香港01

Data Pipeline with Kafka

Dr. Mole T.Y. WONG @ HK OSCON 2018

2018 / 06 / 16 - 17

whoami

Why

深入了解用戶行為，洞悉可行的改善方法

Understand our users.

Provide actionable insights.

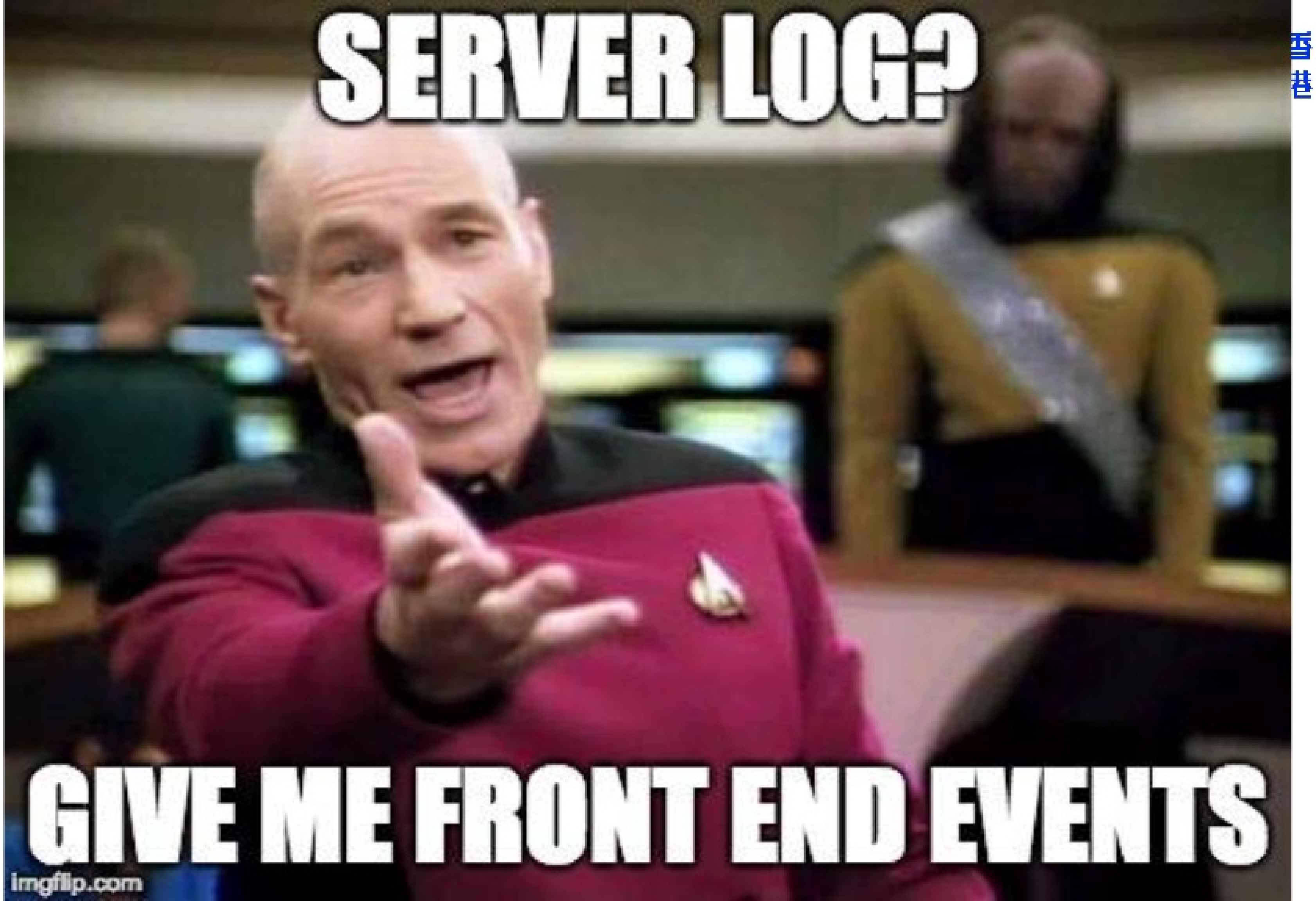
How

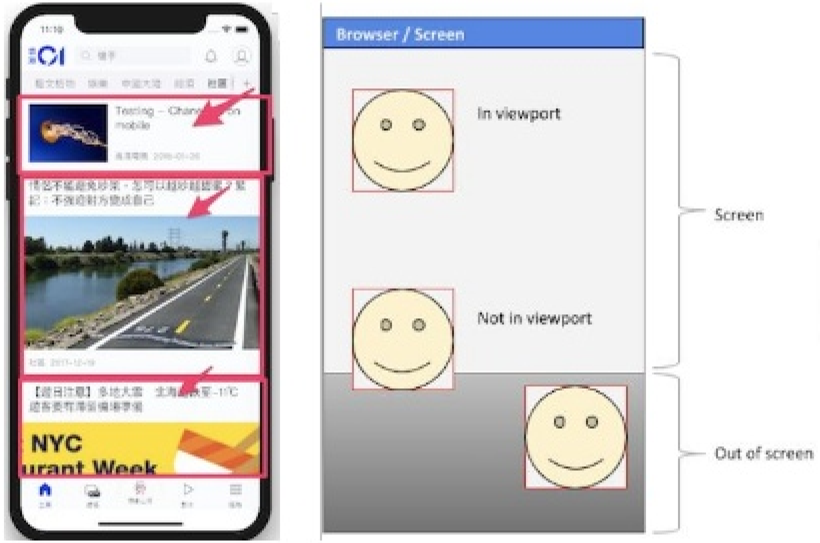
以數據驅動產品方向
Data driven: steer our product direction.

What

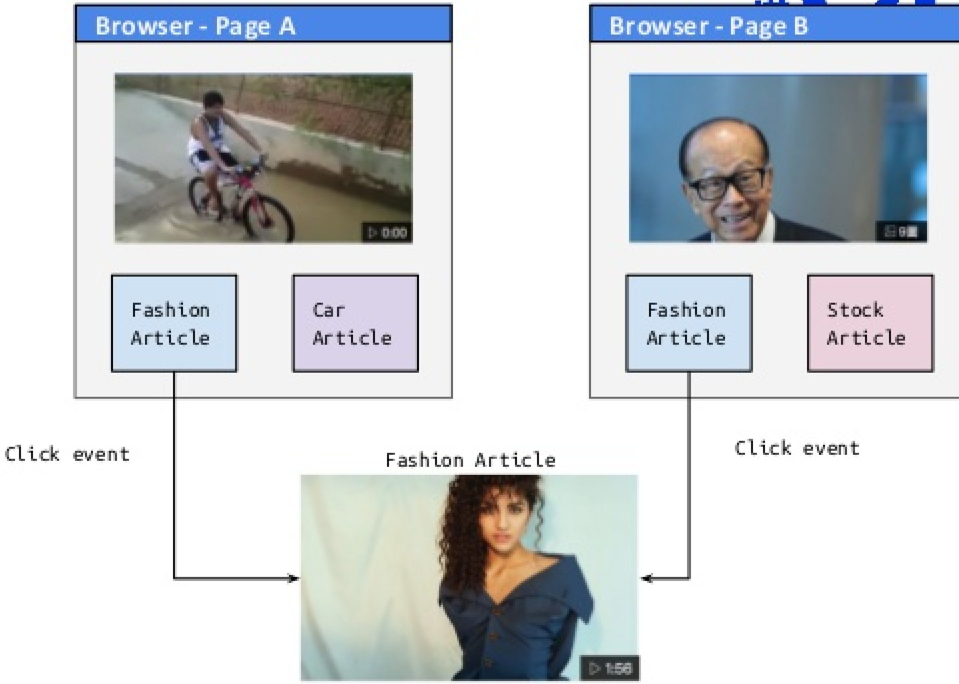
數據：定義、收集、處理、洞見

Data: definition, ingress, process, insight.


SERVER LOG?
GIVE ME FRONT END EVENTS
imgflip.com

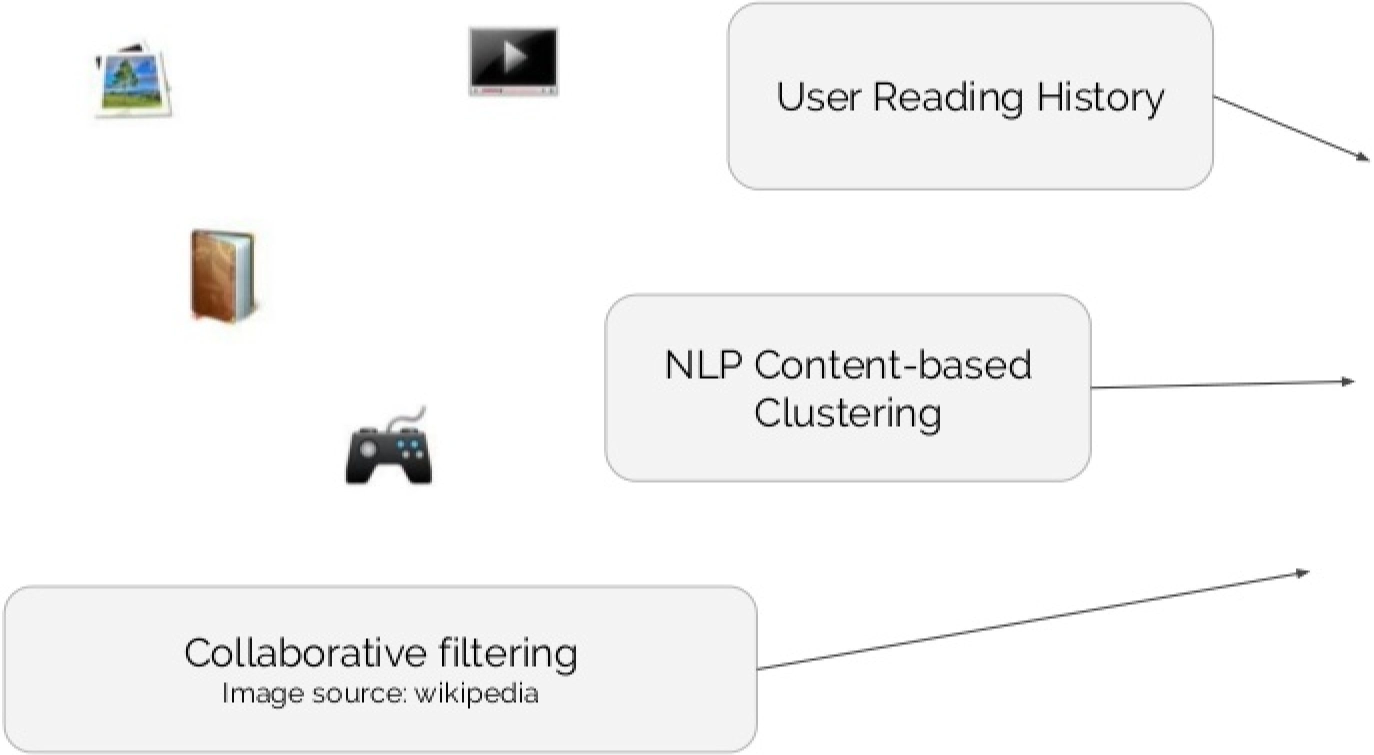


Click-Through Rate VS Pageview



Traffic Source Analysis

Data-Driven Product Development

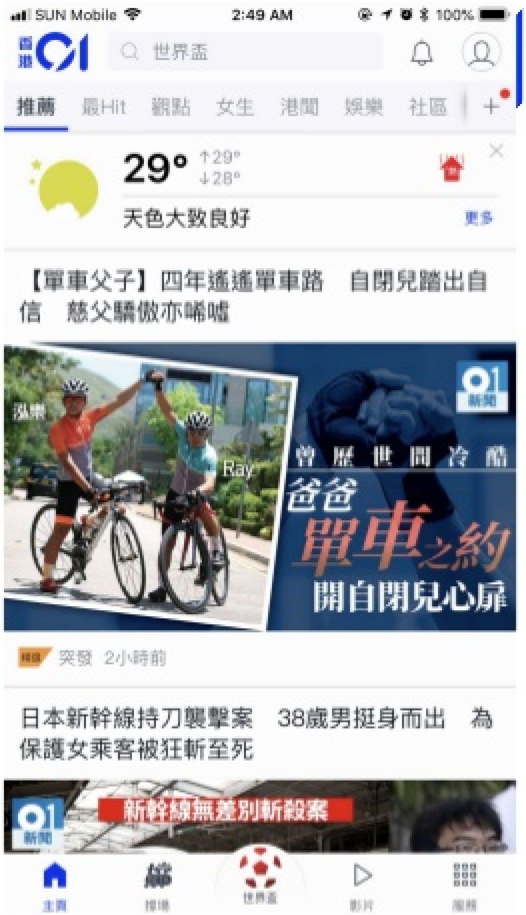


User Reading History

NLP Content-based Clustering

Collaborative filtering

Image source: wikipedia



SUN Mobile 2:49 AM 100%

世界盃

推薦 最Hit 觀點 女生 港聞 娛樂 社區

29° ↑29° ↓28°

天色大致良好

更多

【單車父子】四年遙遙單車路　自閉兒踏出自信　慈父驕傲亦唏噓

突發　2小時前

日本新幹線持刀襲擊案　38歲男挺身而出　為保護女乘客被狂斬至死

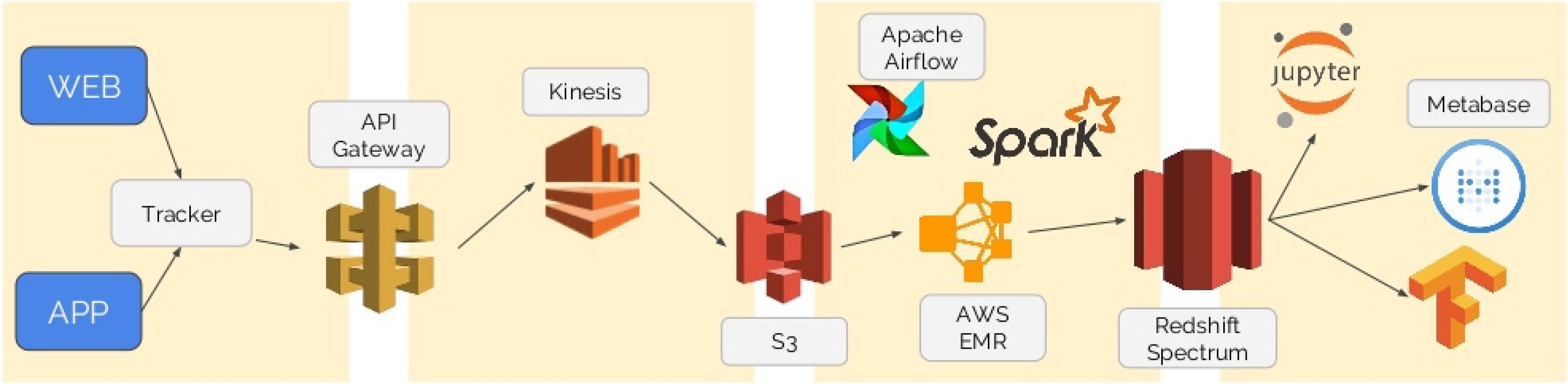
Machine Learning Products

Personalized Recommendation Feed

Outline

- Data pipeline - what is it?
- Kafka - roles in a data pipeline
- Other use cases of Kafka

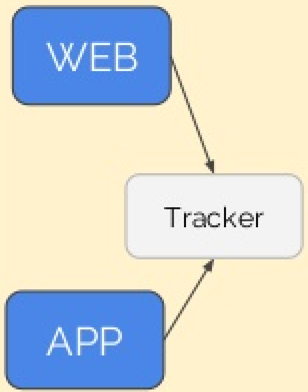
Typical Data Pipeline Setup



Data Ingress

JS Library (WEB)
Native Library (APP)

Google Analytics
Mixpanel
Matomo (Piwik)



Data Tracker

- **Nature**
  - Lightweight
  - Programmable
- **Capability**
  - Page view / Screen view
  - Custom events
  - Device identification
  - Session management

Different Aspects of a Data Pipeline

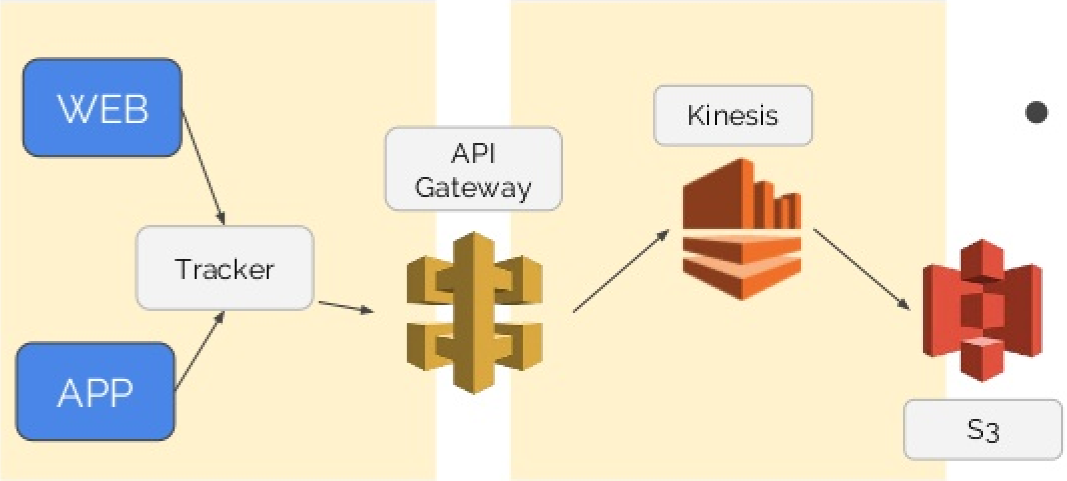
Data Ingress

JS Library (WEB)
Native Library (APP)

Google Analytics
Mixpanel
Matomo (Piwik)

Infrastructure

- AWS Kinesis
- Google Pub/Sub
- Apache Kafka



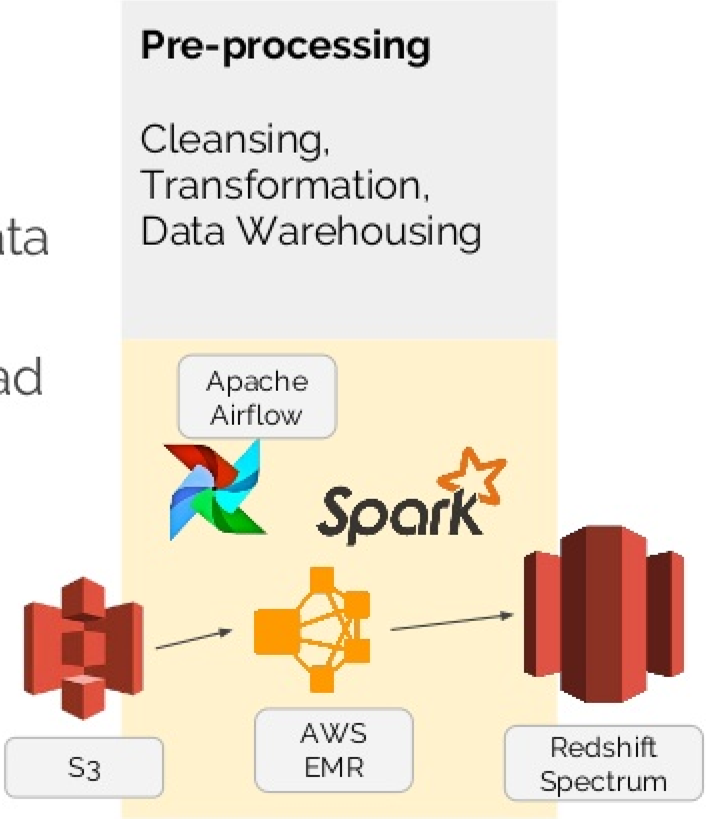
Different Aspects of a Data Pipeline

Data Infrastructure

- **Main Roles**
  - Buffering
  - Routing
  - Writing
- **Characteristics**
  - Multiple producers
  - Multiple consumers
  - Batch / Real-time

Pre-processing

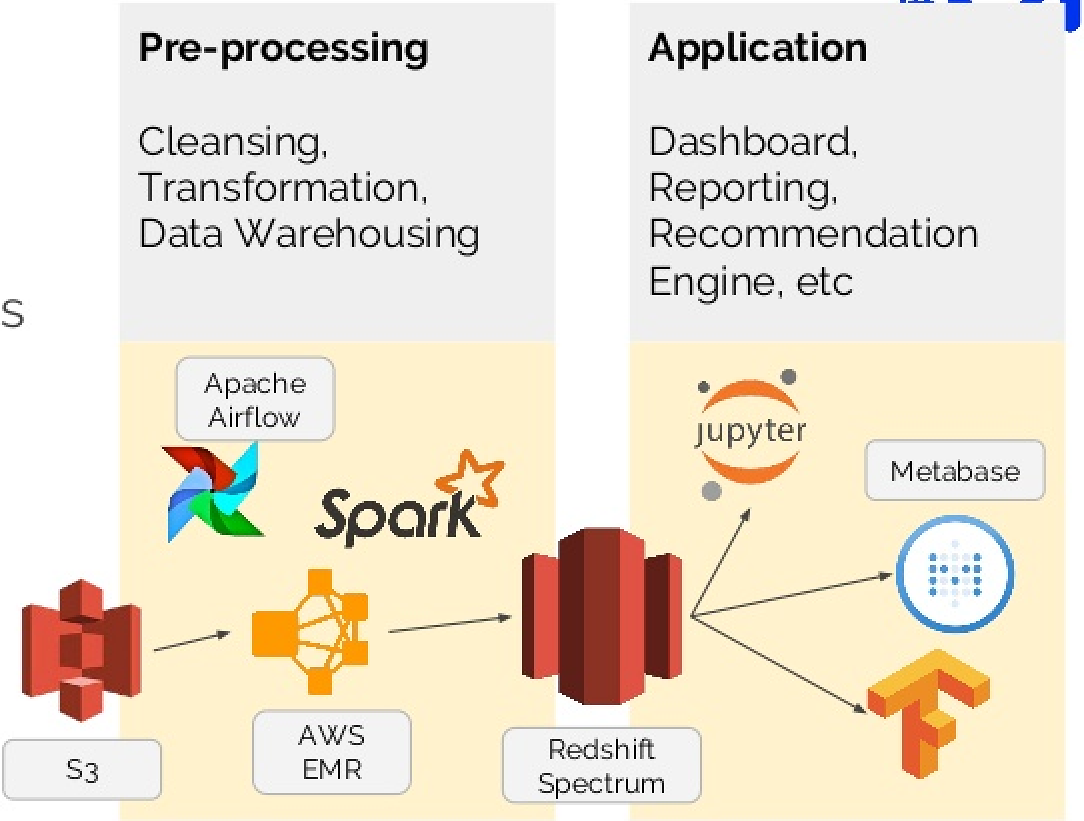
- **Main Roles**
  - Avoid direct querying raw data
  - Cleansing
  - ETL - Extract, Transform, Load
  - Scheduling
- **Characteristics**
  - Defining data sets
  - Time-frame-based queries



Different Aspects of a Data Pipeline

Application

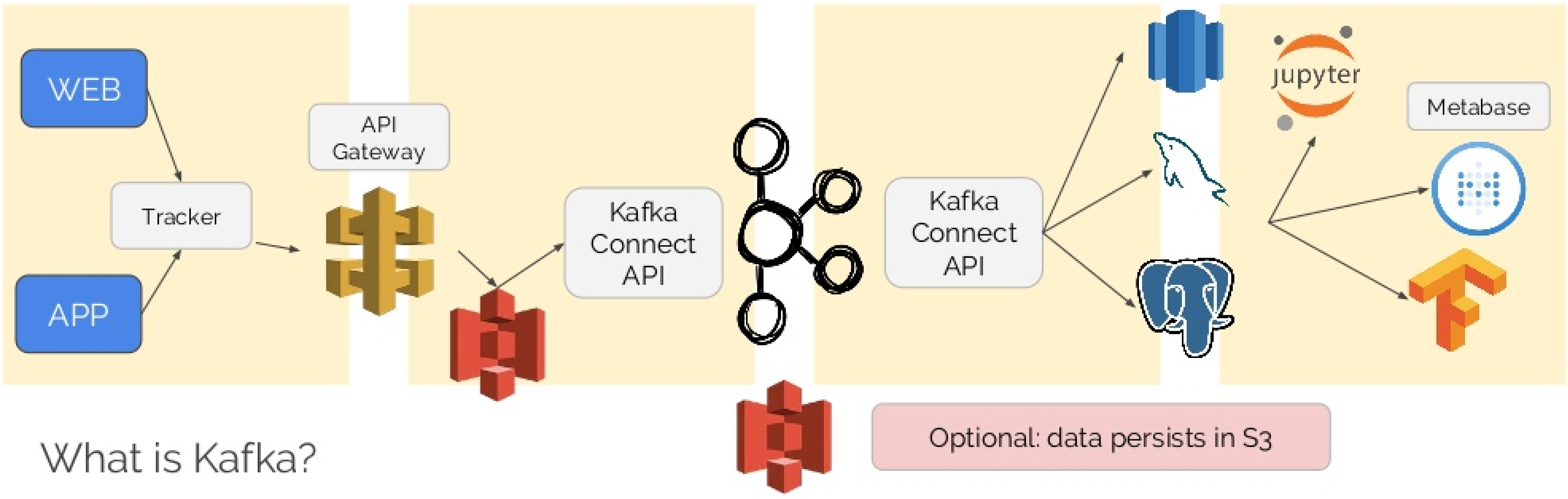
- **Main Roles**
  - KPI VS Exploration
  - Operators VS Data Scientists
  - Planned VS Ad-hoc queries
- **Characteristics**
  - Production-grade data
  - Fast is a must



Different Aspects of a Data Pipeline

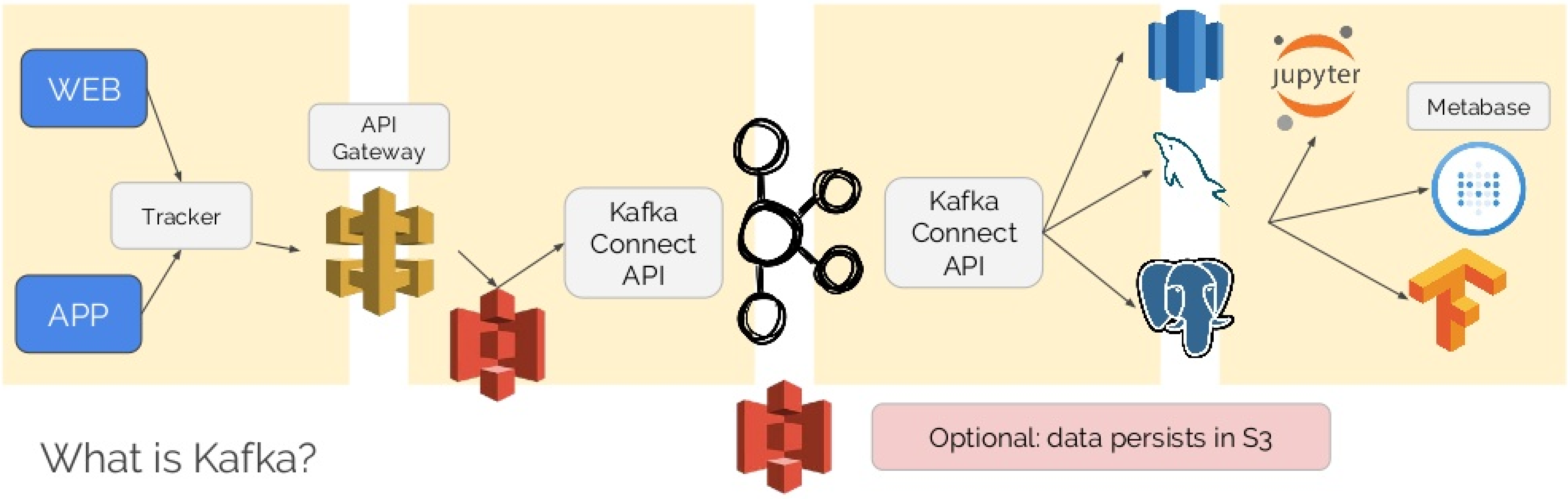
What is Kafka? https://kafka.apache.org/ Main Contributor: Gene NG

Data Pipeline with Kafka



What is Kafka?

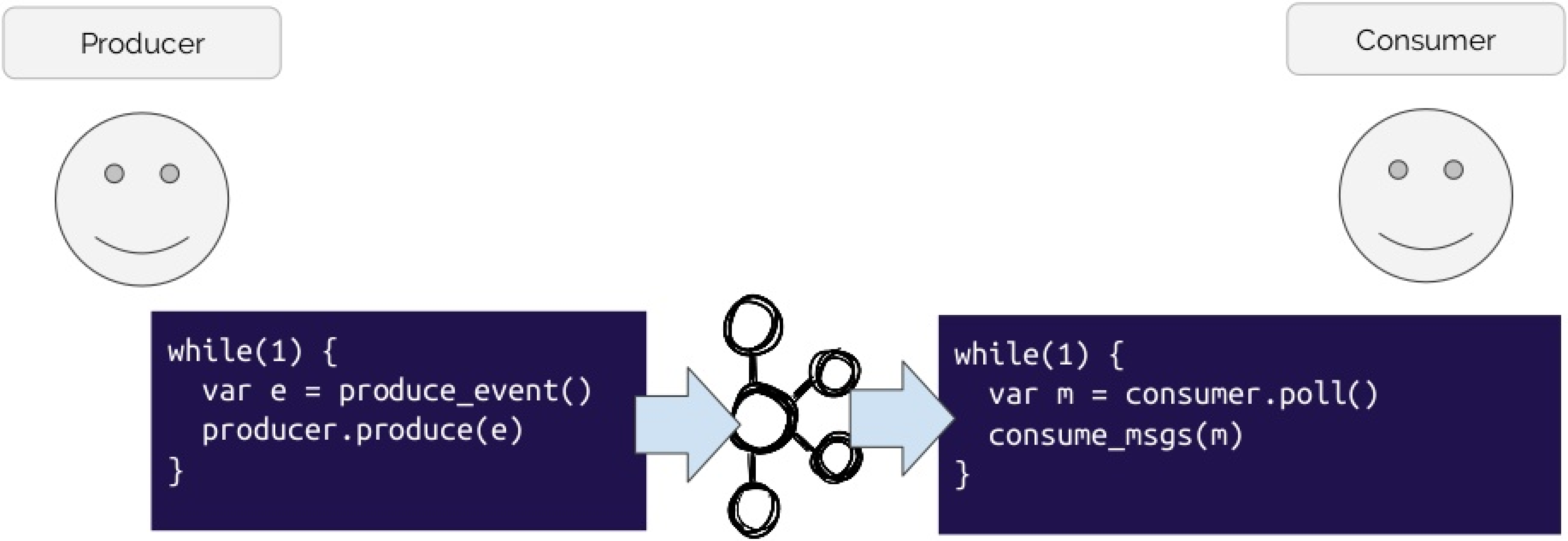
Data Pipeline with Kafka



Optional: data persists in S3

What is Kafka?

Basics: Producer-Consumer Model



Producer

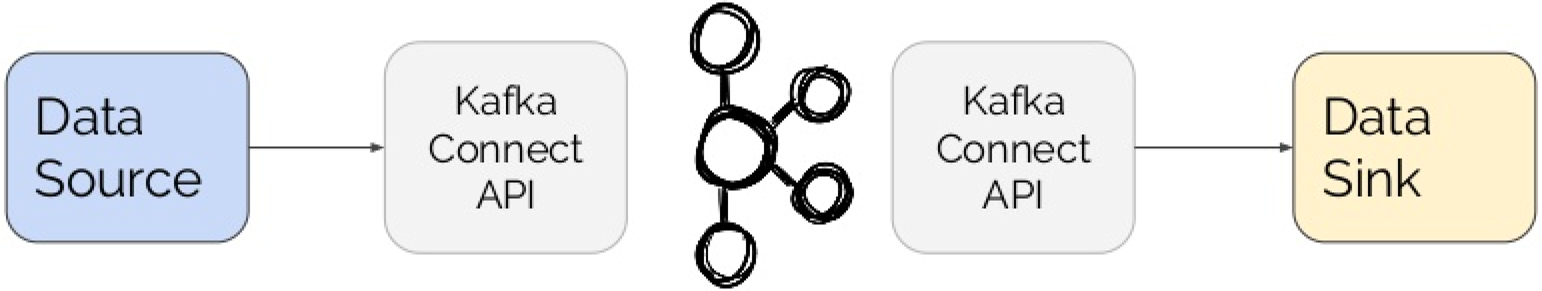
Consumer

```
while(1) {
  var e = produce_event()
  producer.produce(e)
}
```

```
while(1) {
  var m = consumer.poll()
  consume_msgs(m)
}
```

Connect API

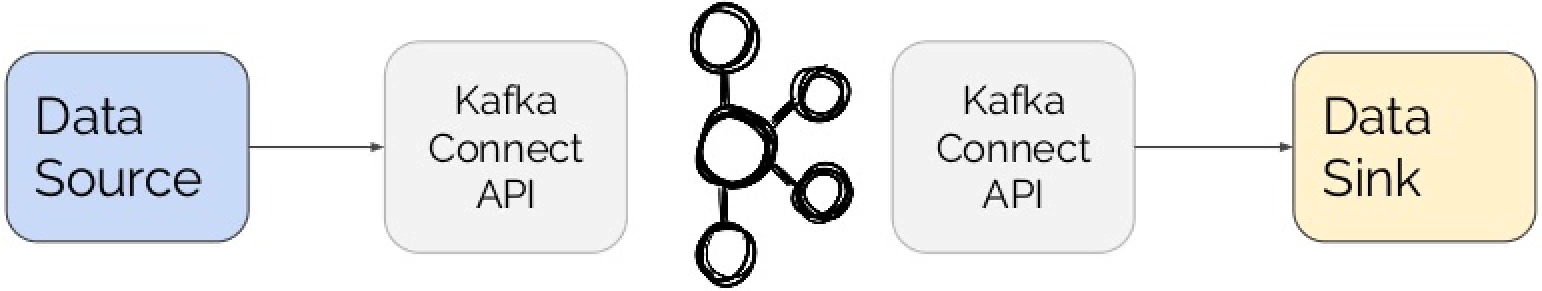
- For database / data source
- Wrapped consumer & producer code
- Nice thing: config file only!

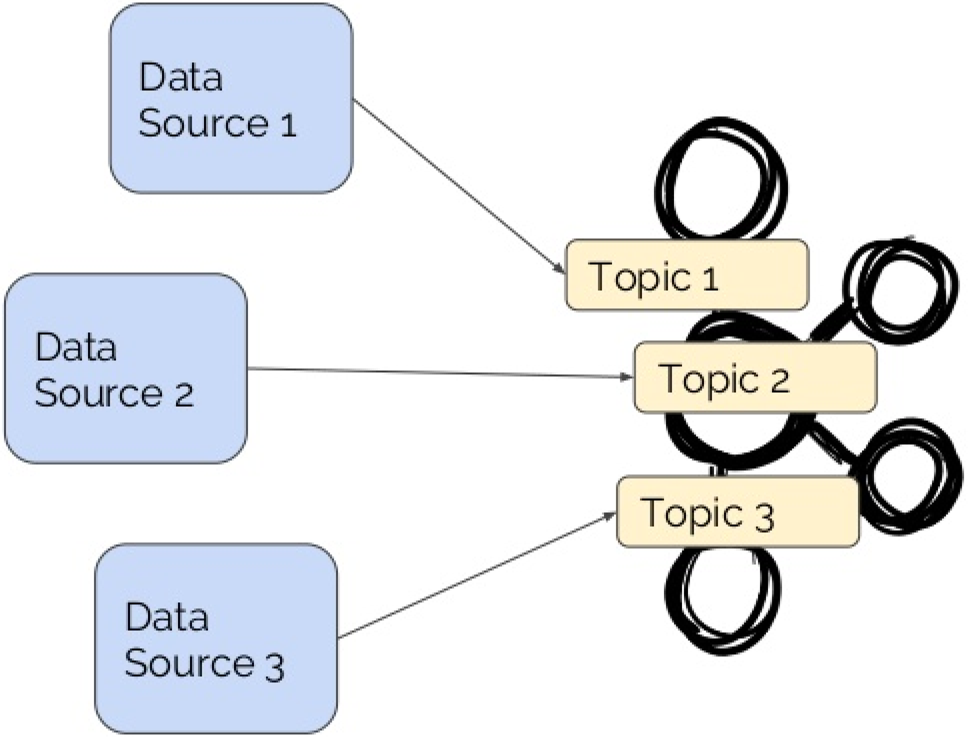


What is Kafka - terminology

Connect API - common connectors

JDBC - MySQL, PgSQL	S3
HDFS	ElasticSearch





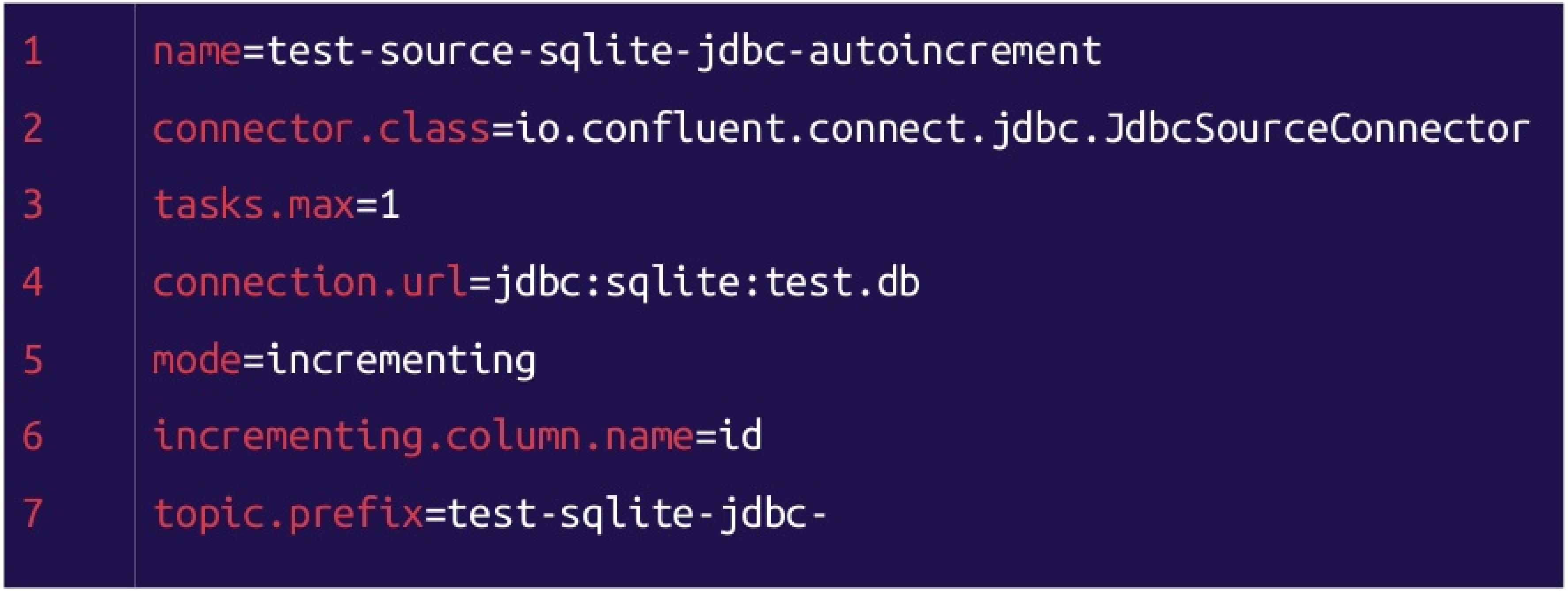
Data Topic Model

- One-to-one (most common)

Feature

- Autonomous
  - Loads data from sources whenever changes occur
- Storage
  - Writes data to the hosted HDD
  - Optional: sync data to S3

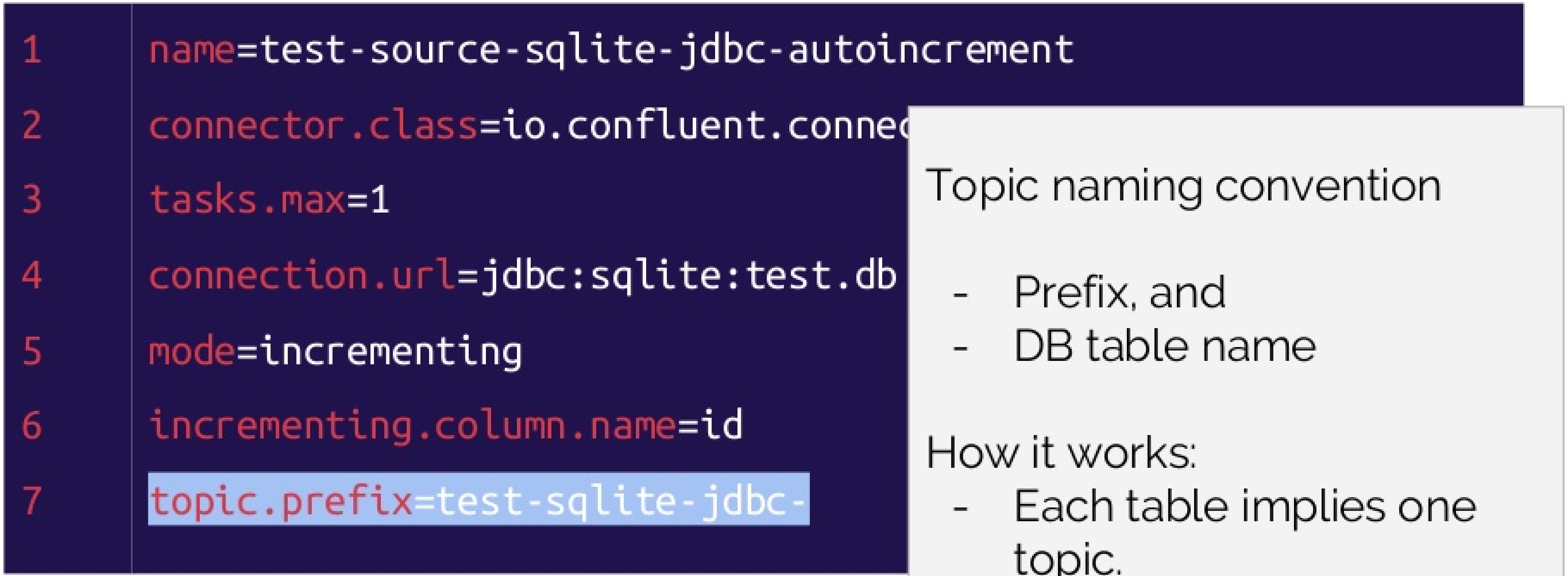
Kafka Connect



```
name=test-source-sqlite-jdbc-autoincrement
connector.class=io.confluent.connect.jdbc.JdbcSourceConnector
tasks.max=1
connection.url=jdbc:sqlite:test.db
mode=incrementing
incrementing.column.name=id
topic.prefix=test-sqlite-jdbc-
```

Kafka Connect - Source Property File

Source: https://github.com/confluentinc/kafka-connect-jdbc/blob/master/config/source-quickstart-sqlite.properties



```
name=test-source-sqlite-jdbc-autoincrement
connector.class=io.confluent.connec
tasks.max=1
connection.url=jdbc:sqlite:test.db
mode=incrementing
incrementing.column.name=id
topic.prefix=test-sqlite-jdbc-
```

Topic naming convention

- Prefix, and
- DB table name

How it works:

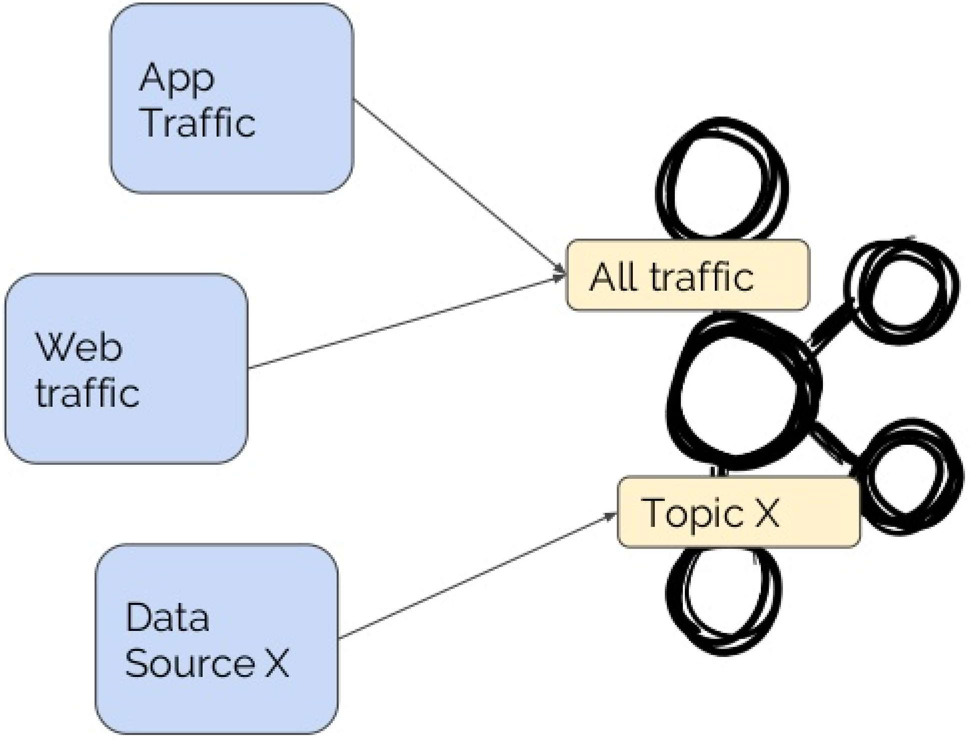
- Each table implies one topic.

Kafka Connect - Source Property File

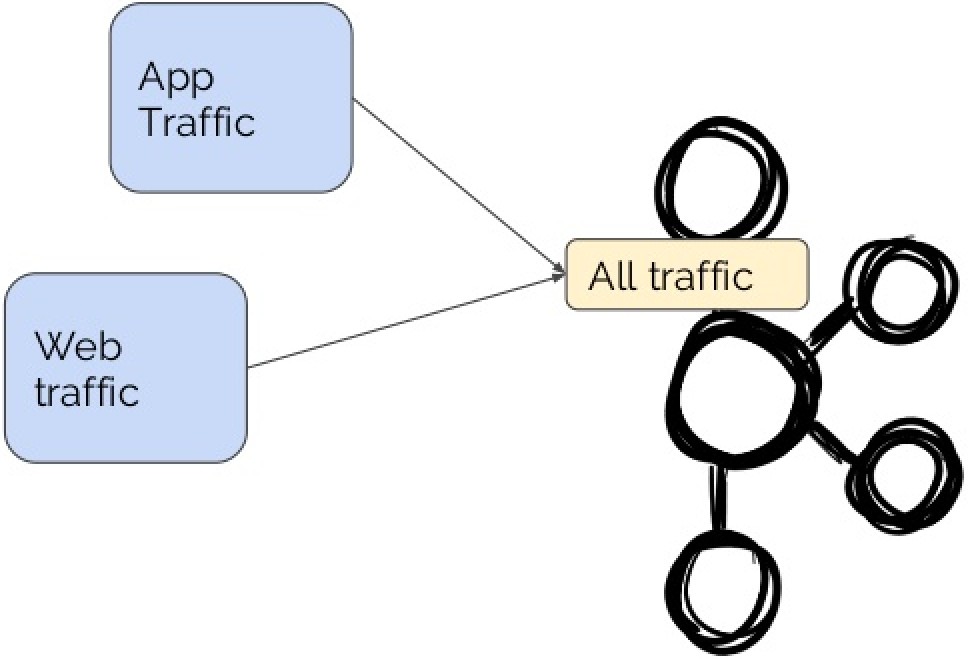
Source: https://github.com/confluentinc/kafka-connect-jdbc/blob/master/config/source-quickstart-sqlite.properties

Data Topic Model

- One-to-one (most common)
- Many-to-one



Kafka Connect



Schema-less

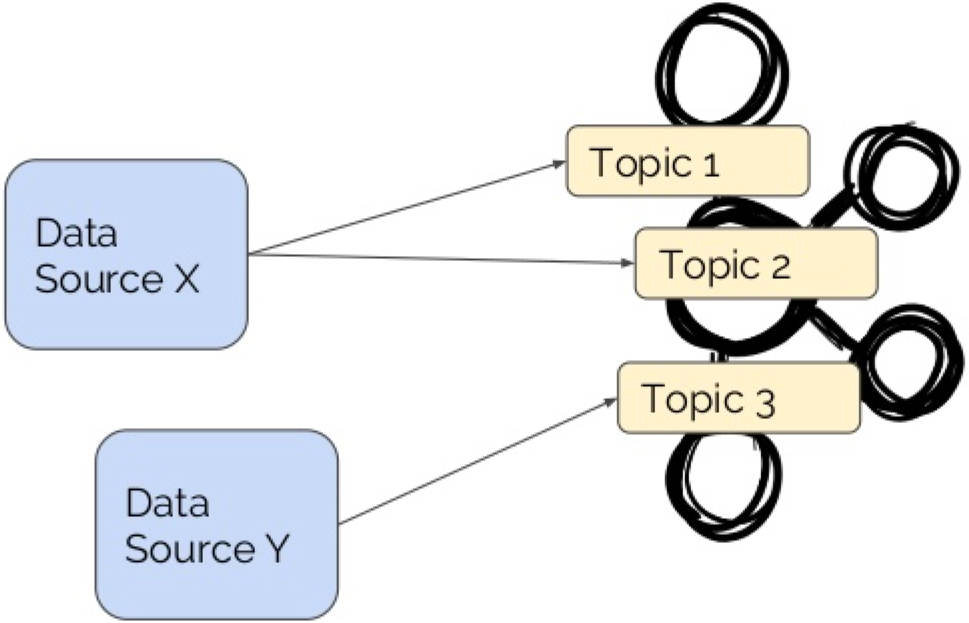
- Practically, you can write any types of data to the topic
- Most common choice is Avro

Btw, Avro is an open-source library for schema specification and data serialization.

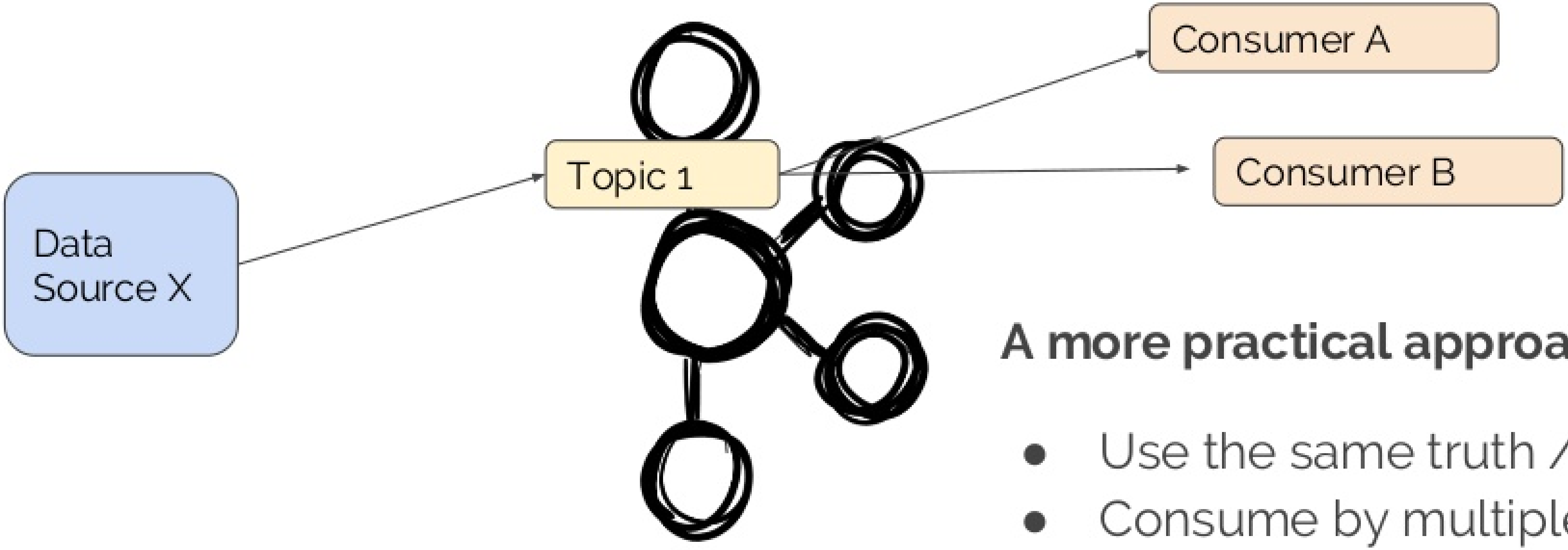
Kafka Connect

Data Topic Model

- One-to-one (most common)
- Many-to-one
- One-to-many (most rare)



Kafka Connect



A more practical approach

- Use the same truth / data
- Consume by multiple guys!

Kafka Connect

Takeaway Messages

- Producers and consumers are actors
  - Push data to or pull data from Kafka

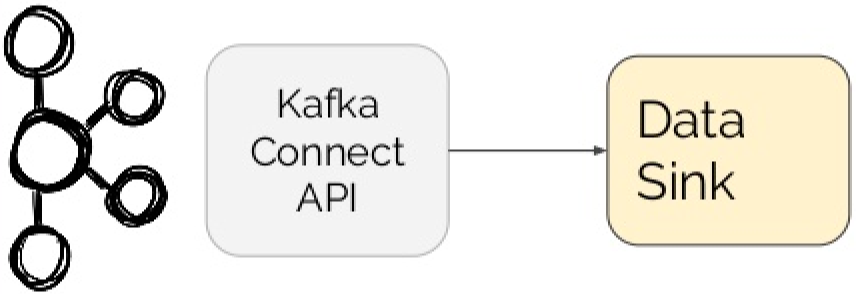
- Connect API automates the above actions
  - Work nicely with databases

Data Pipeline Use Cases

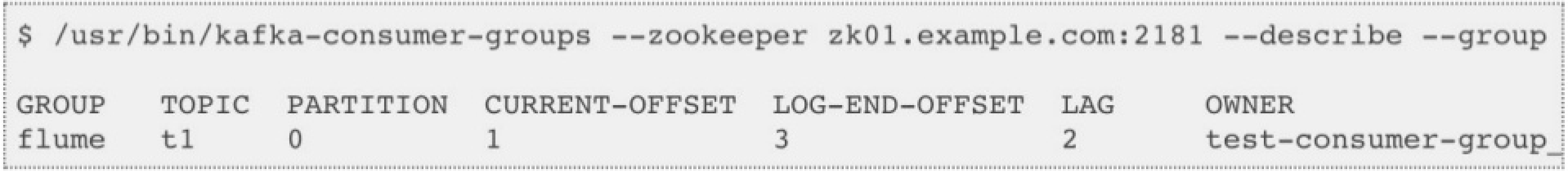
Kafka Internal - consumer’s state				
Consumer	Topic	Current Topic Position	Your last-read position	Lag behind by
hello_world	foobar	1080	1000	80

Kafka keeps track on consumer's state:

- A consumer can always resume work-in-progress
- New consumer can start fresh!



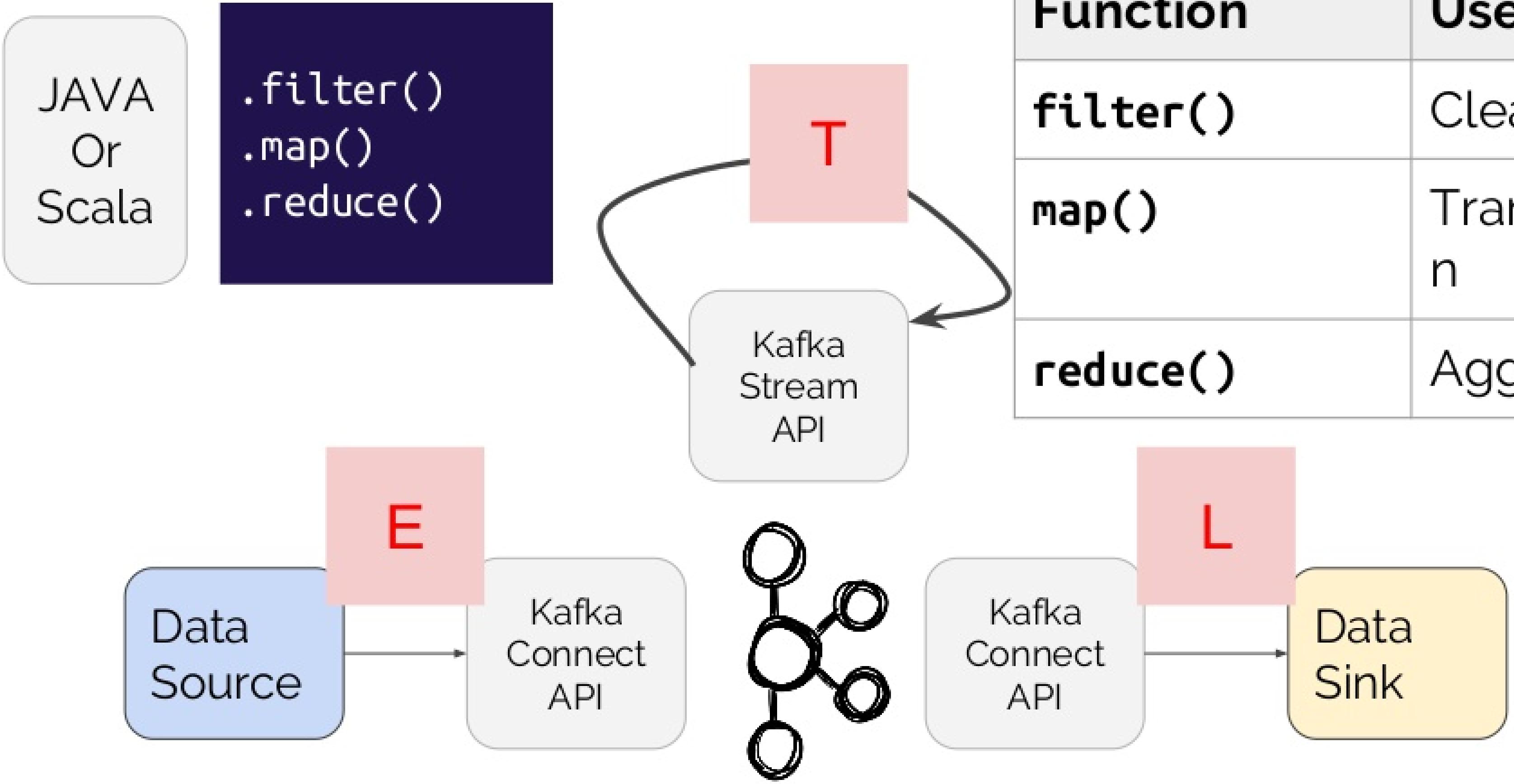
Kafka as a data pipeline - data resiliency



```
$ /usr/bin/kafka-consumer-groups --zookeeper zk01.example.com:2181 --describe --group

GROUP    TOPIC   PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG     OWNER
flume    t1      0          1               3               2       test-consumer-group_
```

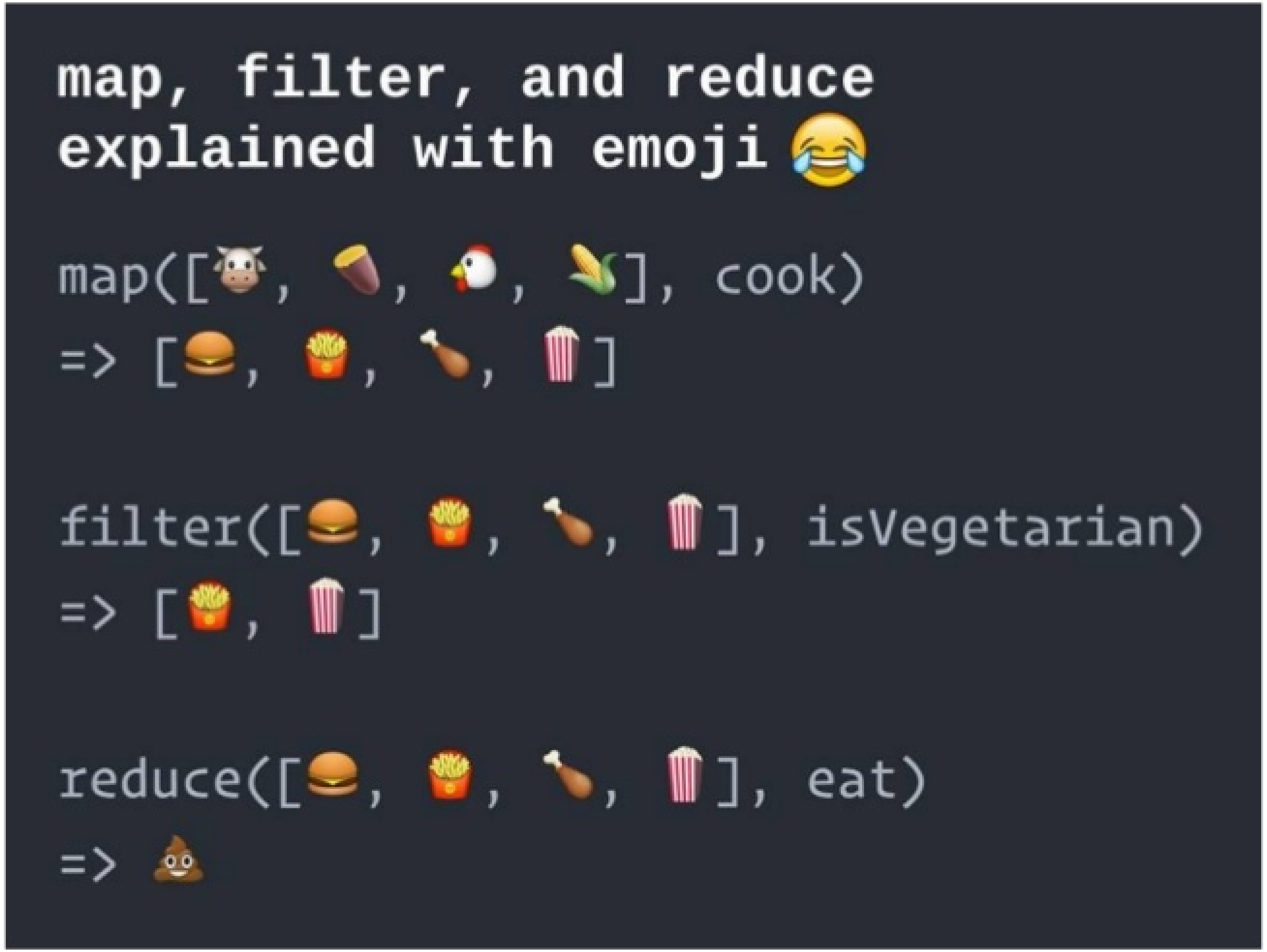
Source:
https://www.cloudera.com/documentation/kafka/latest/topics/kafka_command_line.html



```
.filter()
.map()
.reduce()
```

Function	Use Case
`filter()`	Cleansing
`map()`	Transformation
`reduce()`	Aggregation

Kafka as a data pipeline - Replace ETL



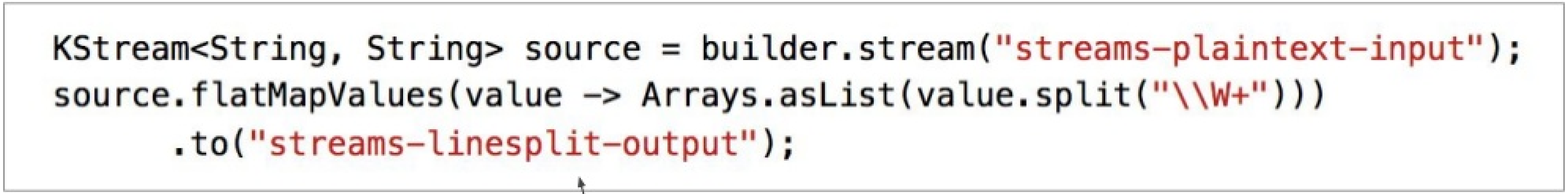
map, filter, and reduce explained with emoji 😂

```
map([🐮, 🍠, 🐔, 🌽], cook)
=> [🍔, 🍟, 🍗, 🍿]

filter([🍔, 🍟, 🍗, 🍿], isVegetarian)
=> [🍟, 🍿]

reduce([🍔, 🍟, 🍗, 🍿], eat)
=> 💩
```

Source: https://i.redd.it/yf7rw3pjiapx.jpg

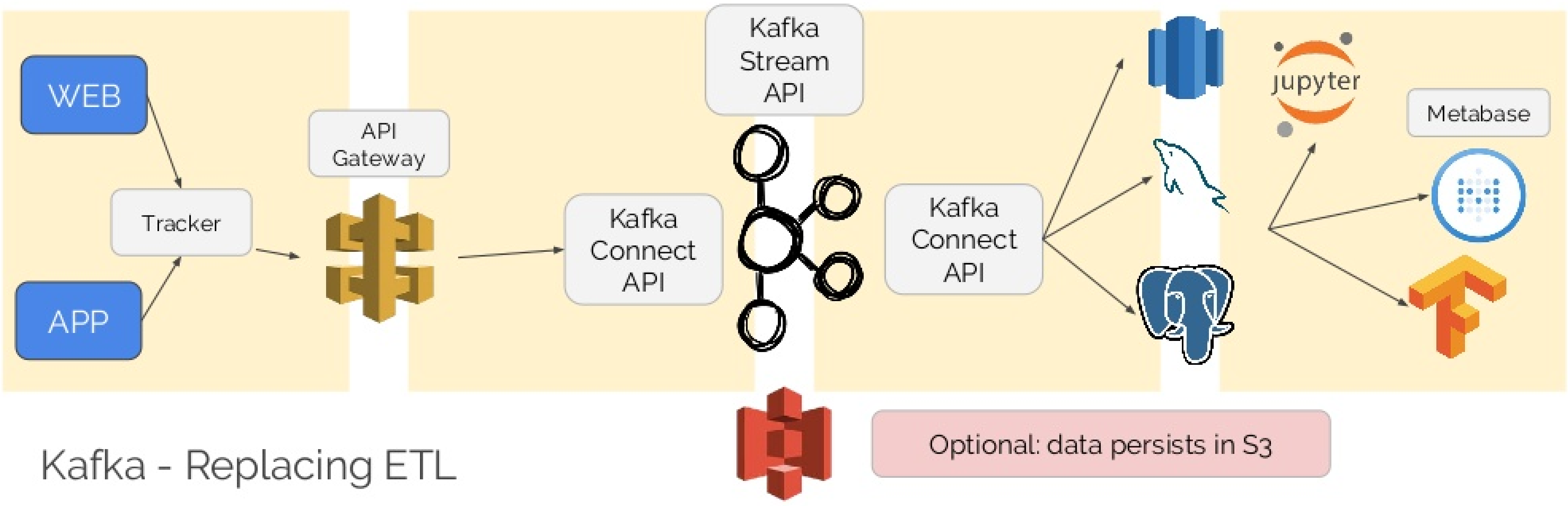


```
KStream<String, String> source = builder.stream("streams-plaintext-input");
source.flatMapValues(value -> Arrays.asList(value.split("\\W+")))
      .to("streams-linesplit-output");
```

A New Topic is Created!

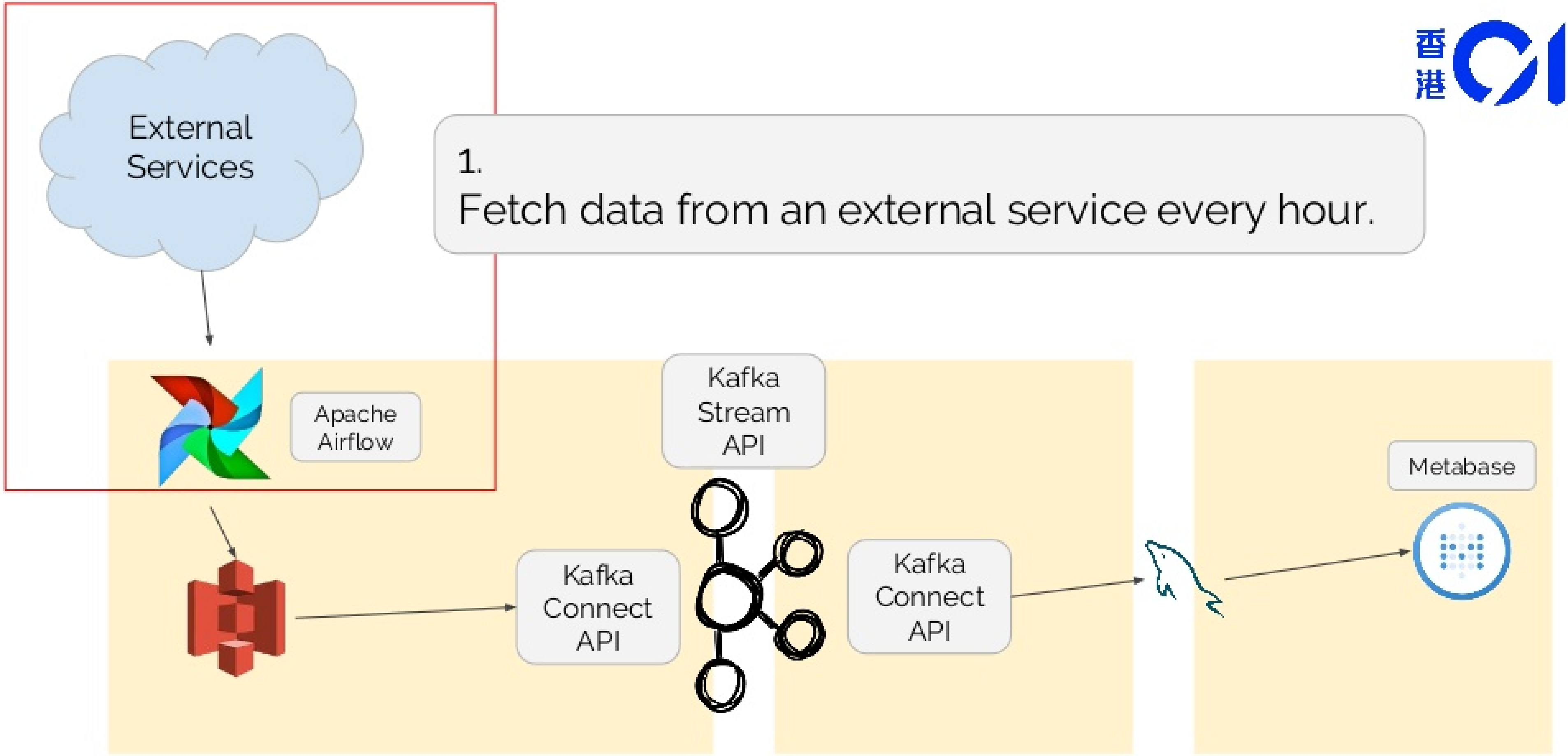
Kafka - Streaming Example Code
Source: https://kafka.apache.org/11/documentation/streams/tutorial

Data Pipeline with Kafka v2



Kafka - Replacing ETL

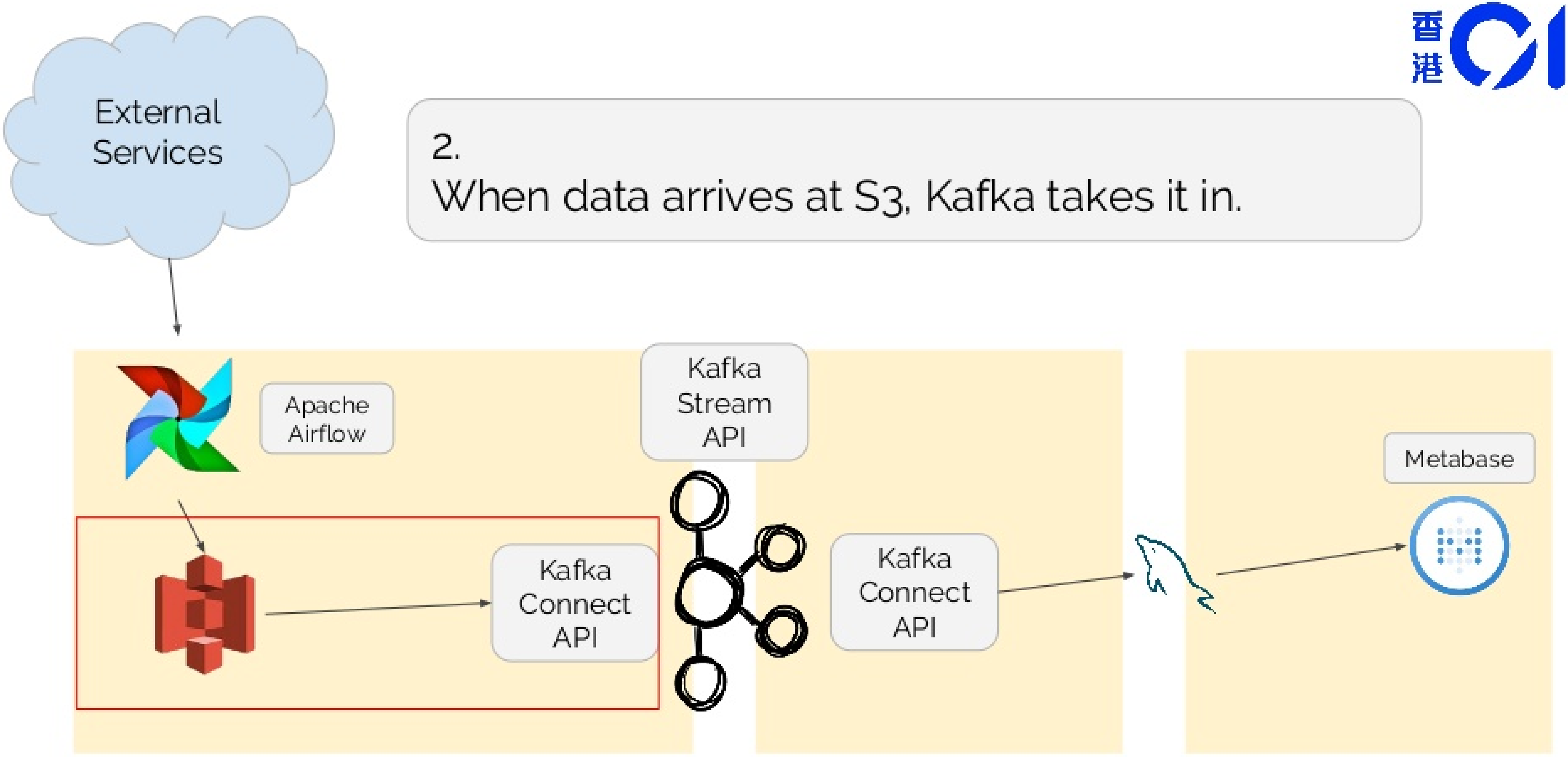
Experimenting Kafka in HK01



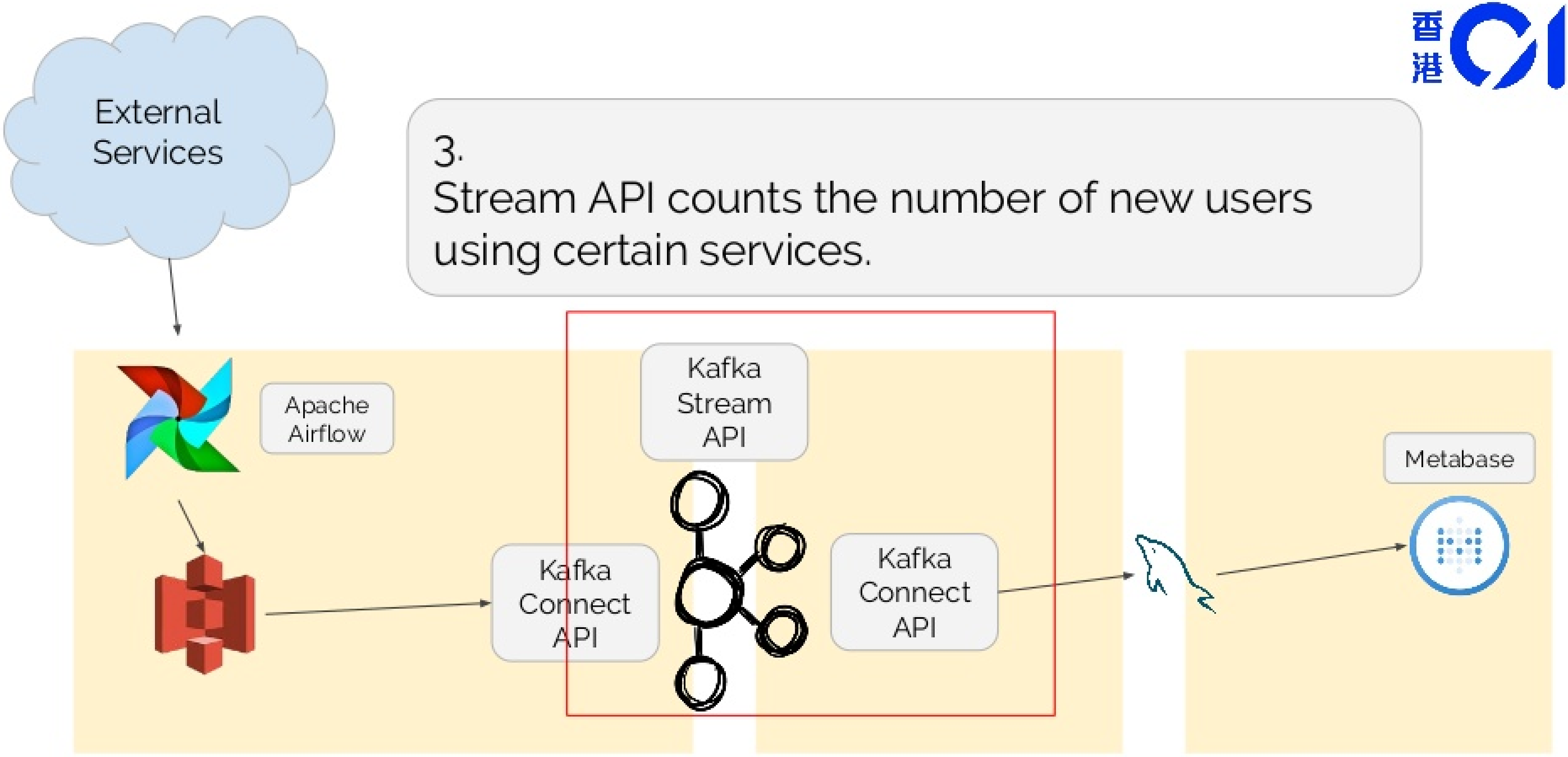
1.
Fetch data from an external service every hour.

Experimenting Kafka in HK01

2.
When data arrives at S3, Kafka takes it in.



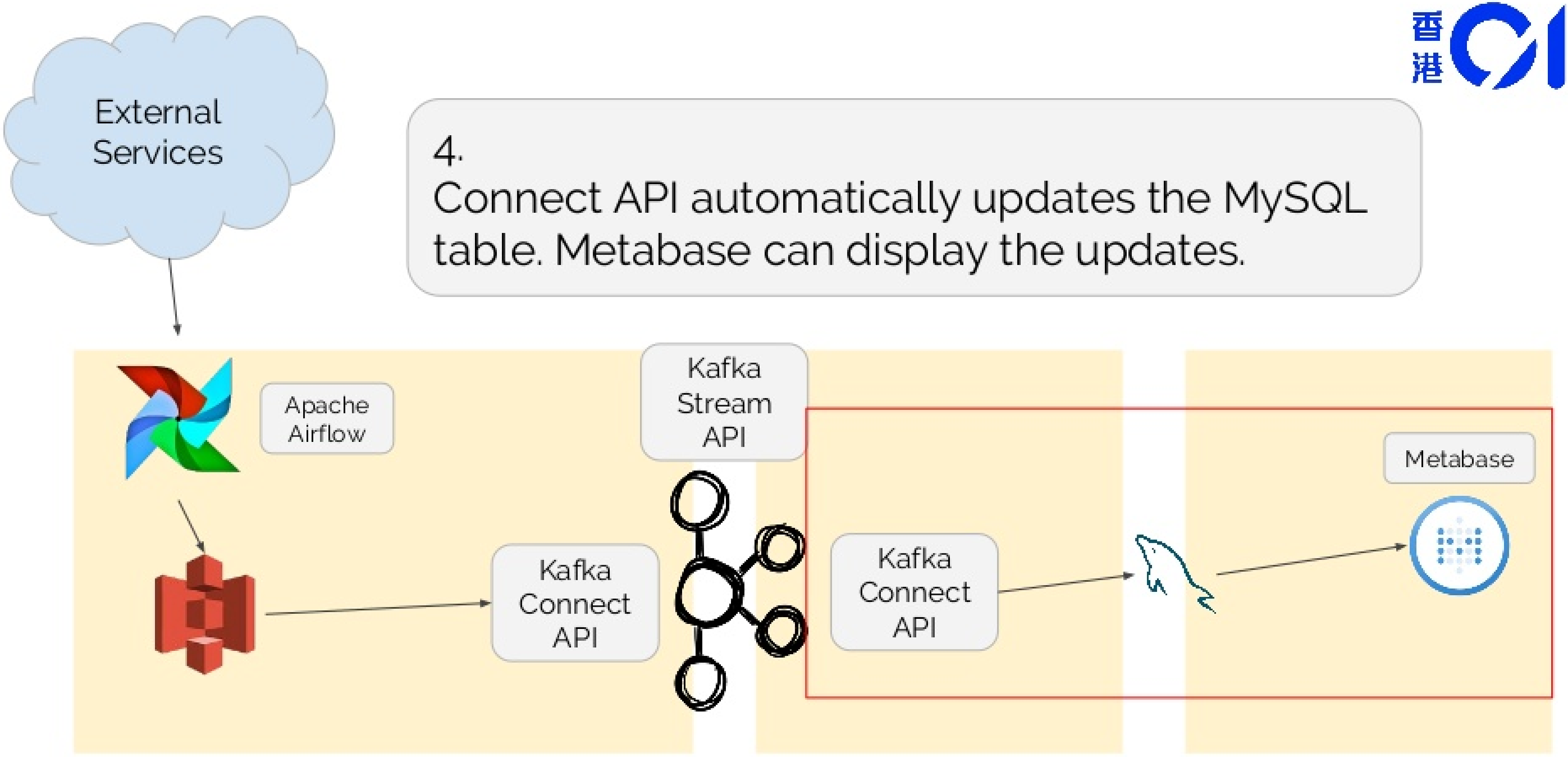
Experimenting Kafka in HK01



3.
Stream API counts the number of new users using certain services.

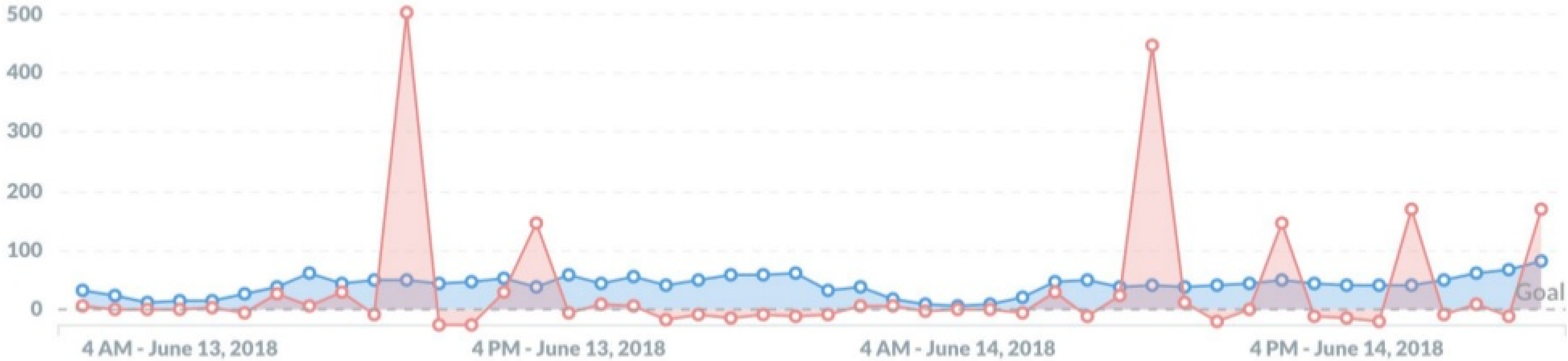
Experimenting Kafka in HK01

4.
Connect API automatically updates the MySQL table. Metabase can display the updates.

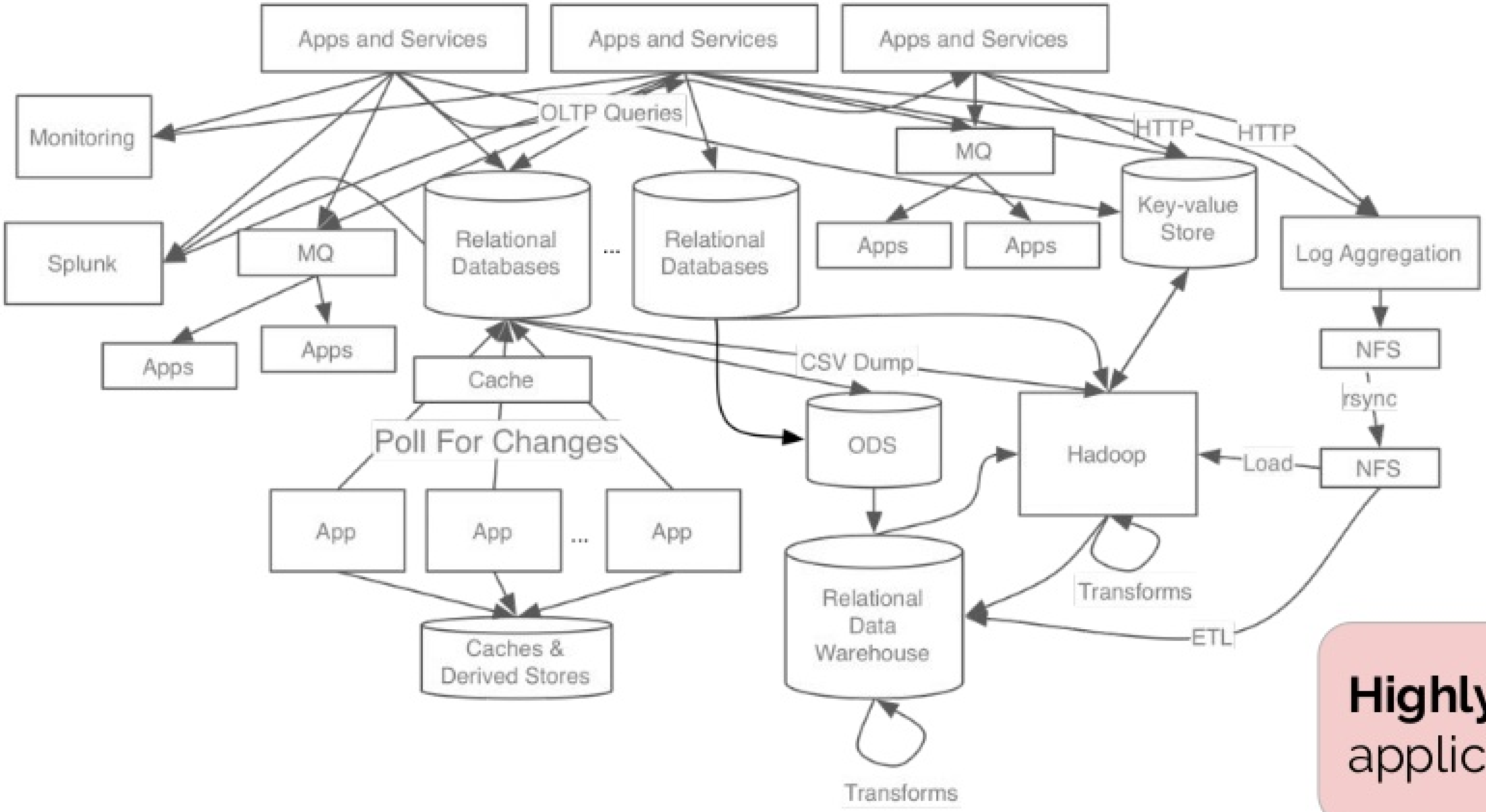


Experimenting Kafka in HK01

Will display live dashboard during the talk

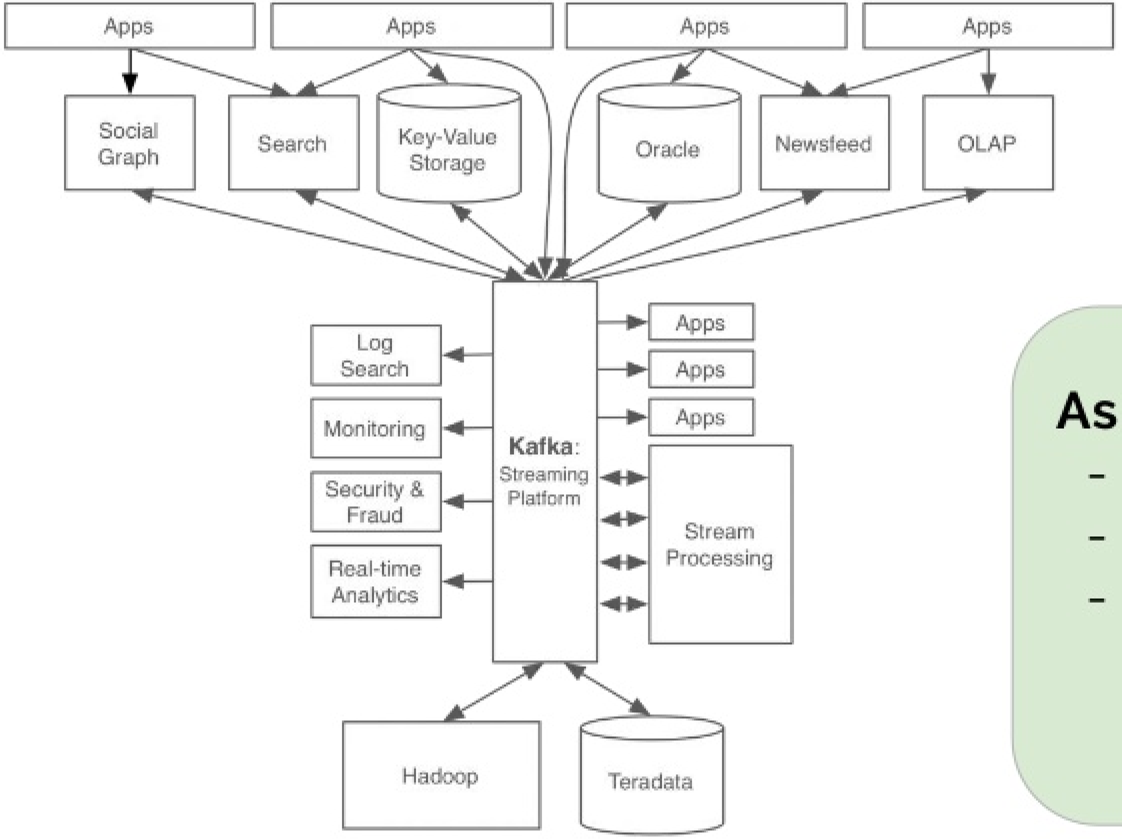


Other Use Cases



Highly-coupled: application & storage

Message Queue | Source: https://www.confluent.io/blog/stream-data-platform-1/



As a message queue (MQ):

- Pub/Sub
- Transformation
- Roles; clear that who are the sources and the sinks, respectively

Message Queue | Source: https://www.confluent.io/blog/stream-data-platform-1/

Things that we didn't explore

- Logs aggregation
- Database log compaction
- Event sourcing

Other Use Cases | Source: https://kafka.apache.org/uses

Key Takeaways

Pros

1. Kafka simplifies your ETL tasks.
2. Kafka unitifies your data storage.
3. Kafka gives your other possibilities.

Key Takeaways

Cons

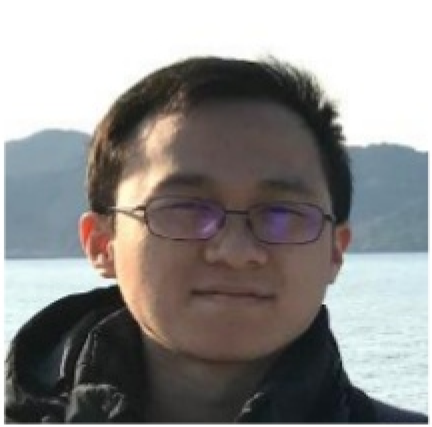
1. Ops problems - scalability, HA, Zookeeper, etc.

2. Learning curve is *STEEP*.

We Love to Share

Mole Wong
Data Pipeline with Apache Kafka

Day 1 17:40
Conference Hall 4-5



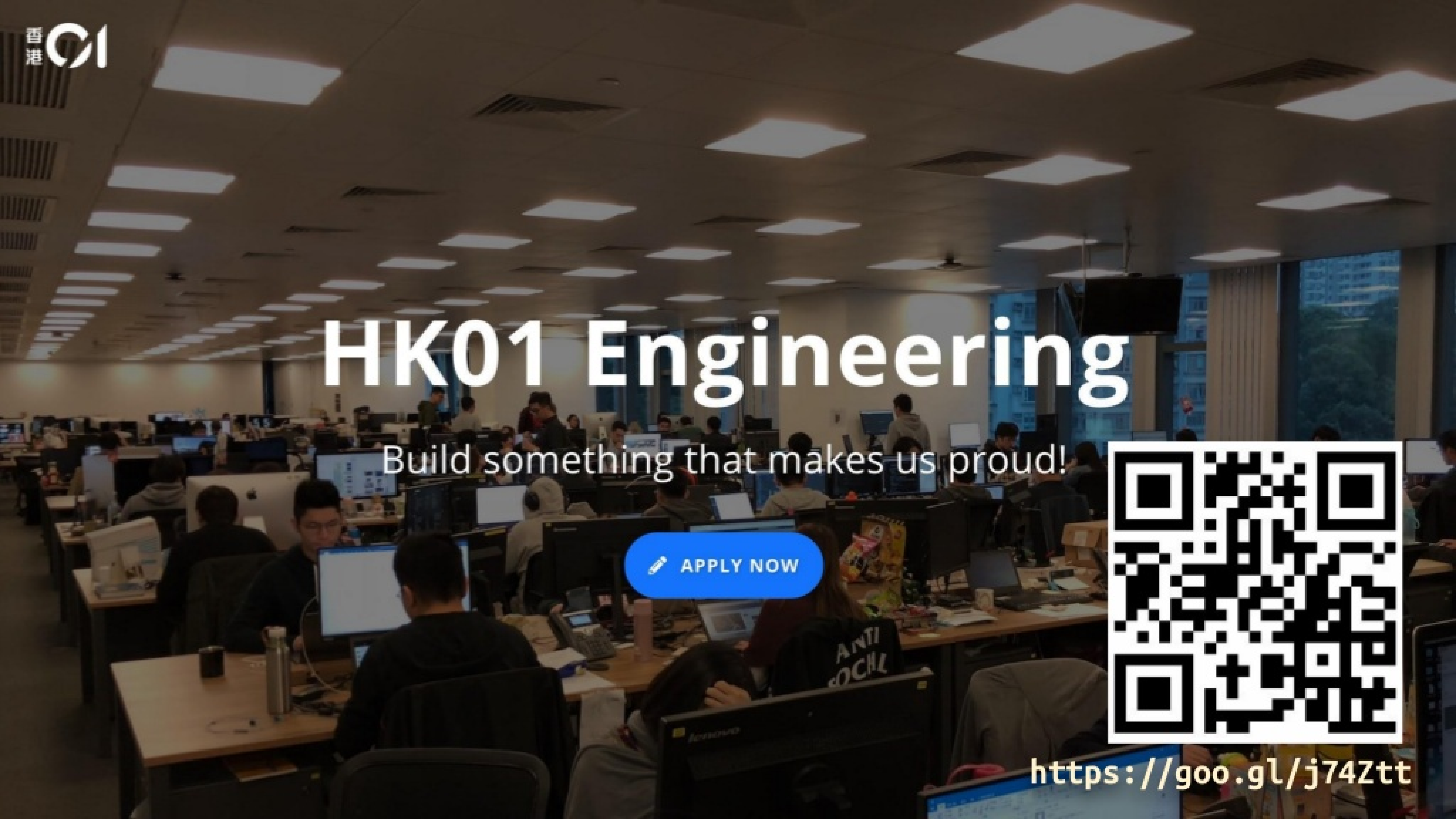
Sunday Ku
Video.js with HLS

Day 2 12:30
Conference Hall 4-5

Ivan Ha
React Async Rendering - Paradigm Shift After React Fiber

Day 2 15:10
Conference Hall 6

香港01



HK01 Engineering

Build something that makes us proud!

✎ APPLY NOW

https://goo.gl/j74Ztt