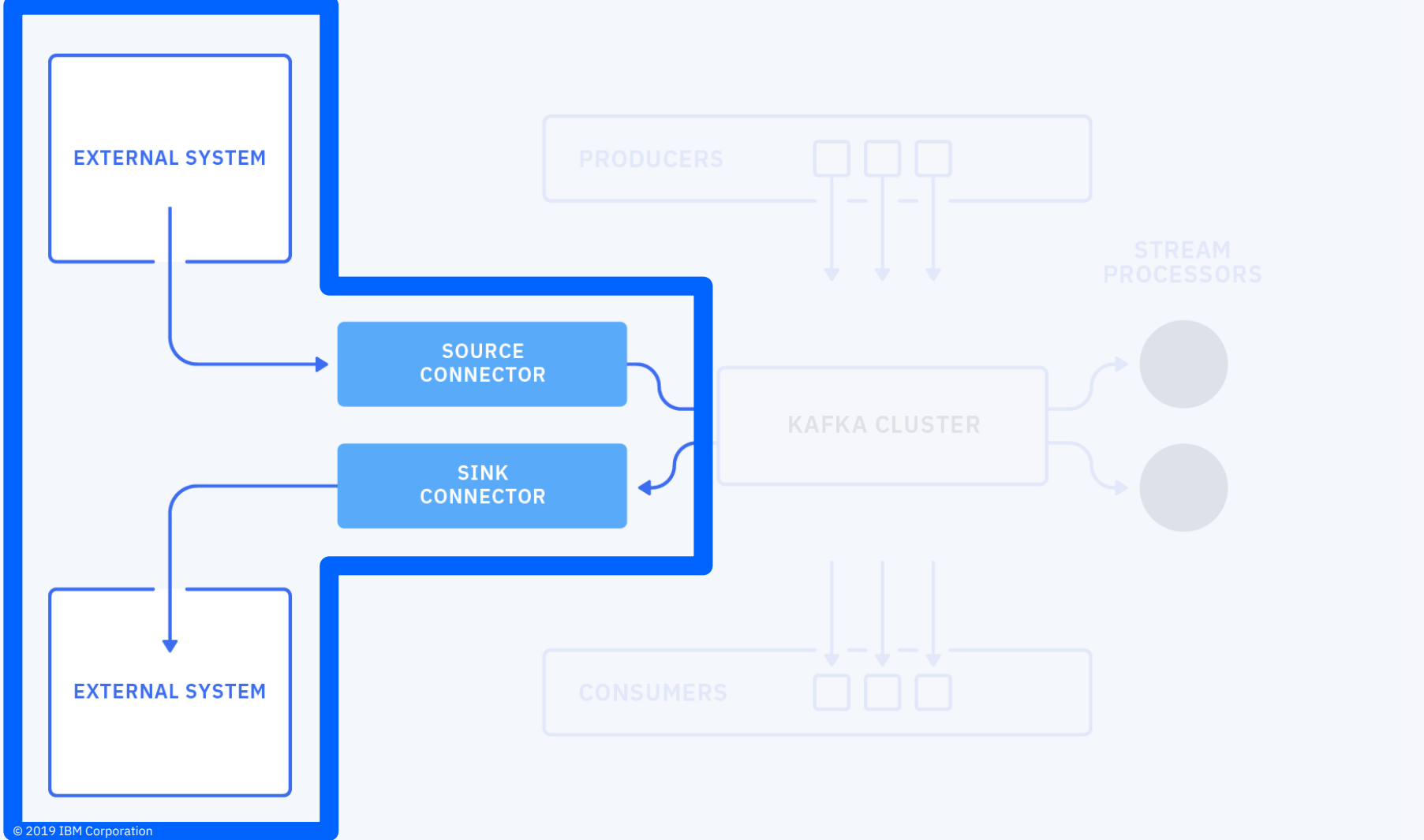


```
KStream<String, String> source = builder.stream("my-input")  
    .filter((key,value) -> key.equals("bingo"))  
    .map((key,value) -> KeyValue.pair(key, value.toUpperCase()))  
    .to("my-output");
```

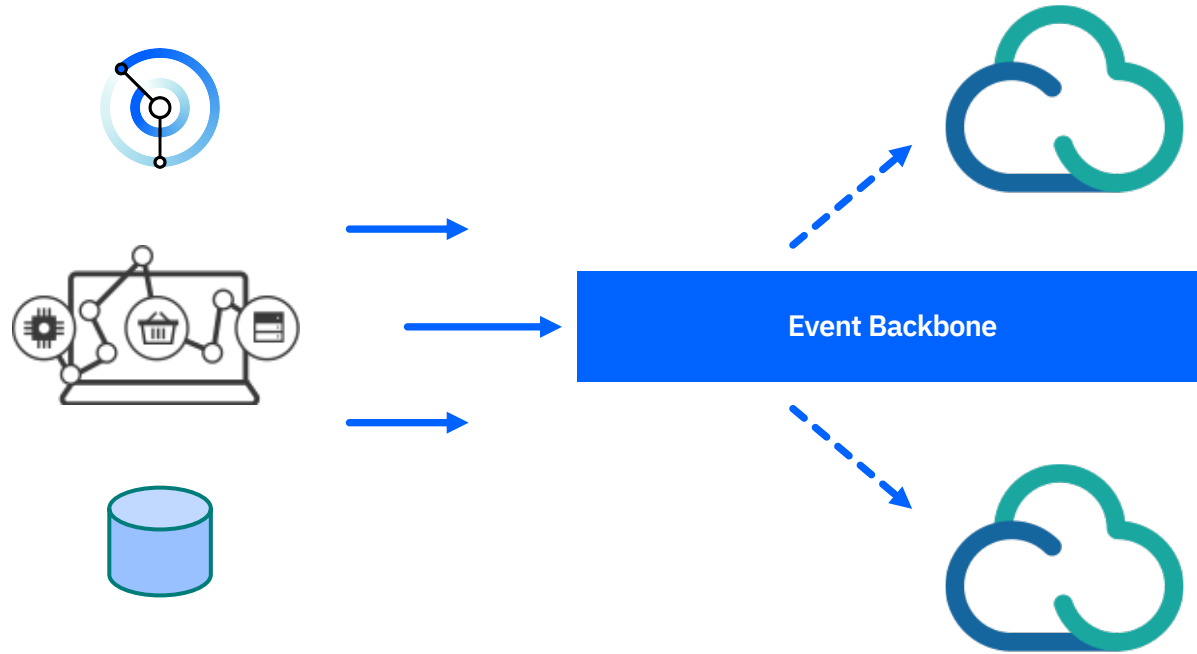

Kafka Streams



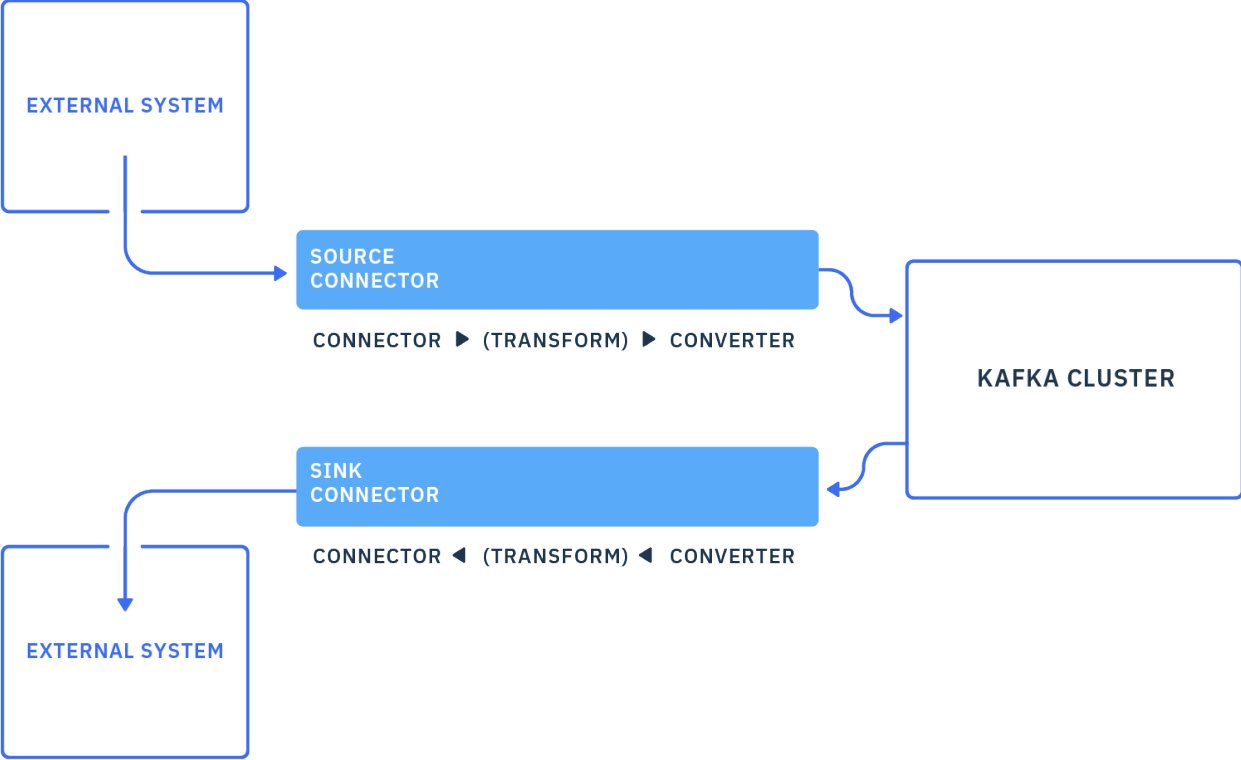




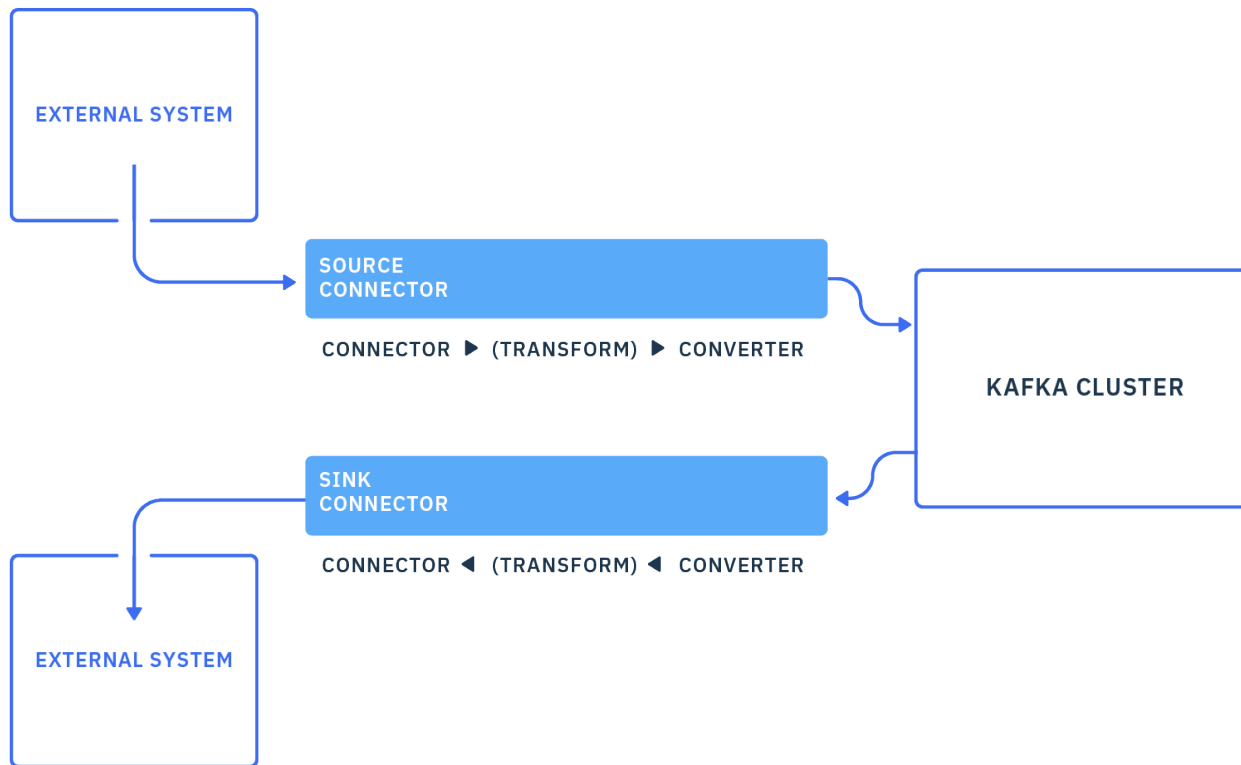
Kafka Connect – bridge to cloud-native apps



Kafka Connect



Kafka Connect



Over 80 connectors

IBM MQ

HDFS

Elasticsearch

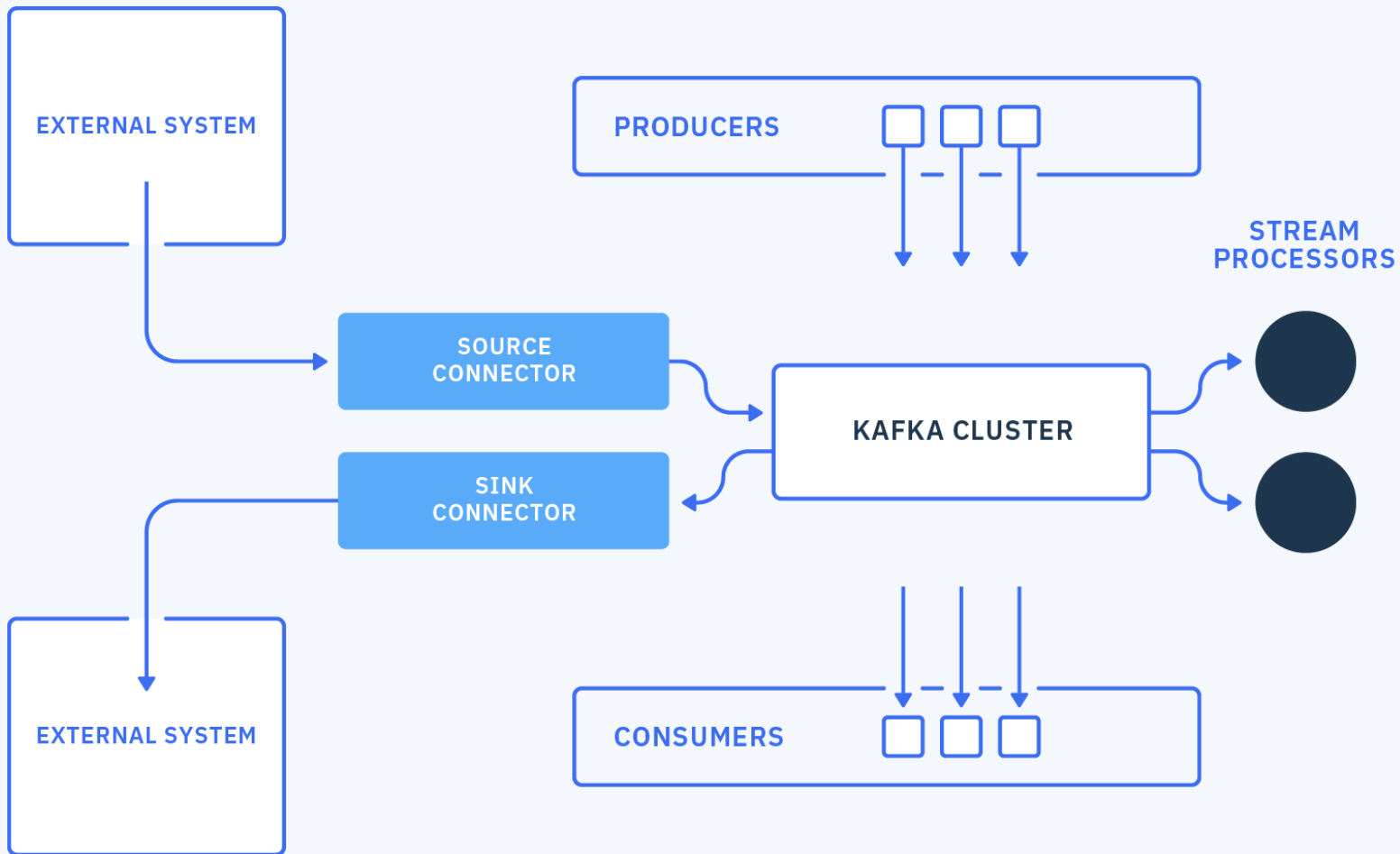
MySQL

JDBC

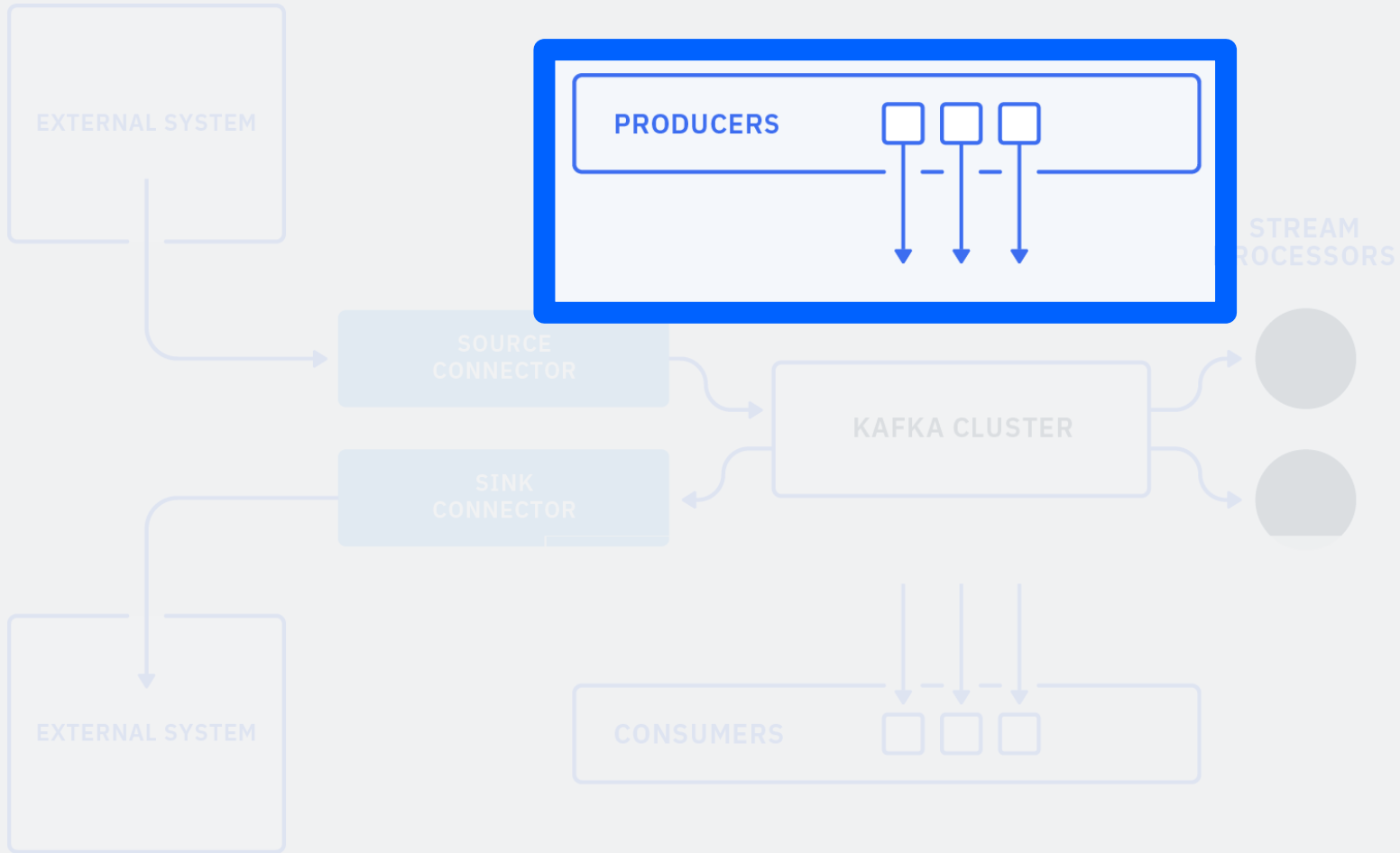
MQTT

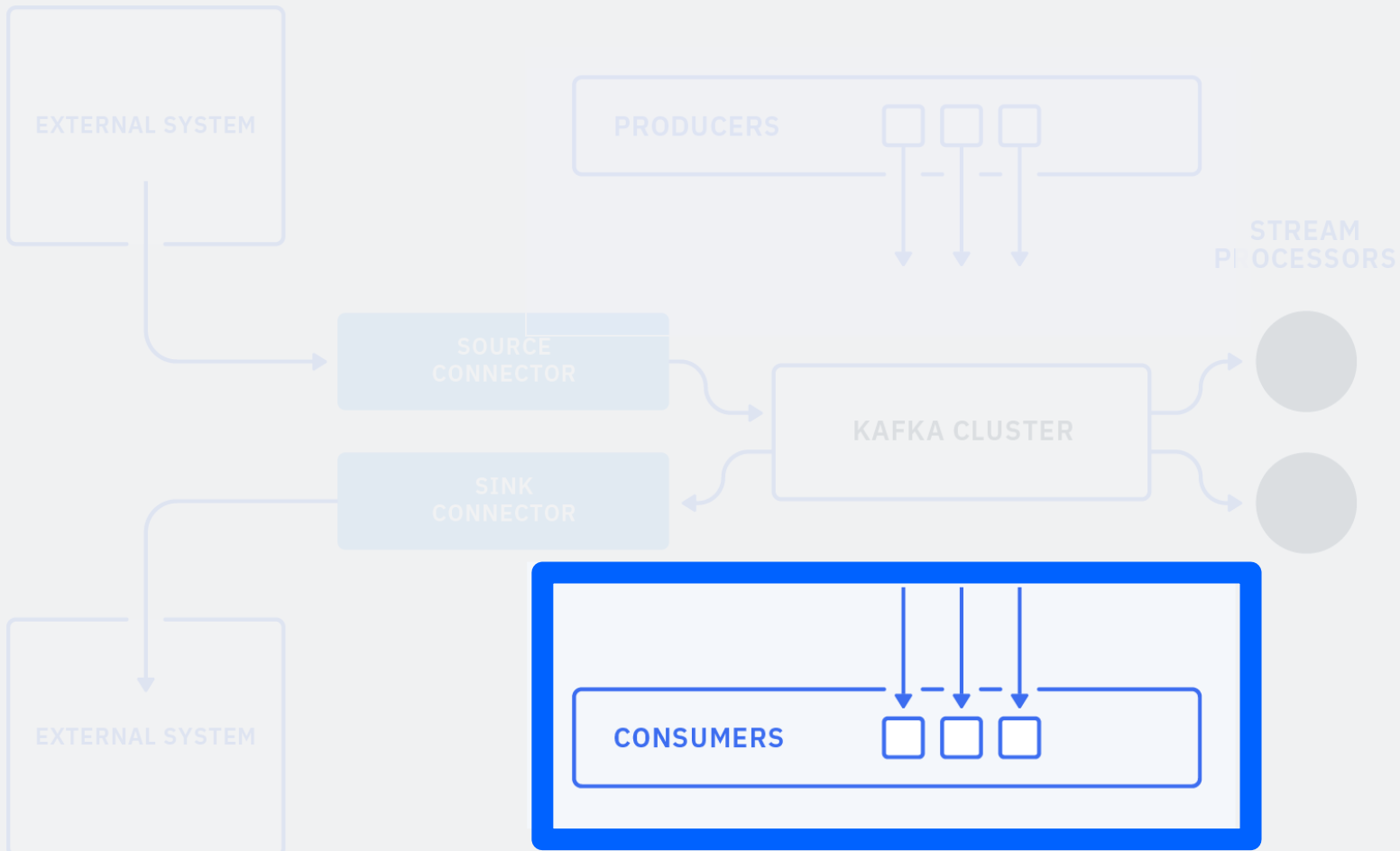
CoAP

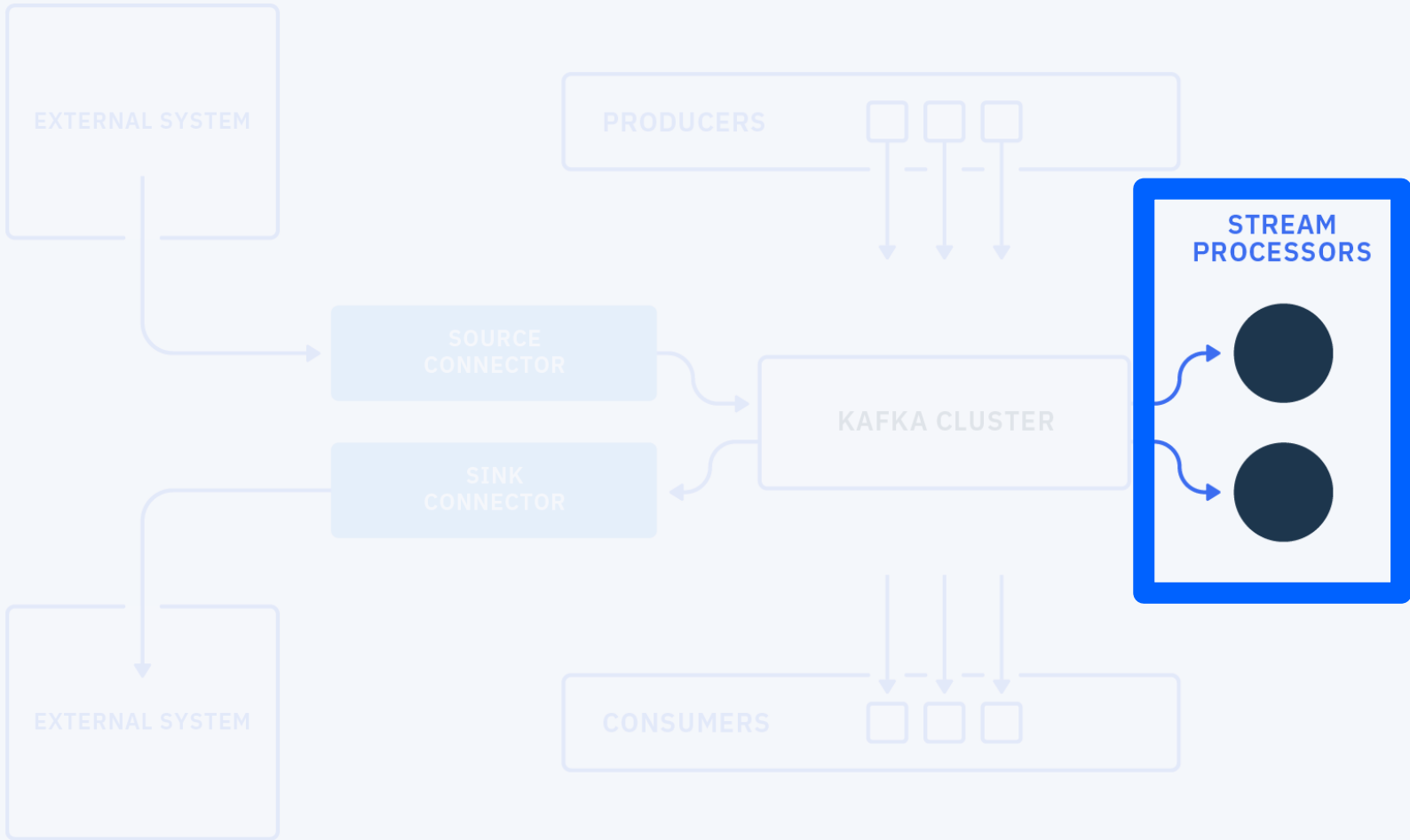
+ many others

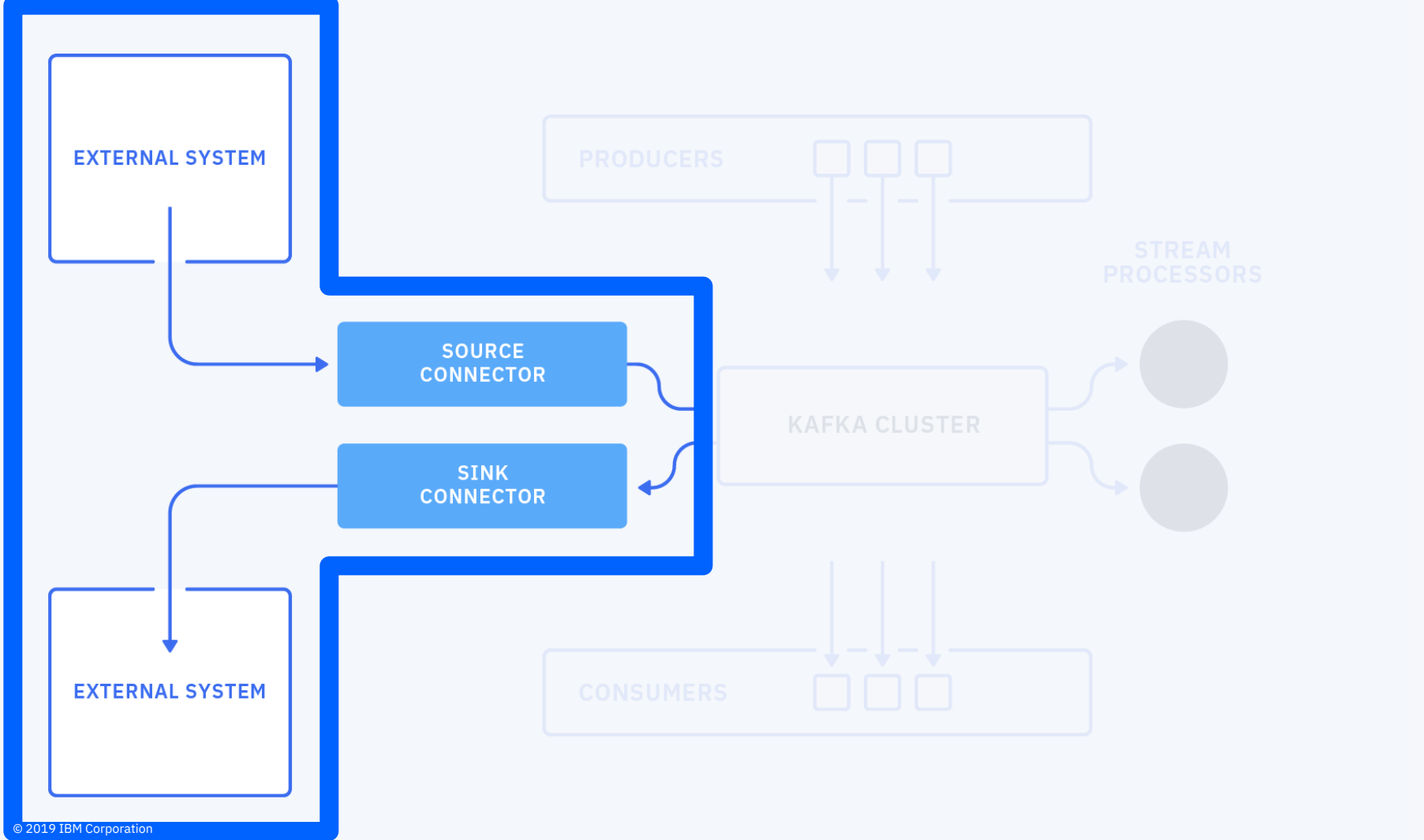


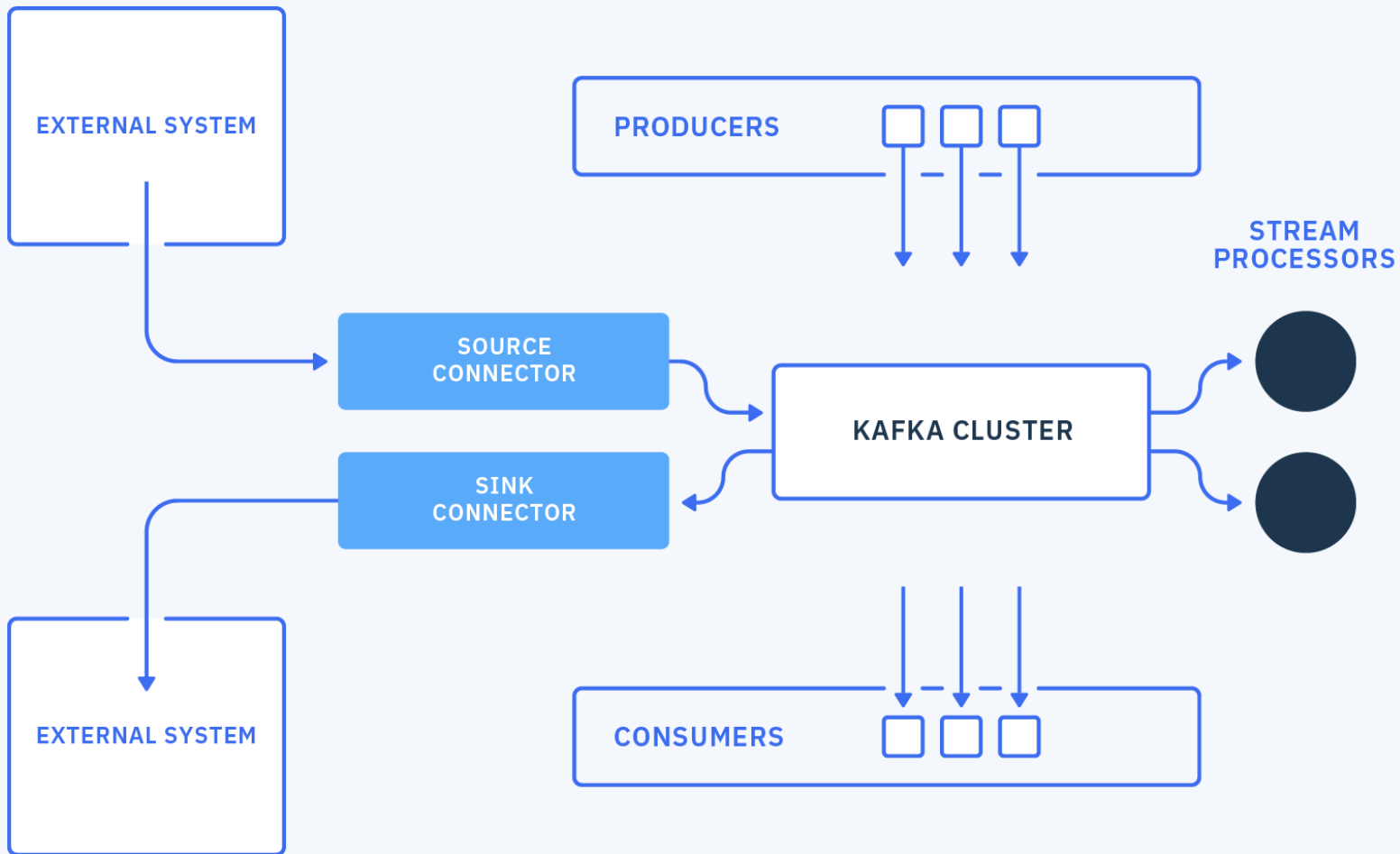














IBM Event Streams



IBM Event Streams is fully supported Apache Kafka® with value-add capabilities

IBM Event Streams

Award Winning User Experience

Powerful Ops Tooling

Schema Registry

Geo-replication for DR

Connector Catalog

Unrivalled MQ connectivity

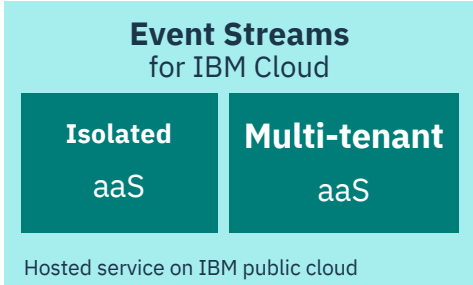
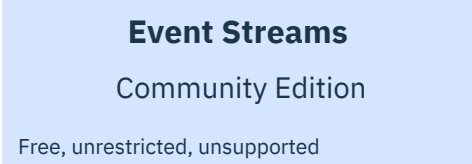
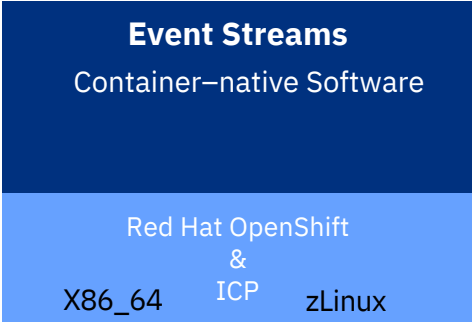
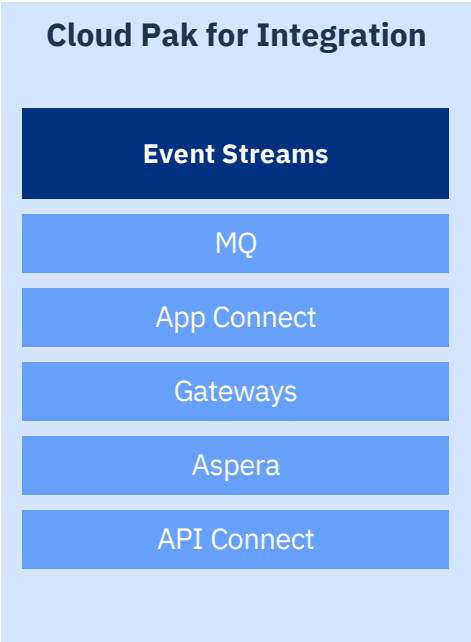
Support you can Trust



INDIGO

SILVER WINNER

Event Streams | Packaging and Deployment Options



Benefit from IBM's Kafka Expertise

IBM has years of experience running Apache Kafka across the globe



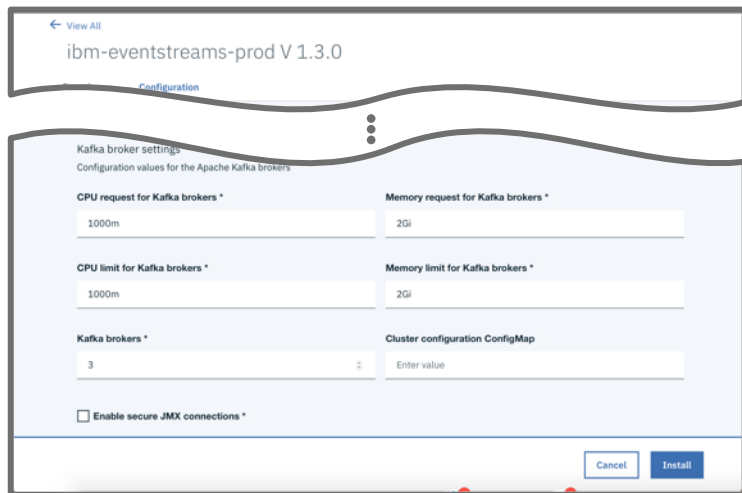
- Public Multi Tenant service
- Dedicated Single Tenant service



In 2015 IBM was the **first vendor** to offer a fully managed, Apache Kafka cloud service

IBM Event Streams | Making Apache Kafka Intuitive and Easy

Easy to deploy



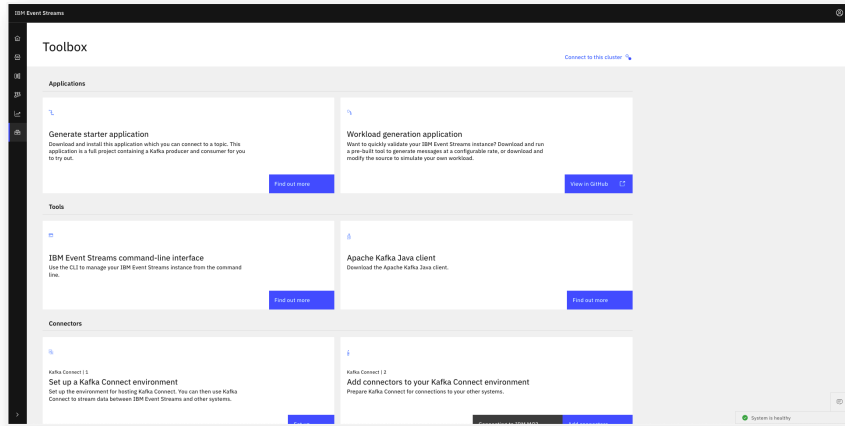
The screenshot shows a configuration page for 'ibm-eventstreams-prod V 1.3.0'. The page is titled 'Configuration' and contains the following fields:

- Kafka broker settings**
Configuration values for the Apache Kafka brokers
- CPU request for Kafka brokers ***: 1000m
- Memory request for Kafka brokers ***: 2Gi
- CPU limit for Kafka brokers ***: 1000m
- Memory limit for Kafka brokers ***: 2Gi
- Kafka brokers ***: 3
- Cluster configuration ConfigMap**: Enter value
- Enable secure JMX connections ***

At the bottom right, there are 'Cancel' and 'Install' buttons.

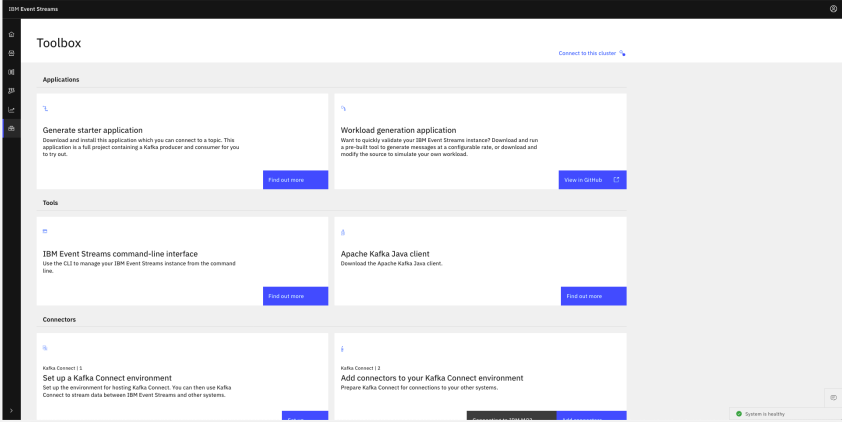
- Deploying Kafka in Kubernetes is **not easy**
- Best practices are essential to ensure production-level availability
- Ensuring consistent and repeatable deployment

Making Apache Kafka Intuitive and Easy



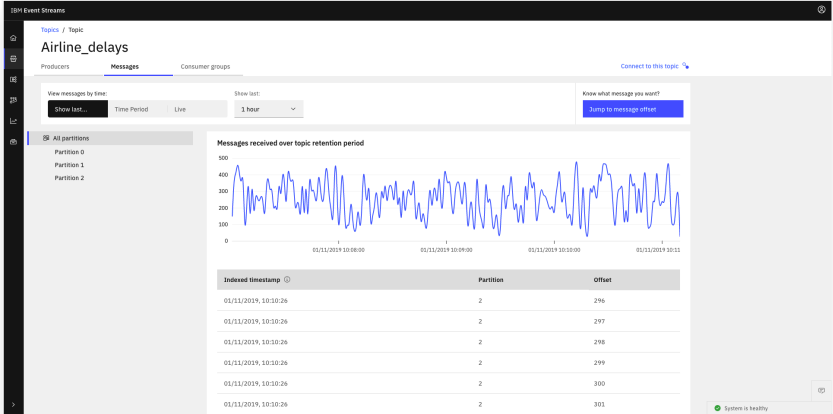
Tools to boost productivity

Making Apache Kafka Intuitive and Easy

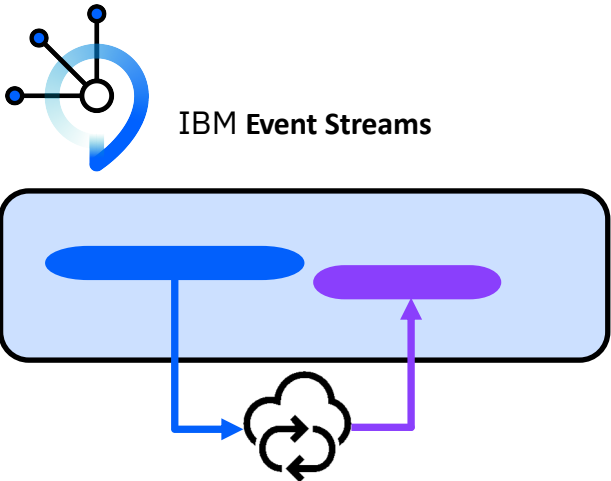


Tools to boost productivity

Visualisation of your topic data



Integrated with Key Monitoring Tools



External monitoring tools
Datadog, Splunk, etc

IBM Event Streams | **Enterprise-Grade Reliability**

Integrated geo-replication for disaster recovery



Disaster Recovery with Geo-replication

1

Geo-replication
Geo-replication is a way to duplicate your topics to different instances of Event Streams. These are usually located in different locations to minimize the risk of data loss in the event of a cluster failure.

Destination local... Origin locations

Destination locations

Add destination cluster +

Destination cluster ID	Topics	Workers
TestDest1	10	2
TestDest2	6	4
TestDest3	2	2

2

Geo-replication / Destination cluster
Destination cluster ID: TestDest3
Topics: 2 Workers: 2

5 topics are being geo-replicated from this cluster

Name	Replication Health
Stoker_West_topic_3	Awaiting creation
MissingMan_North_topic_4	Awaiting creation
Stoker_North_topic_0	Awaiting creation
MissingMan_East_topic_1	Running
Trial_South_topic_2	Error

Geo-replication has not been configured on this destination.

You are about to create replicators for 3 topics.
 Add prefix to destination topic names
 Include message history names
 CANCEL Create

3

Geo-replication / Destination cluster
Destination cluster ID: TestDest1
Topics: 10 Workers: 2

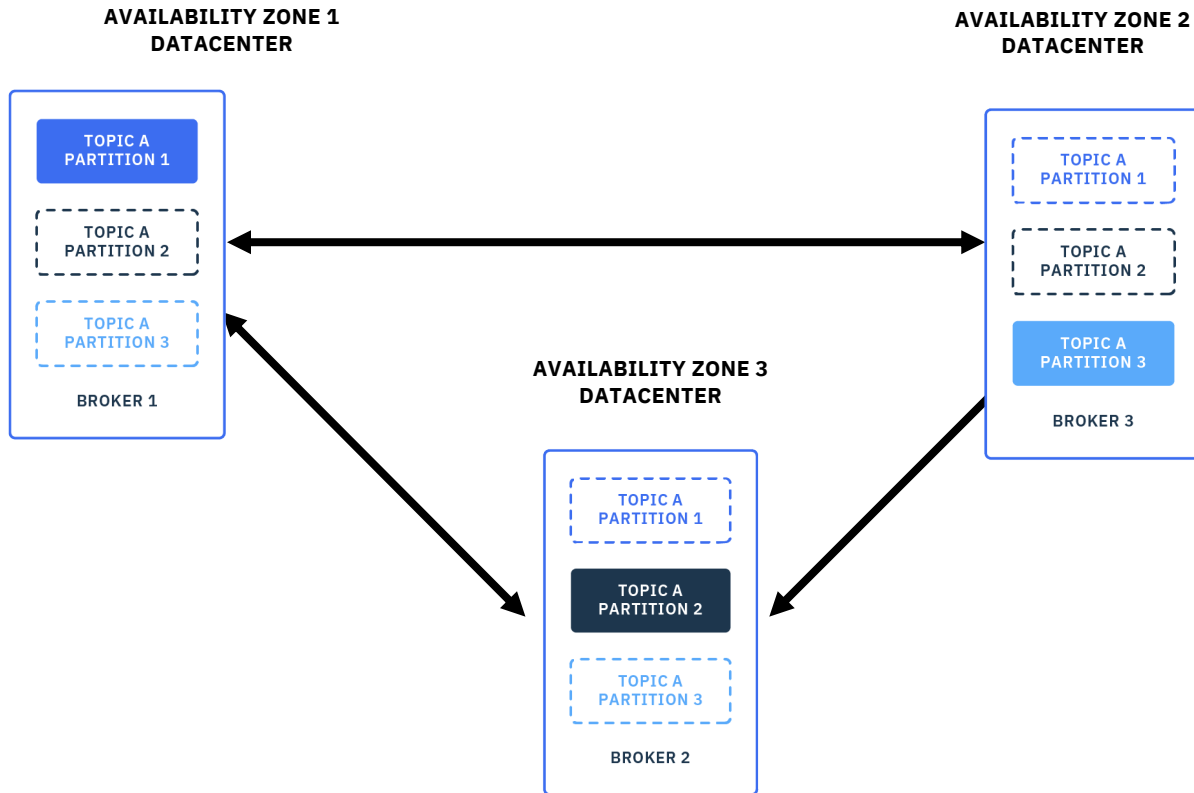
10 topics are being geo-replicated from this cluster

Name	Replication Health
Stoker_North_topic_0	Running
MissingMan_East_topic_1	Running
Trial_South_topic_2	Running
Stoker_West_topic_3	Running
MissingMan_North_topic_4	Running
Trial_East_topic_5	Running
Stoker_South_topic_6	Running
MissingMan_West_topic_7	Running
Trial_North_topic_8	Running
Stoker_East_topic_9	Running

Topics

Name	Replicas	Partitions	Geo-replication
→ Stoker_North_topic_0	3	1	To: TestDest1 From: 3 clusters
MissingMan_East_topic_1	3	1	To: 3 clusters
Trial_South_topic_2	3	1	To: 3 clusters ⚠

Enhanced resilience with clusters across multiple zones



Multi-Availability Zone

- Must have at least 3 zones
- Kafka brokers and ZooKeeper servers span across zones
- Can tolerate failure of a zone with no service degradation
- High-speed network with low latency between zones required (< 20ms)

Develop Apps Efficiently with a Flexible Schema Registry

Avro Schema

```
{
  "namespace": "example.avro",
  "type": "record",
  "name": "User",
  "fields": [
    {
      "name": "name",
      "type": "string"
    },
    {
      "name": "favorite_number",
      "type": [
        "int",
        "null"
      ]
    },
    {
      "name": "favorite_color",
      "type": [
        "string",
        "null"
      ]
    }
  ]
}
```

Schema Registry

Name	Latest version
User	1.0.0

Schema Registry / Schema

User

Manage schemas

Versions

Add new version

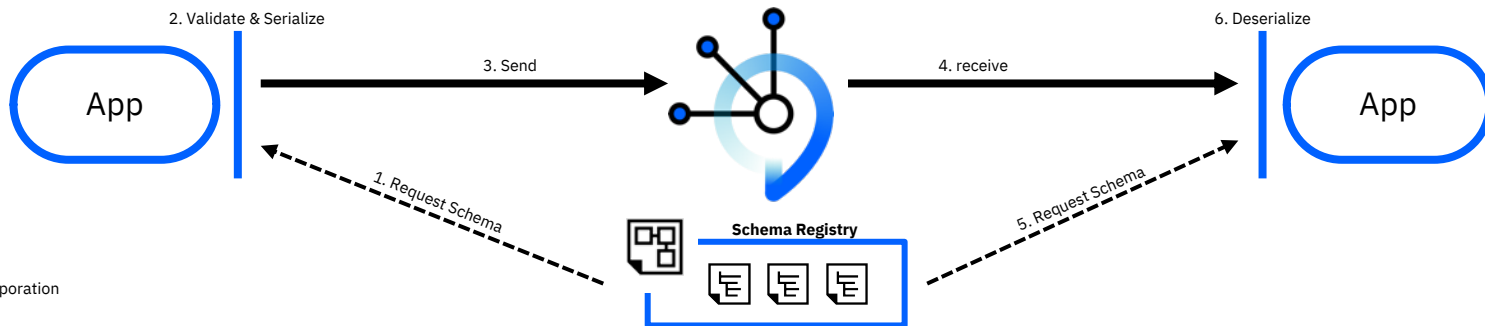
Version 1.0.0

Version 1.0.0

Created: 10/10/2019, 14:29:54 | Status: Active

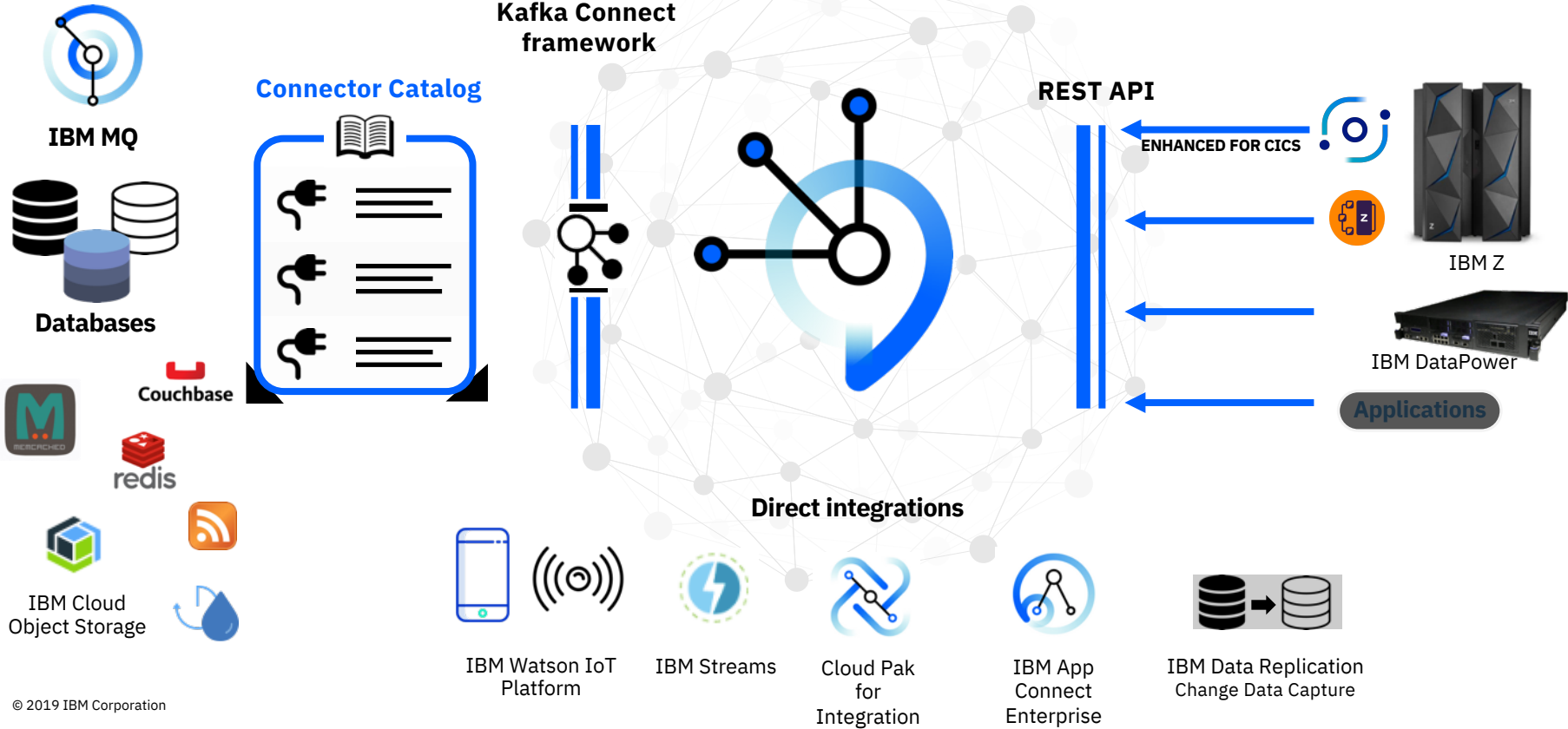
Schema definition

```
{
  "namespace": "example.avro",
  "type": "record",
  "name": "User",
  "fields": [
    {
      "name": "name",
      "type": "string"
    },
    {
      "name": "favorite_number",
      "type": [
        "int",
        "null"
      ]
    },
    {
      "name": "favorite_color",
      "type": [
        "string",
        "null"
      ]
    }
  ]
}
```



Use Existing Data in New Ways that Yield Competitive Advantage

Unmatched Connectivity to Core Systems



Welcome to the IBM Event Streams









Connector catalog

Kafka Connect is a framework for connecting Kafka to external systems. It uses source connectors to move data into Kafka, and sink connectors to move data out of Kafka.

The Event Streams connector catalog contains a list of tried and tested connectors from both the community and IBM.

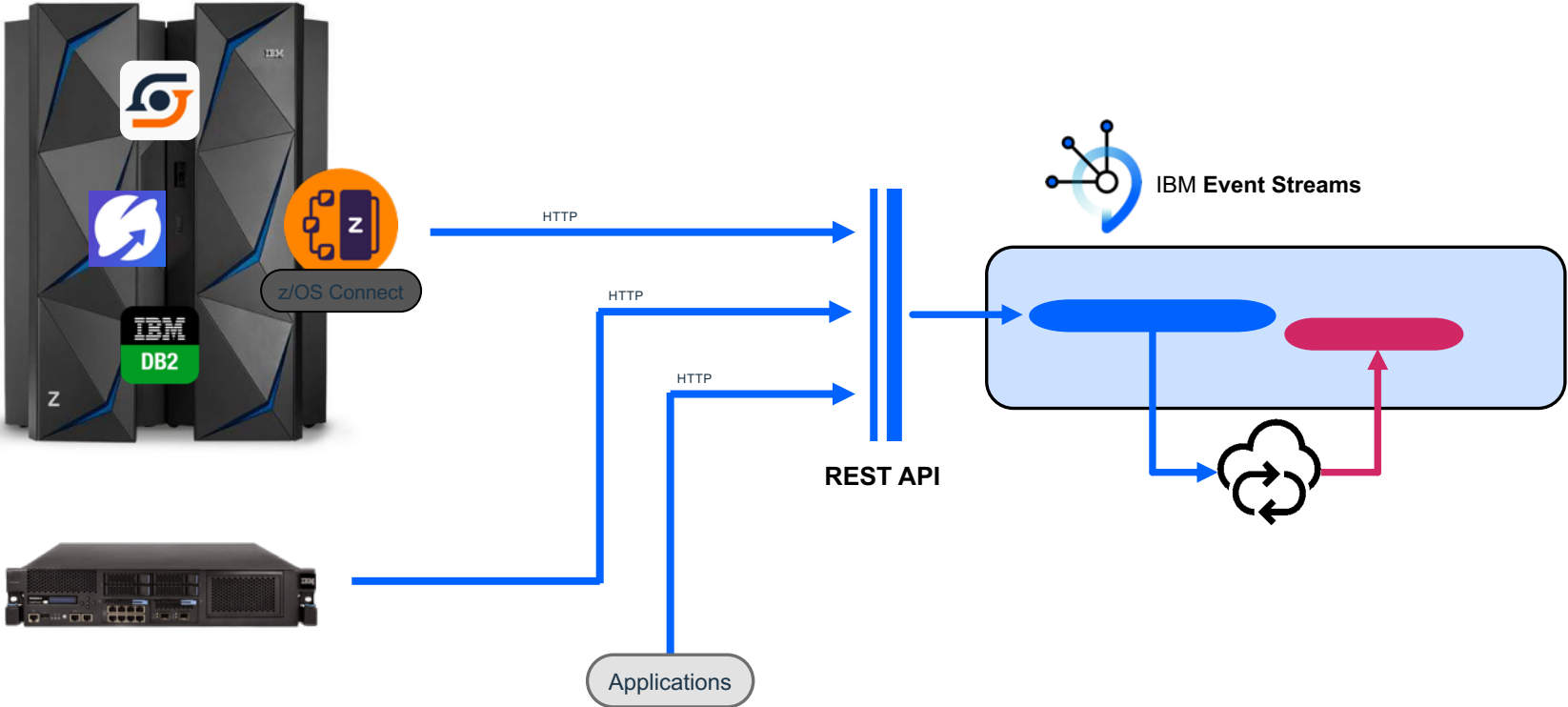
Find out more about Kafka Connect



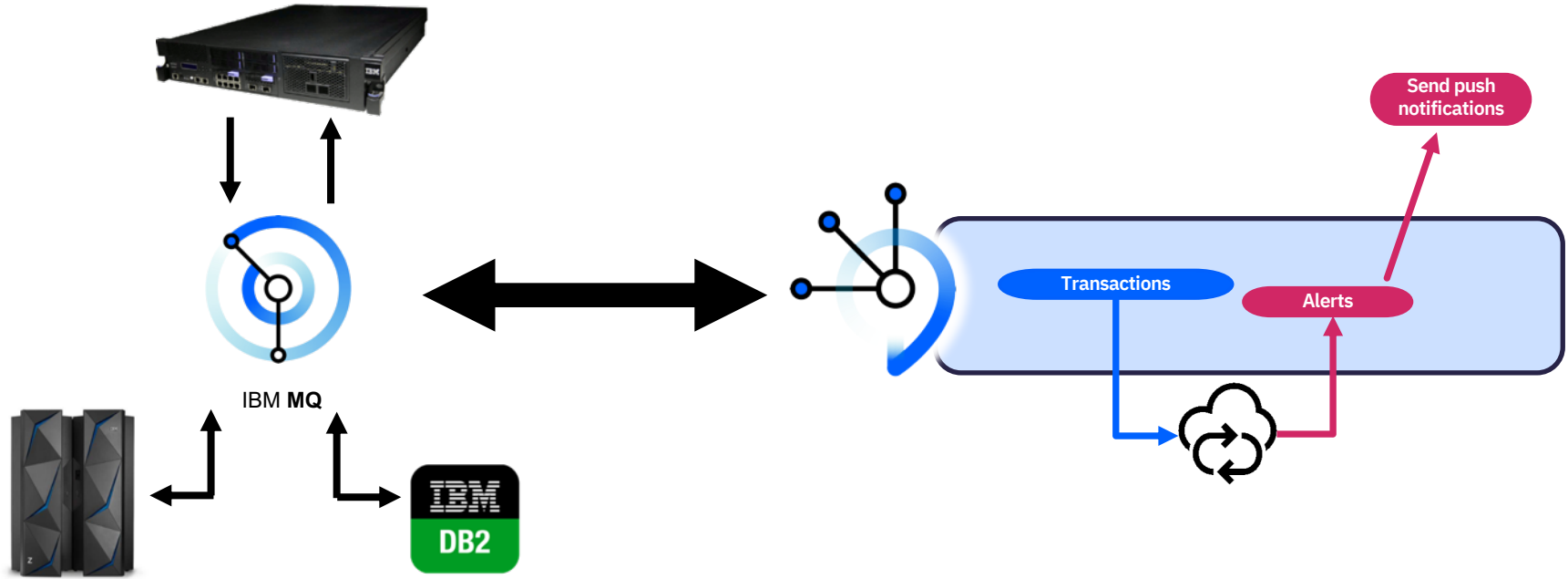
All (17)	Source (8)	Sink (9)	
 Source connector Kafka Connect IBM MQ	 Sink connector Kafka Connect IBM MQ		
 Sink connector Kafka Connect ArangoDB		 Sink connector Kafka Connect IBM Cloud Object Storage	
 Source connector Kafka Connect Couchbase		 Sink connector Kafka Connect Couchbase	
 Sink connector Kafka Connect HTTP		 Sink connector Kafka Connect Memcached	

Unlock Events from Systems where Kafka Connectivity is a Problem

REST API for Inbound Data



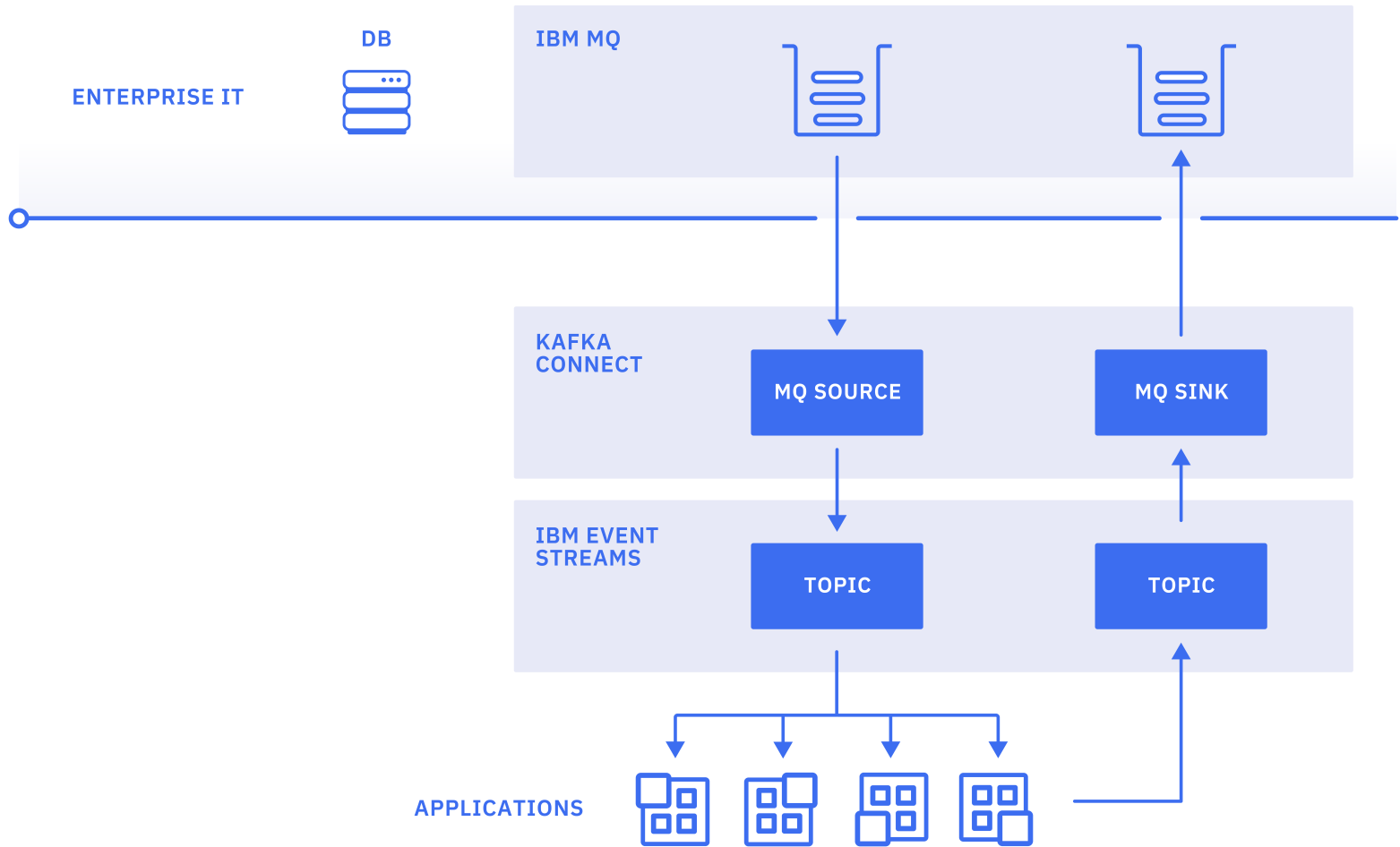
Integrates Seamlessly with IBM MQ



IBM MQ connects mission-critical Systems of Record, requiring **transactional, once-only delivery** E.g. payment transactions

IBM Event Streams distributes and processes streams of events in real-time to intelligently engage with customers E.g. alerts on spending patterns

IBM Event Streams with IBM MQ



It's easy to connect IBM MQ to Apache Kafka

IBM has created a pair of connectors, available as source code or as part of IBM Event Streams

Source connector

From MQ queue to Kafka topic

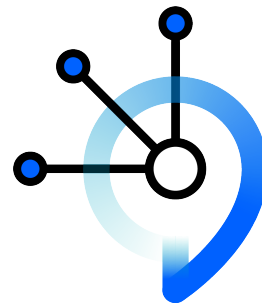
<https://github.com/ibm-messaging/kafka-connect-mq-source>

Sink connector

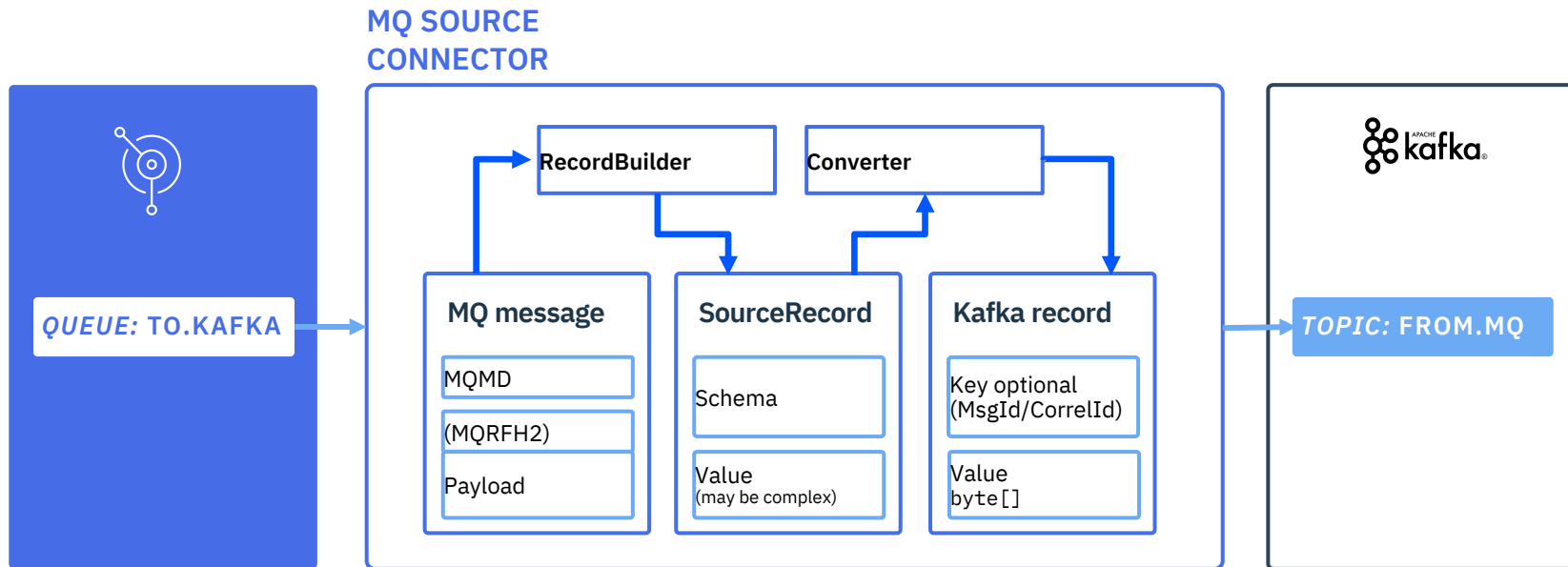
From Kafka topic to MQ queue

<https://github.com/ibm-messaging/kafka-connect-mq-sink>

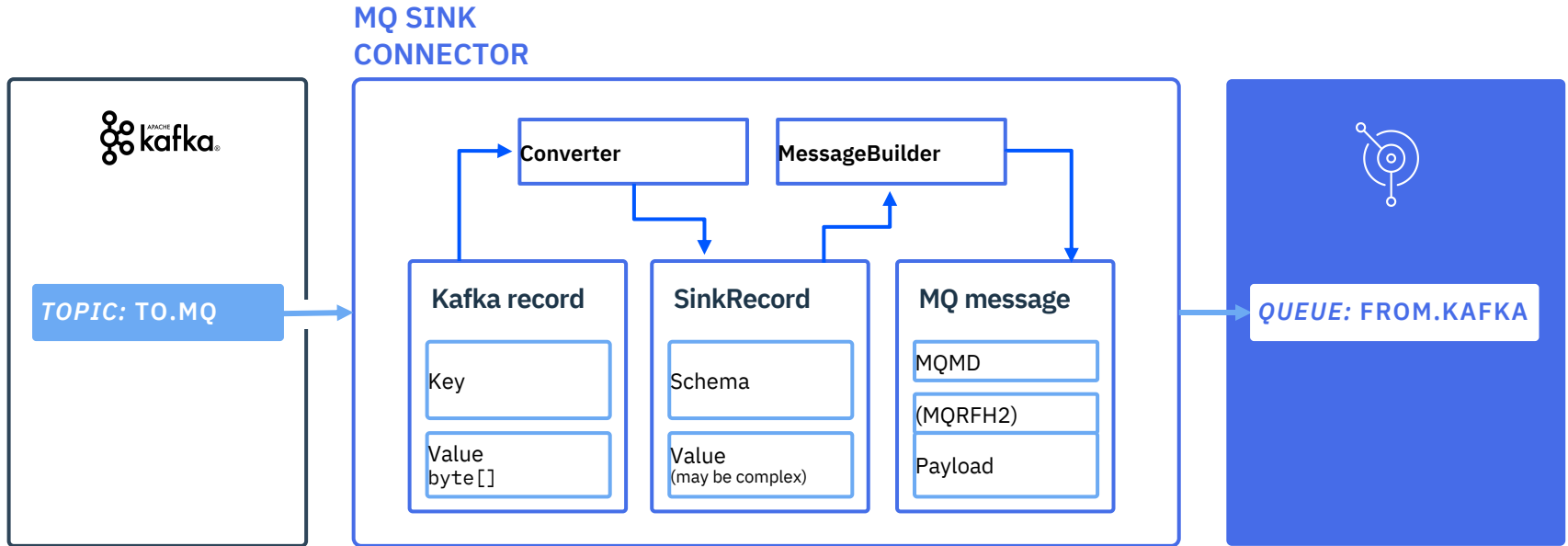
Fully supported by IBM for customers with support entitlement for IBM Event Streams



MQ source connector

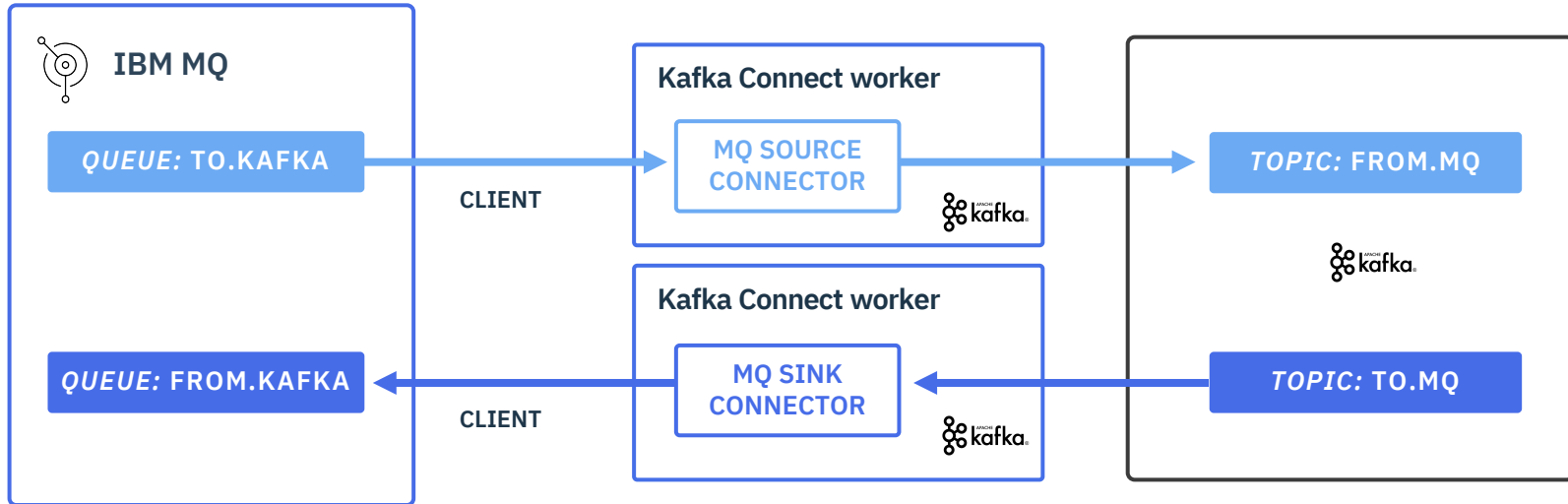


MQ sink connector



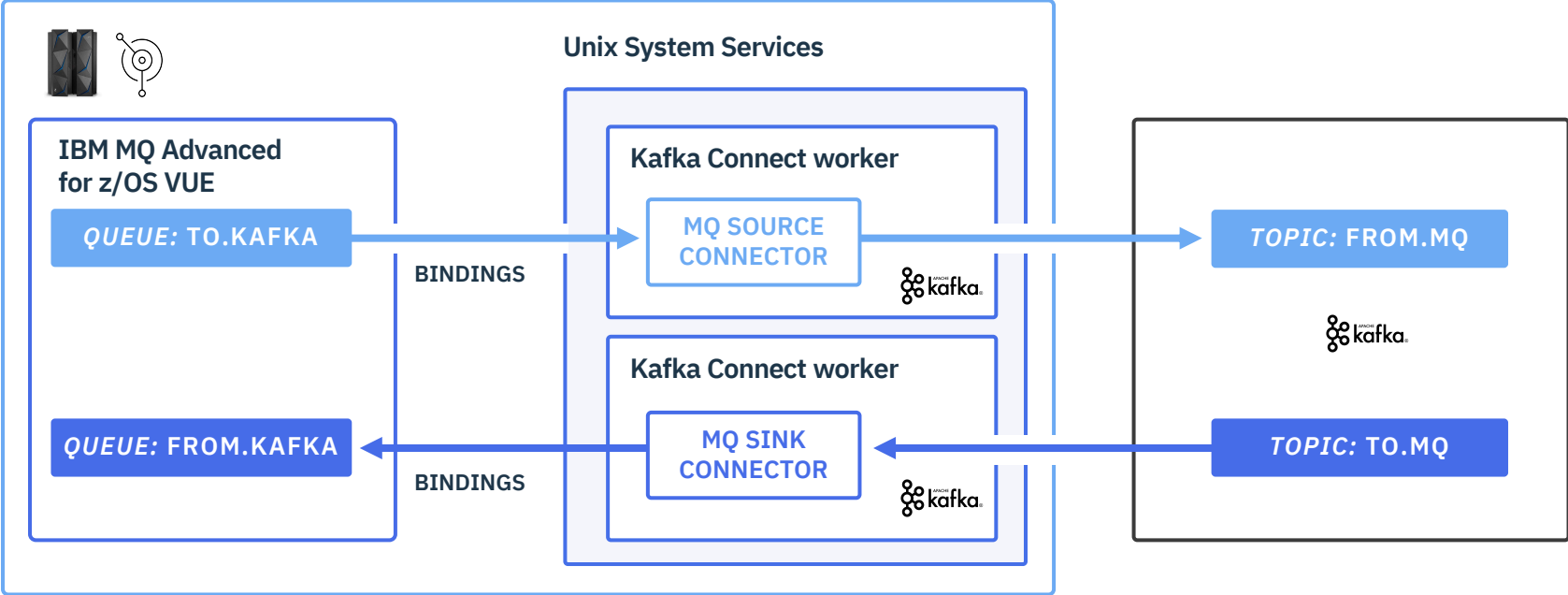
Connecting IBM MQ to Apache Kafka

The connectors are deployed into a Kafka Connect runtime



Running Kafka Connect on a mainframe

IBM MQ Advanced for z/OS VUE provides support for the Kafka Connect workers to be deployed onto z/OS Unix System Services using bindings connections to MQ



Configuration of the connectors

Configuration is provided in a properties file

Required:

mq.queue.manager – MQ QMgr name

mq.connection.name.list – MQ client conname

mq.channel.name – MQ svrconn channel name

mq.queue – MQ source queue

topic – Kafka target topic

Optional:

mq.user.name – MQ user name for client

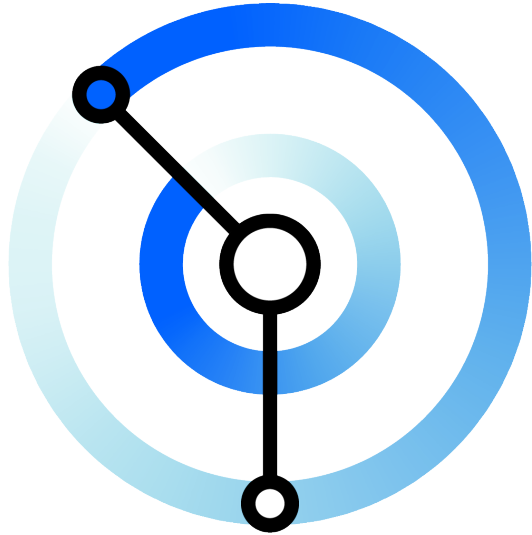
mq.password – MQ password for client

mq.message.body.jms – native MQ or JMS

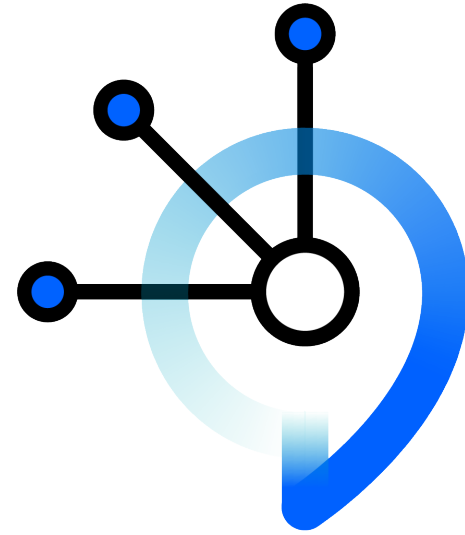
mq.ssl.cipher.suite – MQ SSL cipher suite

mq.ssl.peer.name – MQ SSL peer name

Sample file provided in GitHub

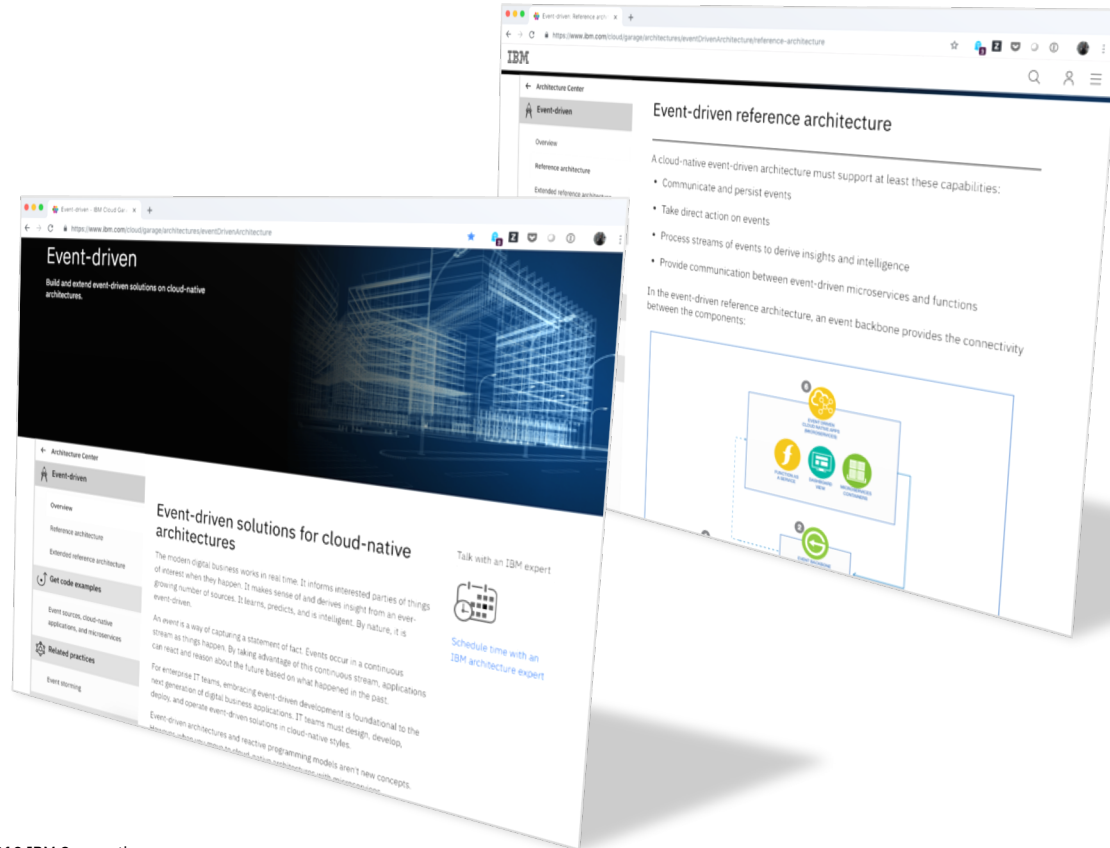


IBM MQ



IBM Event Streams

IBM Cloud Garage | Event-Driven Architecture Center



<http://ibm.biz/EventDriven>

Thank you!

<https://kafka.apache.org/>

<https://ibm.github.io/event-streams/>

<https://github.com/ibm-messaging/kafka-connect-mq-source>

<https://github.com/ibm-messaging/kafka-connect-mq-sink>

<https://ibm.github.io/event-streams/connecting/mq/zos/>

<https://www.linkedin.com/in/andrew-dunnings-a899a1123/>