CONFLUENT

ksqlDB Workshop

June 24th, 2020

Patrick Druley

Senior Solution Engineer @Confluent

Twitter @PatrickLovesAK

WORKSHOP LEADER

Patrick Druley

Sr. Solutions Engineer, Confluent

Patrick Druley is a Senior Solution Engineer at Confluent covering customers in the Ohio Valley territory. His career has focussed on databases, data warehousing and included various analytics and big data projects with companies across multiple industries while working at Teradata and Oracle. In addition to talking to folks about Apache Kafka and Stream Processing everyday at Confluent, he is also the lead Confluent Cloud subject matter expert for Solution Engineers. Patrick is an Ohio native, currently based out of Medina, OH.

JT Smith
Solutions Engineer
Confluent

Dan Parsons
Solutions Engineer
Confluent

Brian Likosar
Systems Engineer
Confluent

Chris Larsen
Solutions Engineer
Confluent

Today's Agenda

10:00 - 10:45 AM

Streams Processing/KSQL Overview
Patrick

10:45 AM - 12:15 PM

Interactive Streams Lab
Patrick with help from JT, Chris, Dan, Brian

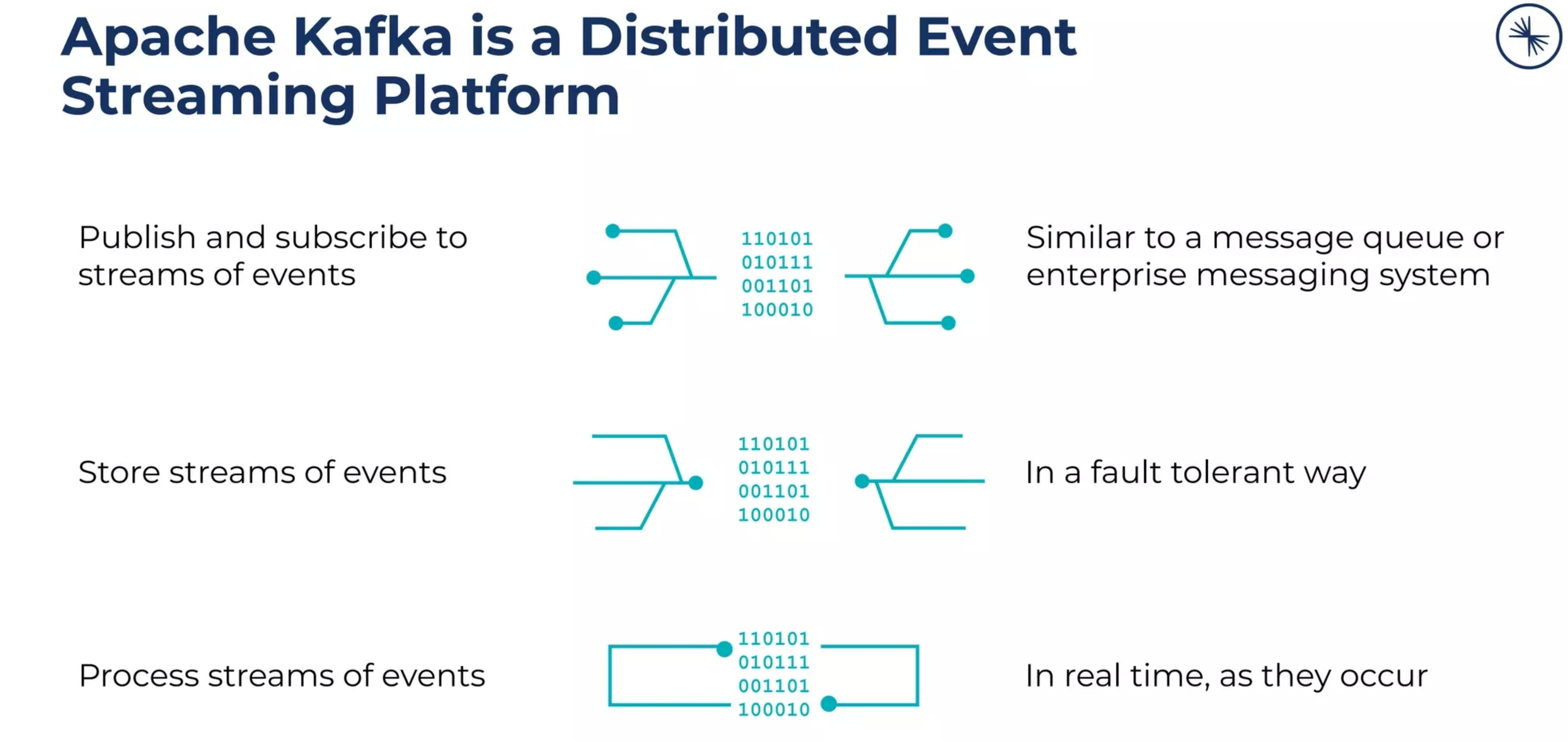
12:15 - 12:30 PM

Q&A and Next Steps
Open Discussion

Workshop Tips & Help:

1. Disconnect from VPN.
2. Check the **'Chat' window** during the session for instructions *[icon located at the bottom of the Zoom toolbar]*
3. For any technical issues, click the **'Raise Hand' button** or post in the **'Chat' window** *[a Confluent team member will assist you]*

Apache Kafka is a Distributed Event Streaming Platform



Publish and subscribe to streams of events

Similar to a message queue or enterprise messaging system

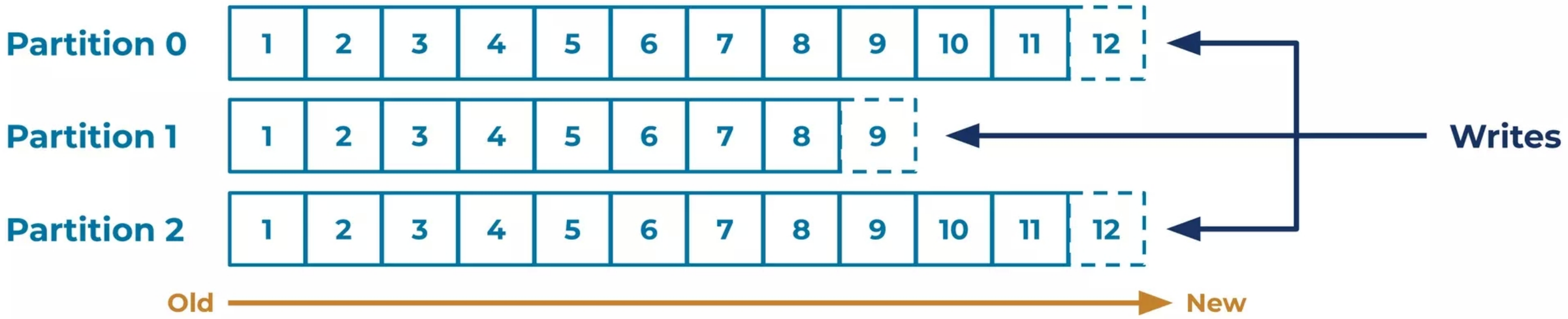
Store streams of events

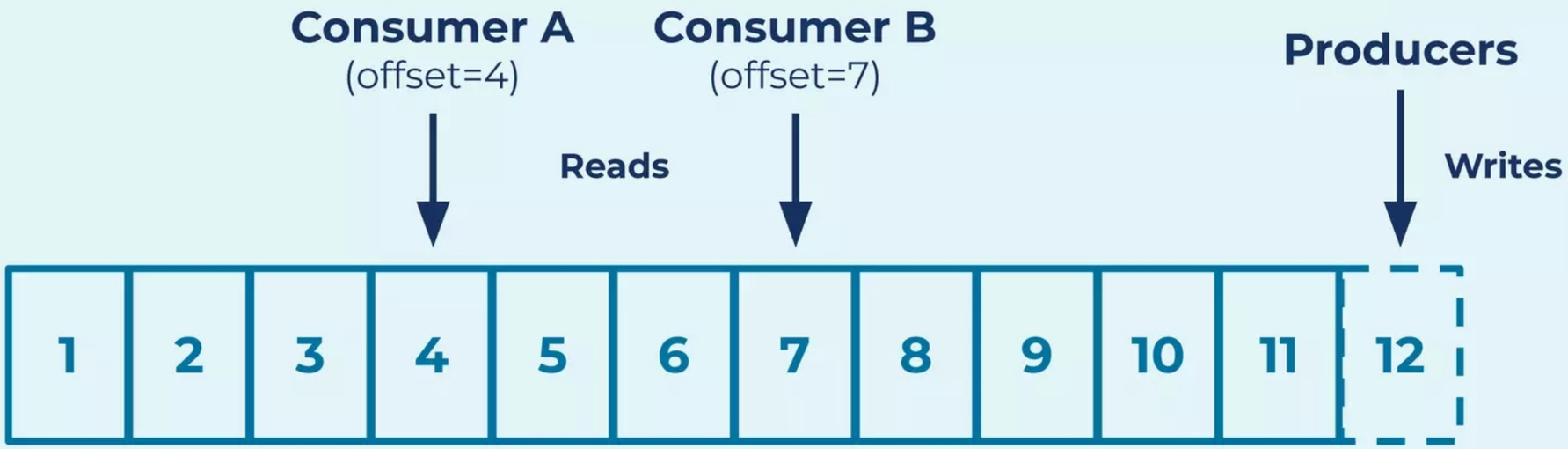
In a fault tolerant way

Process streams of events

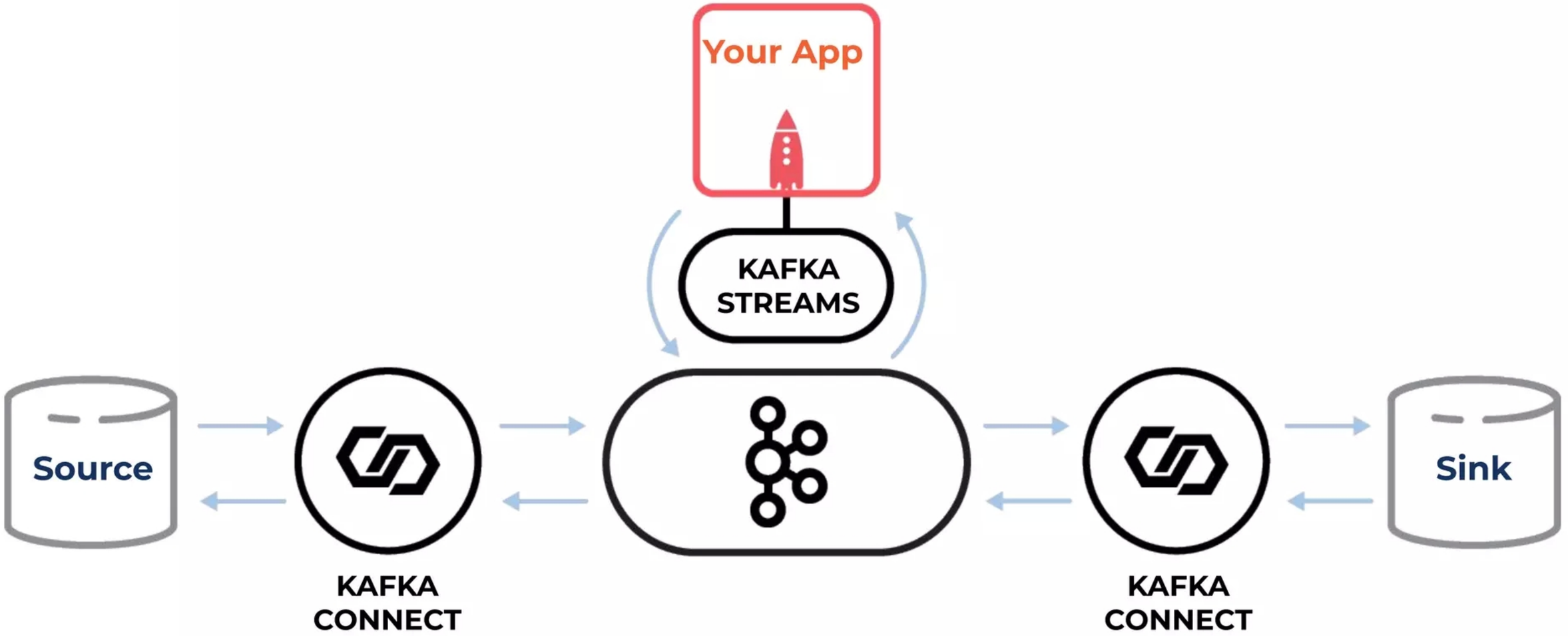
In real time, as they occur

Anatomy of a Kafka Topic

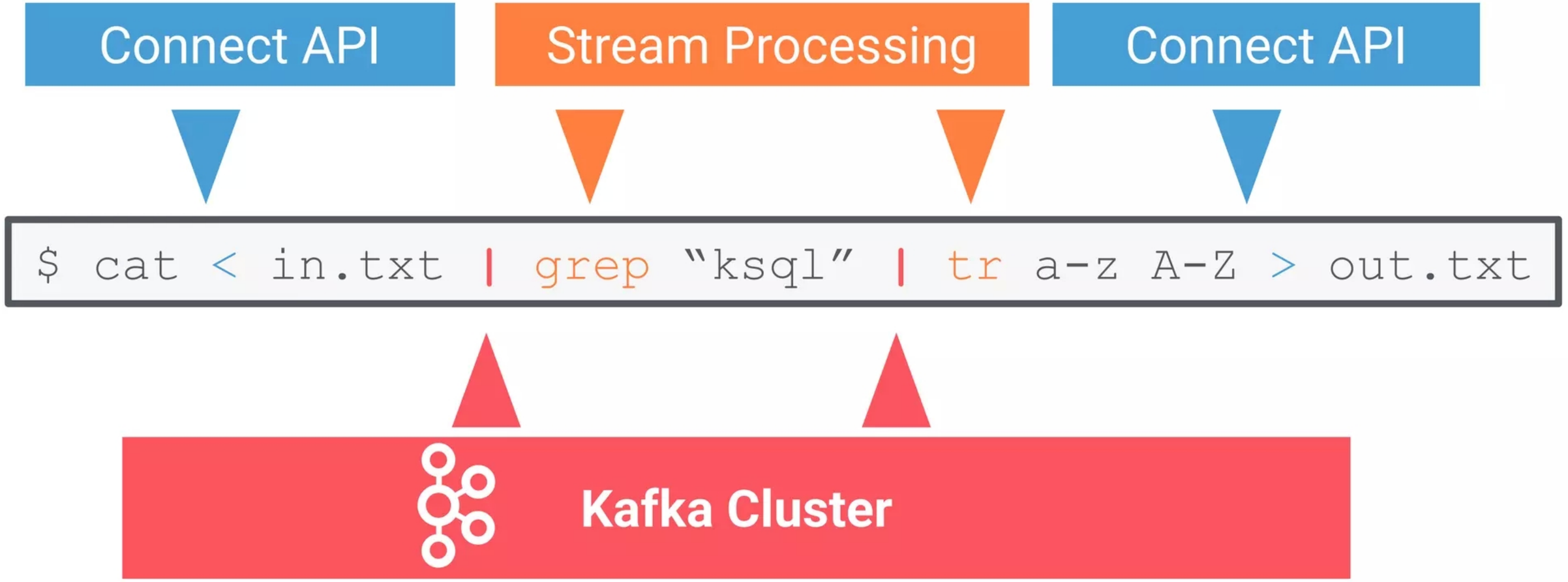




Kafka Connect and Kafka Streams



Stream Processing by Analogy



Client Trade-offs

KSQL

Select…from…
join…where…
group by..

Kafka Streams

mapValues(),
filter(),
aggregate()

Consumer, Producer

subscribe(),
poll(), send(),
flush()

Stream processing with Kafka

```
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 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

Stream processing with Kafka

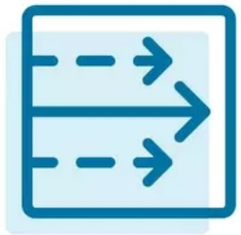
```
CREATE STREAM fraudulent_payments AS
  SELECT * FROM payments
  WHERE fraudProbability > 0.8;
```

Same example, now with **KSQL**.
Not a single line of Java or Scala code needed.

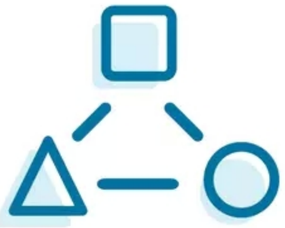
ksqlDB Example Use Cases

Data exploration

Data enrichment



Streaming ETL



Filter, cleanse, mask

Real-time monitoring

Anomaly detection

ksqlDB for Real-Time Monitoring

- Log data monitoring
- Tracking and alerting
- Syslog data
- Sensor / IoT data
- Application metrics

```
CREATE STREAM syslog_invalid_users AS
  SELECT host, message
  FROM syslog
  WHERE message LIKE '%Invalid user%';
```

http://cnfl.io/syslogs-filtering / http://cnfl.io/syslog-alerting

ksqlDB for Anomaly Detection

- Identify patterns or anomalies in real-time data, surfaced in milliseconds

```
CREATE TABLE possible_fraud AS
  SELECT card_number, COUNT(*)
  FROM authorization_attempts
  WINDOW TUMBLING (SIZE 5 SECONDS)
  GROUP BY card_number
  HAVING COUNT(*) > 3;
```

ksqlDB for Streaming ETL

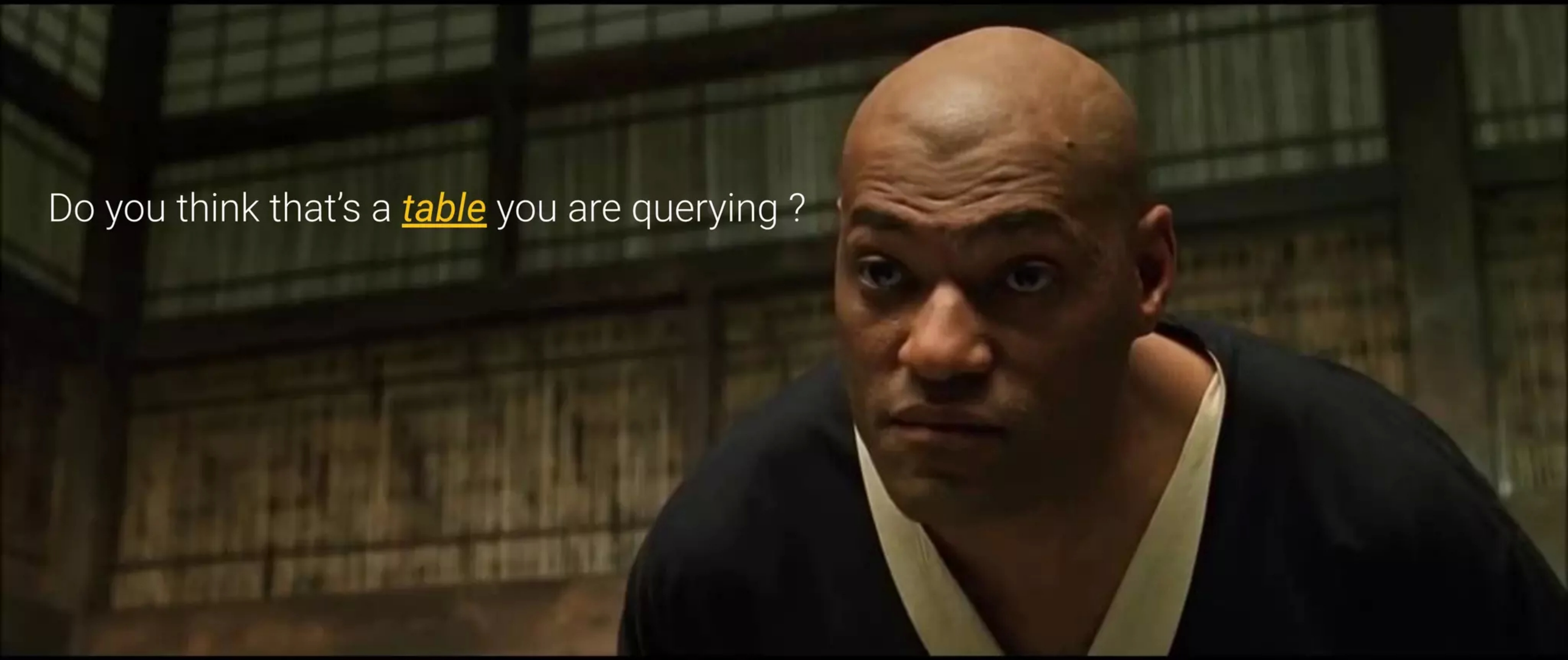
- Joining, filtering, and aggregating streams of event data

```
CREATE STREAM vip_actions AS
  SELECT user_id, page, action
  FROM clickstream c
  LEFT JOIN users u
    ON c.user_id = u.user_id
  WHERE u.level = 'Platinum';
```

ksqlDB for Data Transformation

- Easily make derivations of existing topics

```
CREATE STREAM pageviews_avro
  WITH (PARTITIONS=6,
        VALUE_FORMAT='AVRO') AS
  SELECT * FROM pageviews_json
  PARTITION BY user_id;
```


Do you think that’s a *table* you are querying ?

Where is KSQL not such a great fit?

Post-fact Ad-hoc queries

- Limited span of time usually retained in Kafka
- No indexes for random lookups

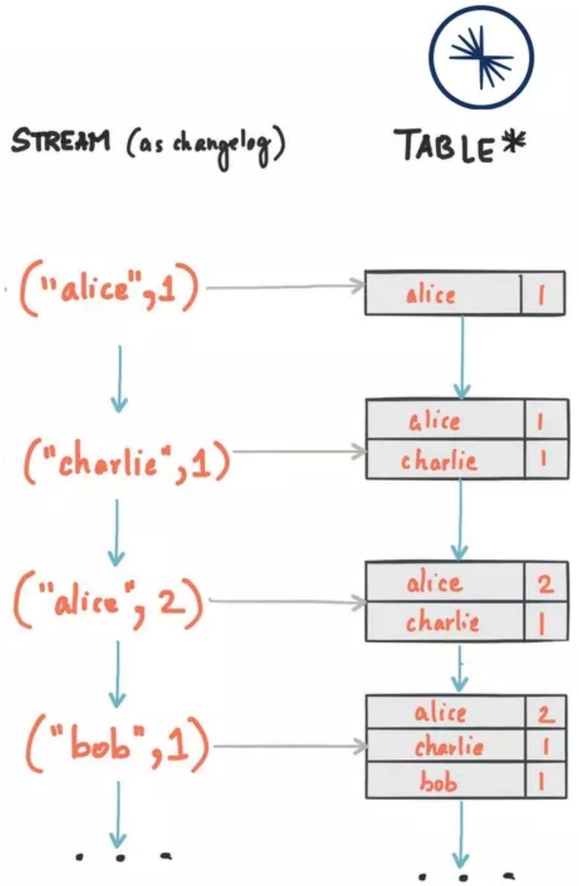
BI reports (Tableau etc.)

- No secondary indexes
- No JDBC (most BI tools are not good with continuous results!)

Stream/Table Duality

Streams & Tables

- STREAM and TABLE as first-class citizens
- Interpretations of topic content
- STREAM - data in motion
- TABLE - collected state of a stream
  - One record per key (per window)
  - Current values (compacted topic)
  - Changelog
- STREAM – TABLE Joins



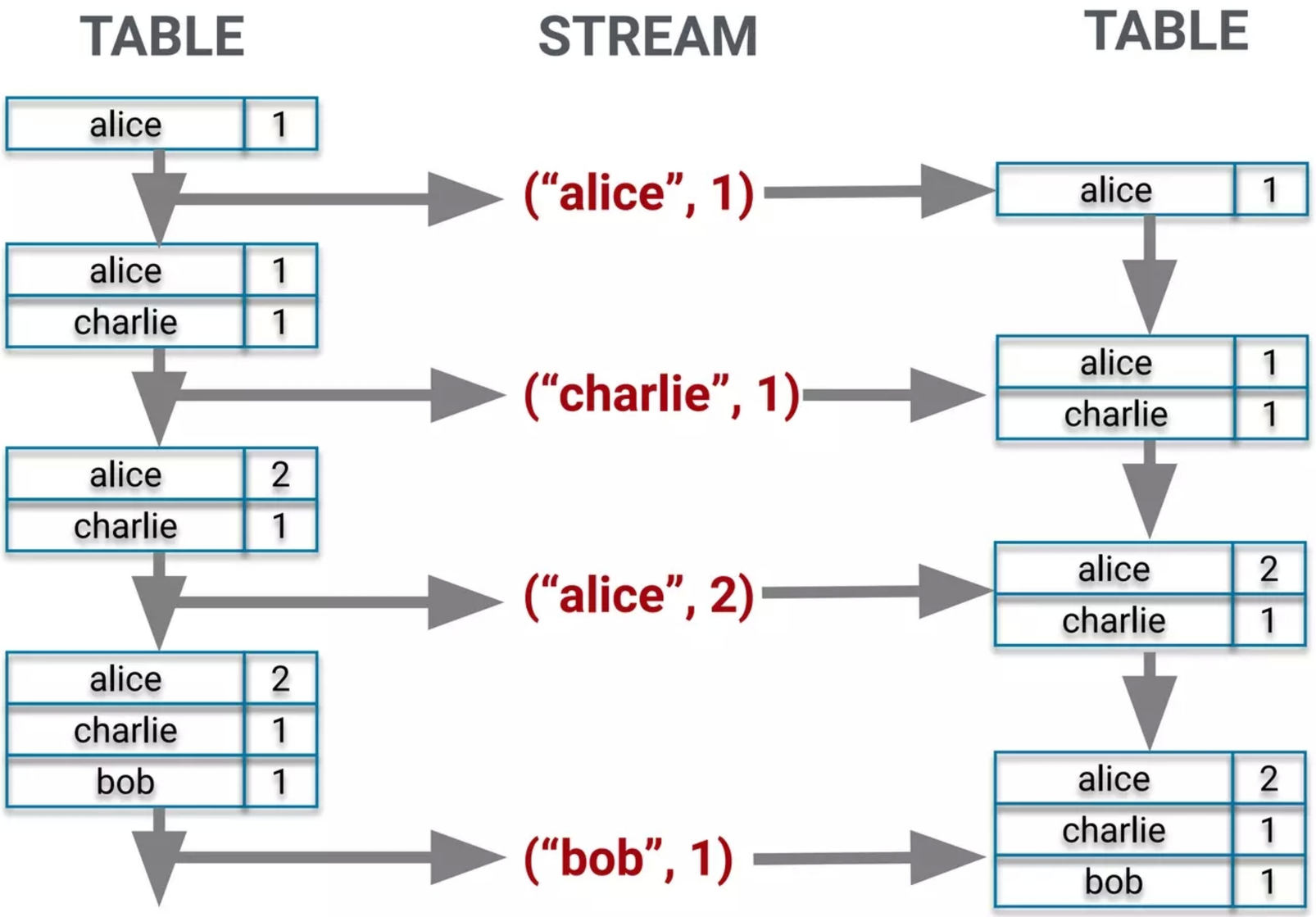
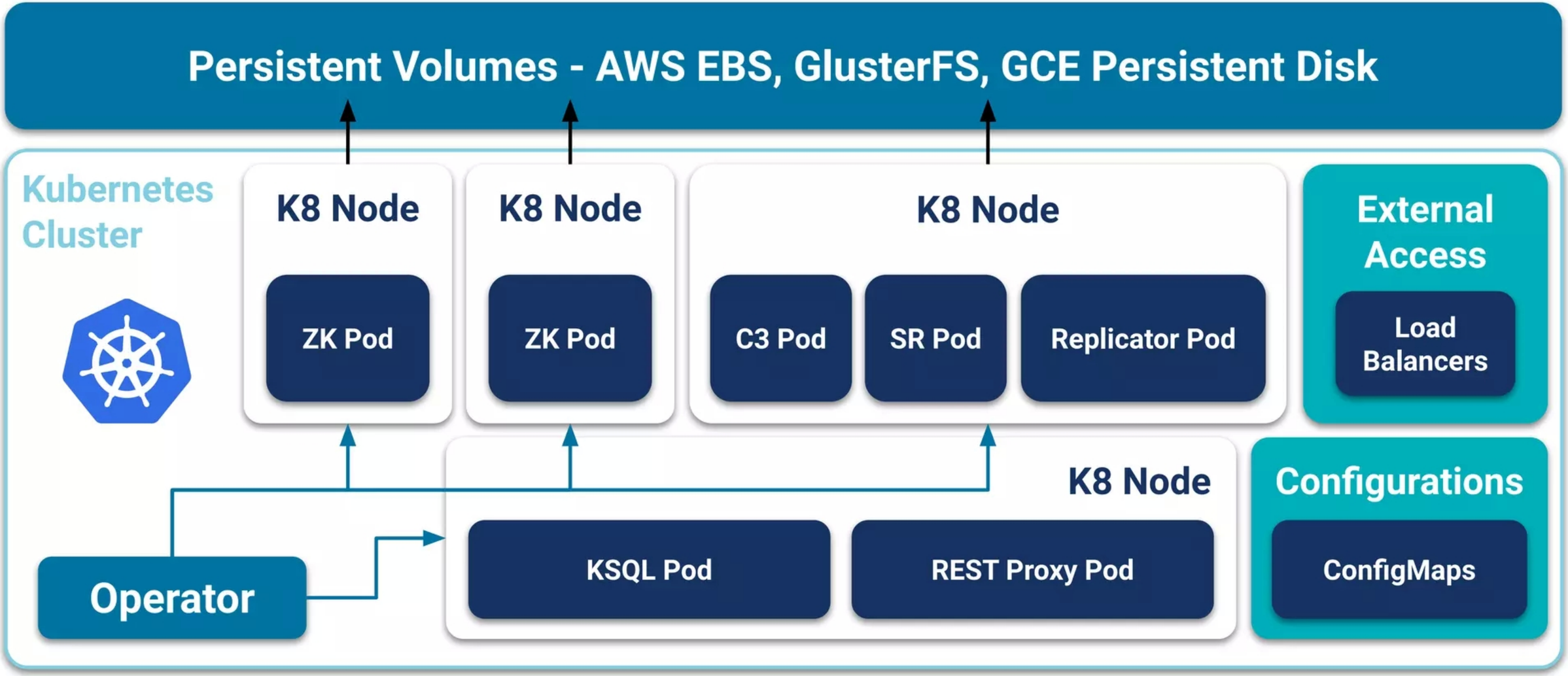


TABLE
STREAM
TABLE
alice 1
alice 1
charlie 1
alice 2
charlie 1
alice 2
charlie 1
bob 1
("alice", 1)
("charlie", 1)
("alice", 2)
("bob", 1)
alice 1
alice 1
charlie 1
alice 2
charlie 1
alice 2
charlie 1
bob 1


HIVE
INBOARD VALVES
RIG FOR DAVE

Confluent Operator Architecture and Deployment



user-id =
first 3 letters of first name + first 3 letters of last name

example: Patrick Druley = patdru

It's up to 3 letters, so if either name is less just use those letters.

Go to
http://<user-id>.us-southeast.gcp.confluent-demo.io

confluent


Learn Kafka.
Start building with Apache Kafka at Confluent Developer.
Confluent Developer
developer.confluent.io

CONFLUENT

Project Metamorphosis

Unveiling the next-gen event streaming platform

For Updates Visit
cnfl.io/pm

Jay Kreps
Co-founder and CEO
Confluent

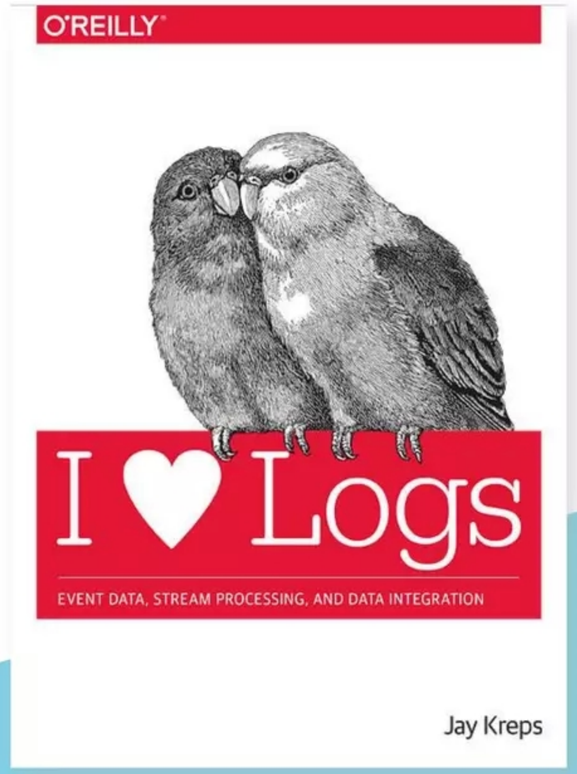
Kafka Summit 2020: Event Streaming Everywhere will be hosted virtually.

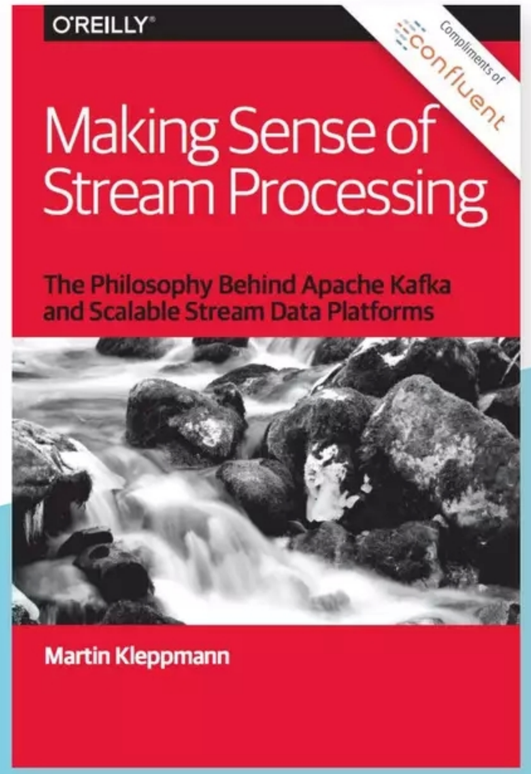
August 24-25, 2020

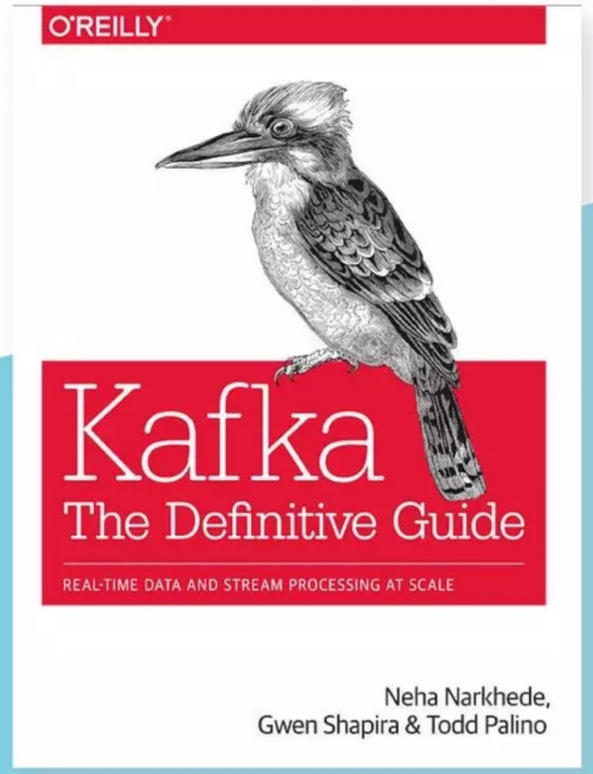
Register Today

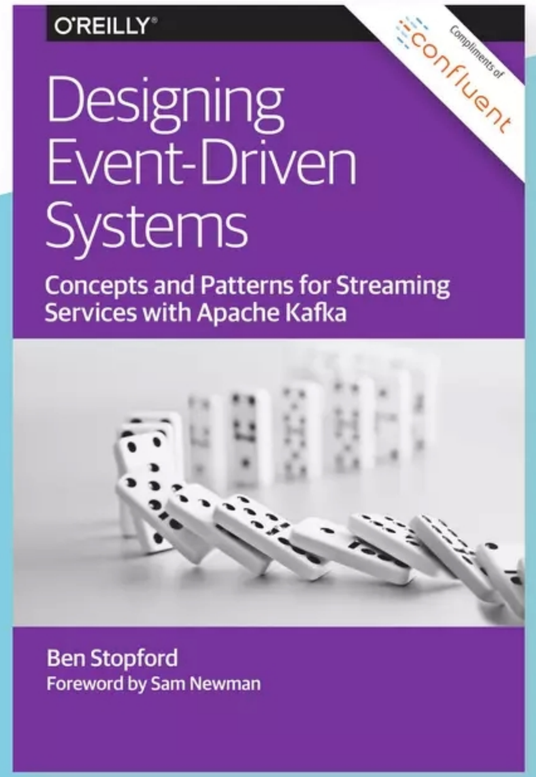
https://events.kafka-summit.org/2020

Download your Apache Kafka and Stream Processing O'Reilly Book Bundle









Download at: https://cnfl.io/FM4OReillyBooks

Stay in touch!

Confluent Blog
cnfl.io/blog

Streaming Audio
cnfl.io/podcast

Try Confluent
cnfl.io/download

Thank you!

 cnfl.io/meetups

 cnfl.io/blog

 cnfl.io/slack