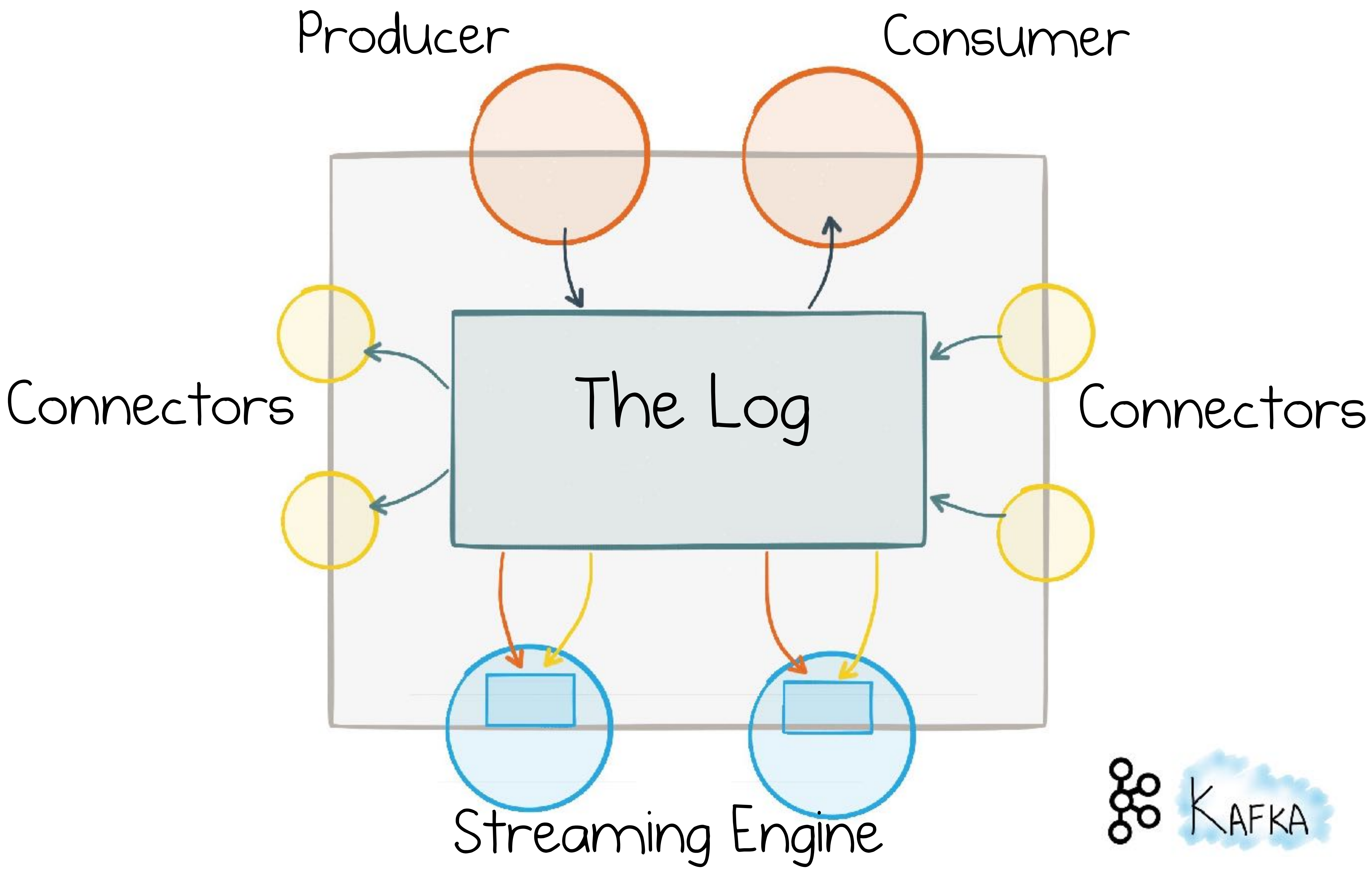
https://cnfl.io/ksql-workshop-prereq

Apache Kafka® and KSQL in Action : Let's Build a Streaming Data Pipeline!

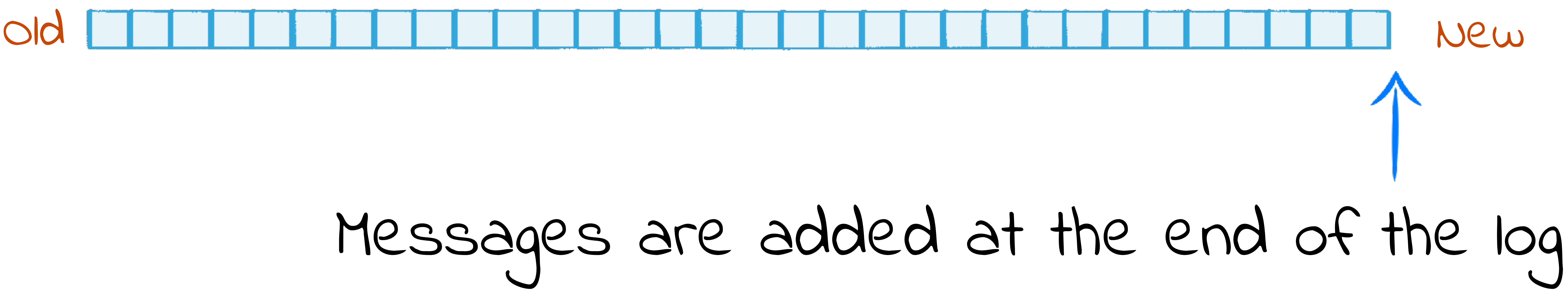
@rmoff

#NDCOslo

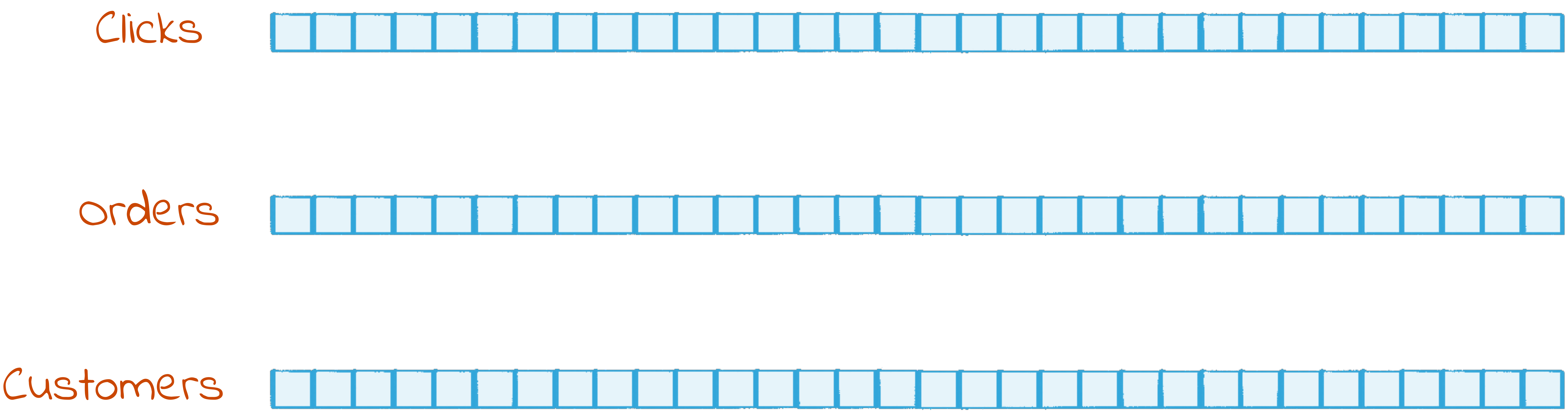
What is an Event Streaming Platform?



Immutable Event Log

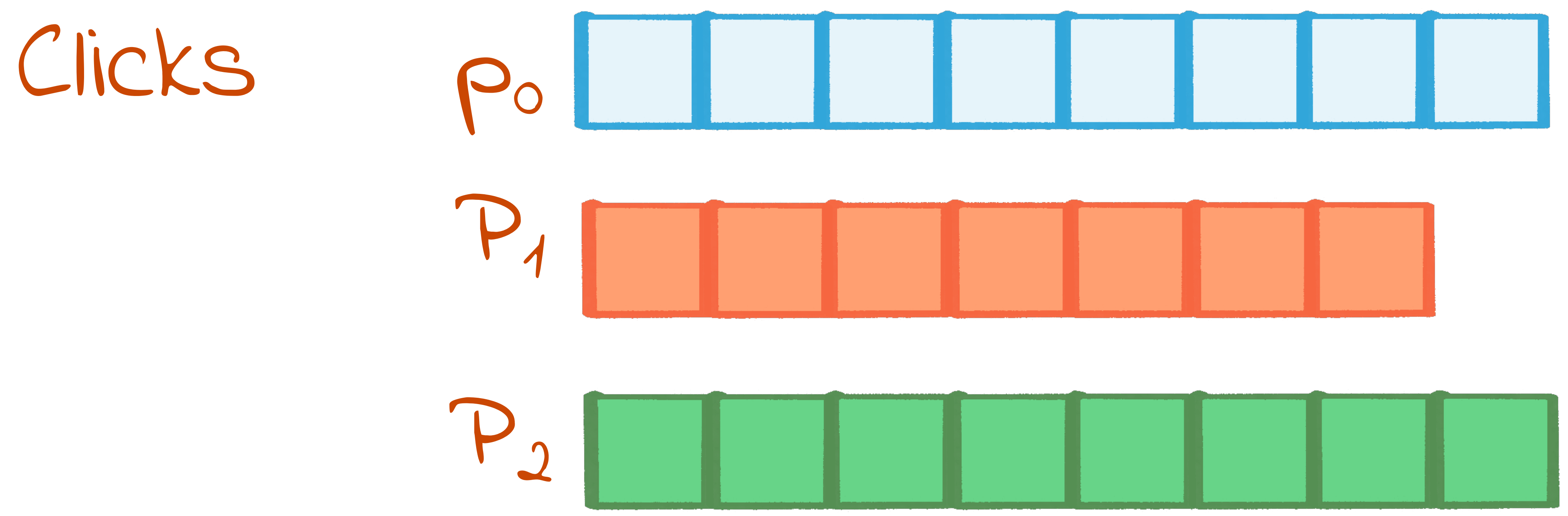


Topics



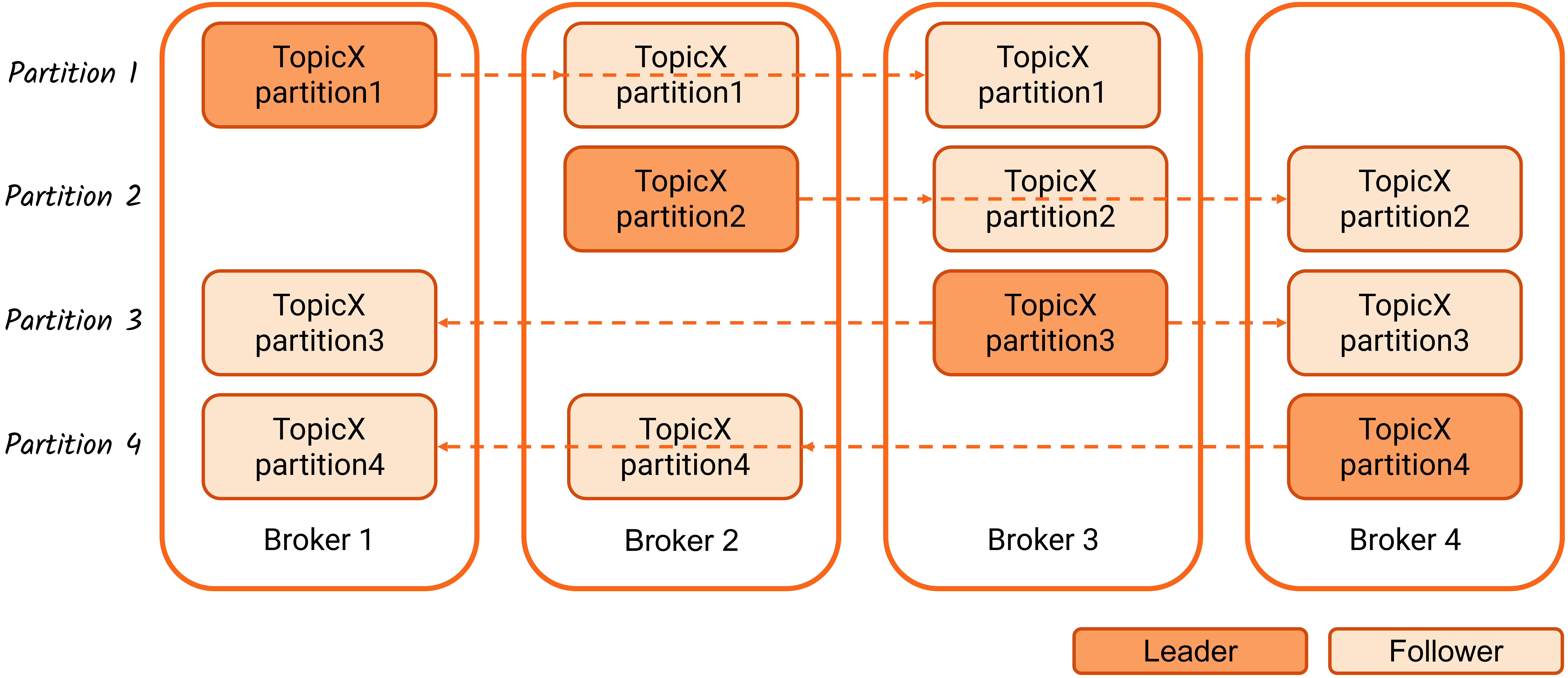
Topics are similar in concept to tables in a database

Partitions



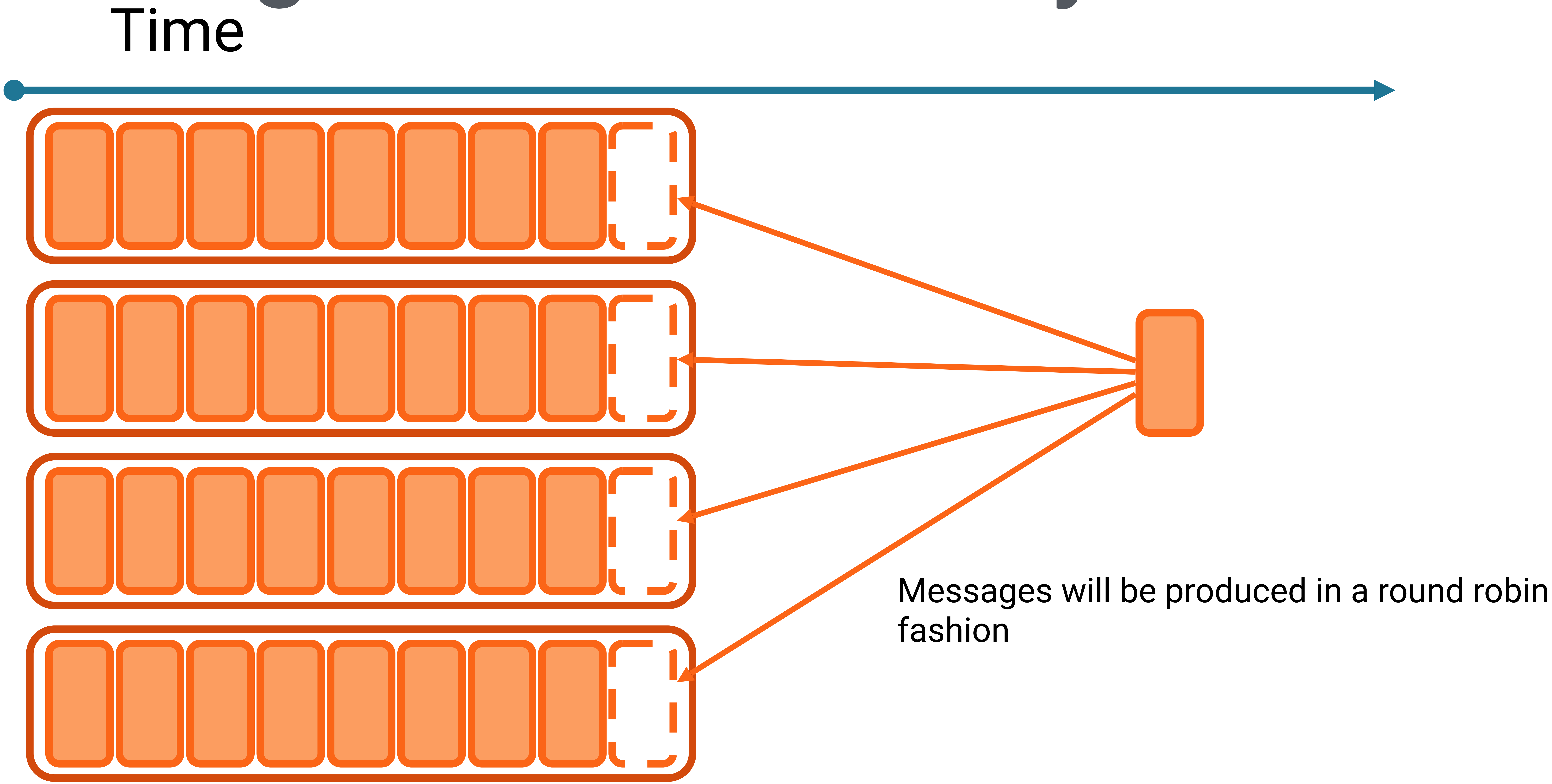
Messages are guaranteed to be strictly ordered <u>within a partition</u>

Partition Leadership and Replication

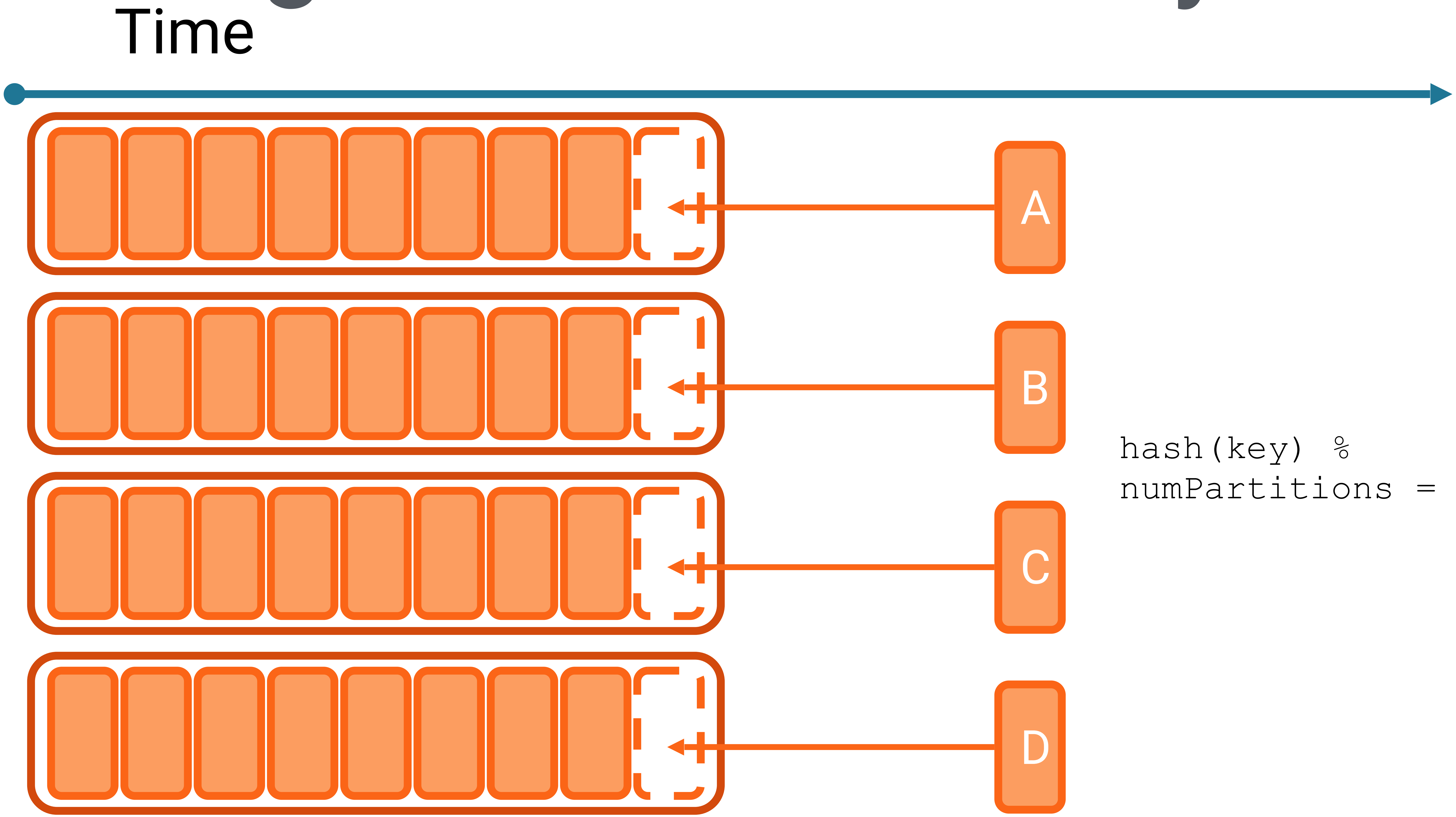


Partition Leadership and Replication - node failure

Producing to Kafka - No Key



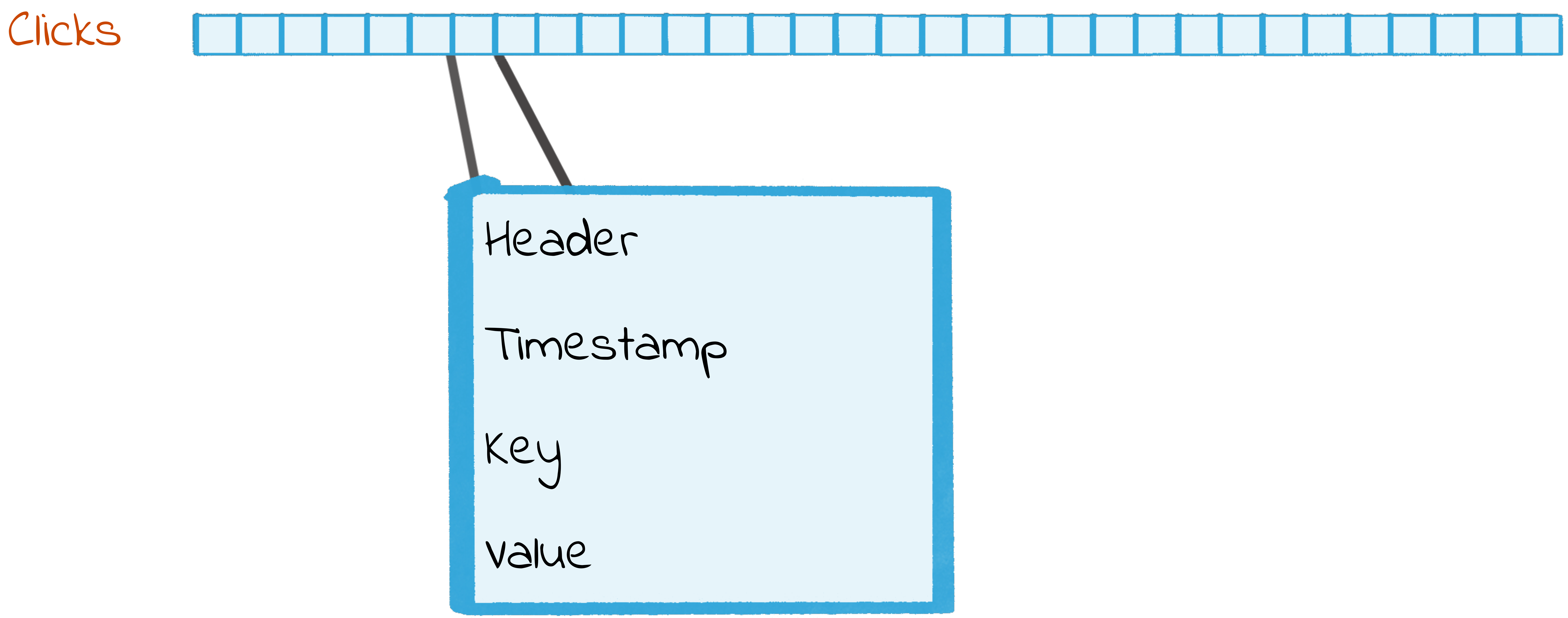
Producing to Kafka - With Key



`hash(key) % numPartitions = N`

Messages are just K/V bytes

plus headers + timestamp



Messages are just K/V bytes

With great power comes great responsibility

Avro -> Confluent Schema Registry

Protobuf

JSON

CSV

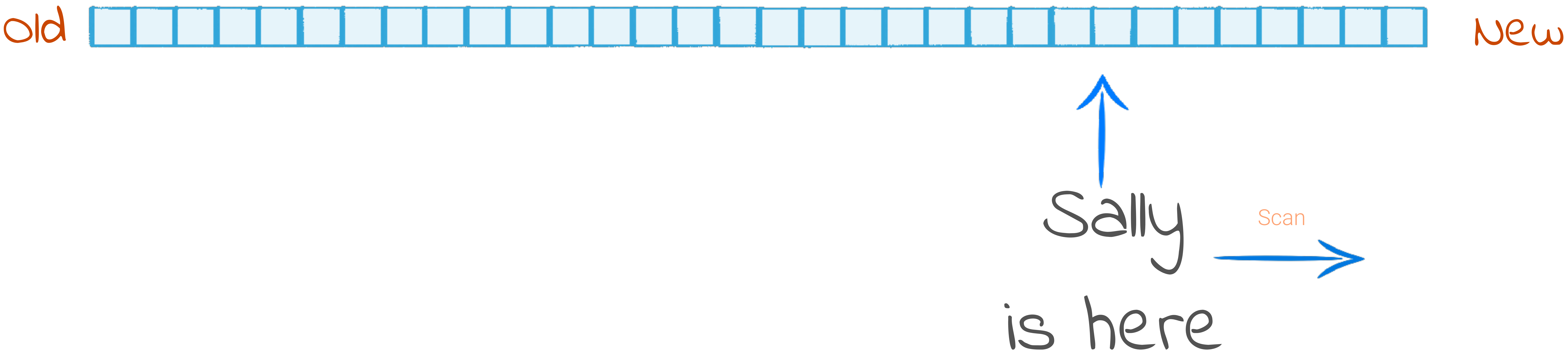
Gwen (Chen) Shapira
@gwenshap

If your dev process doesn't validate schema compatibility somewhere between your IDE and production - you are screwed and don't know it.

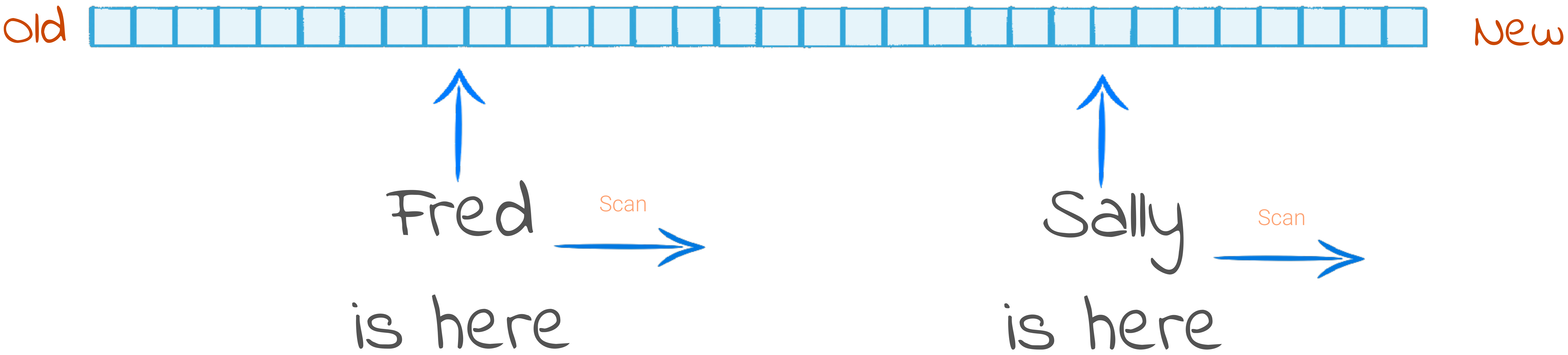
5:50 AM - 5 Apr 2017

https://qconnewyork.com/system/files/presentation-slides/qcon_17_-_schemas_and_apis.pdf

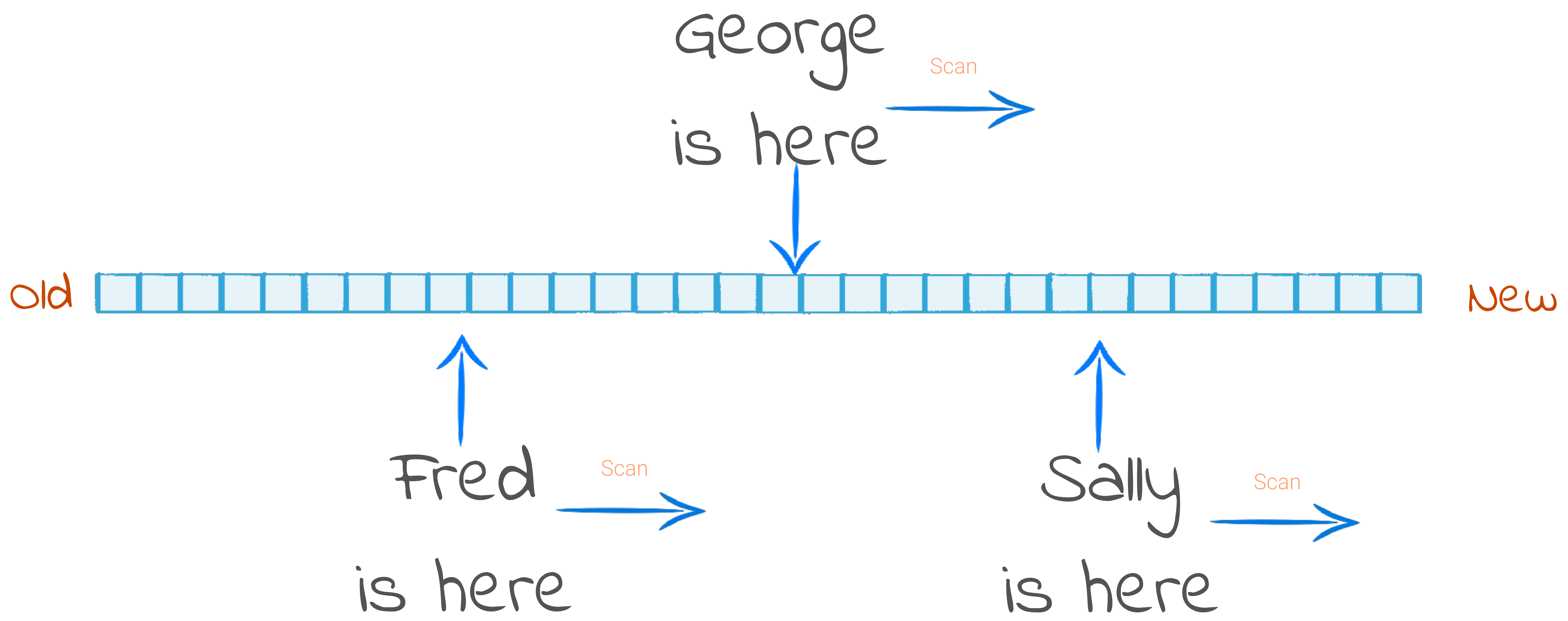
Consumers have a position all of their own



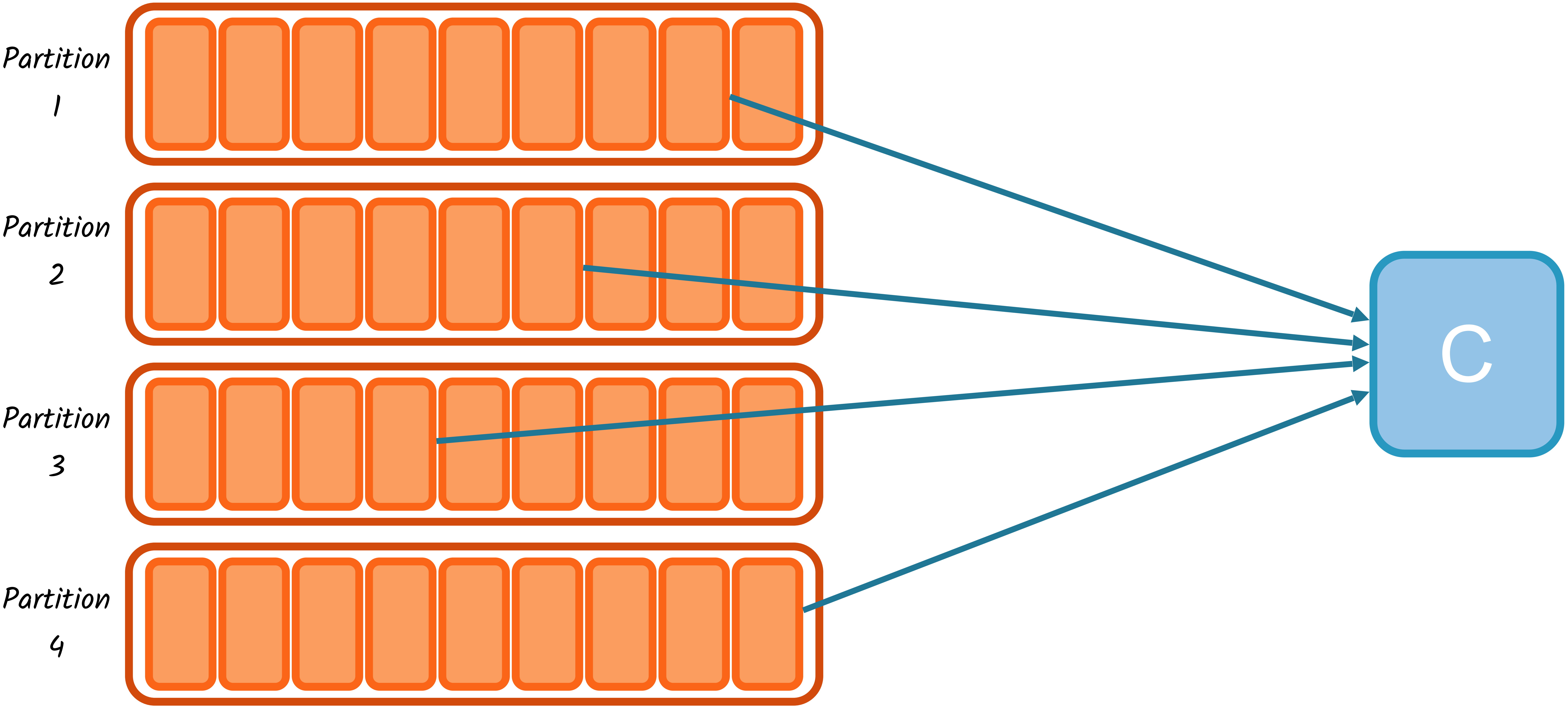
Consumers have a position all of their own



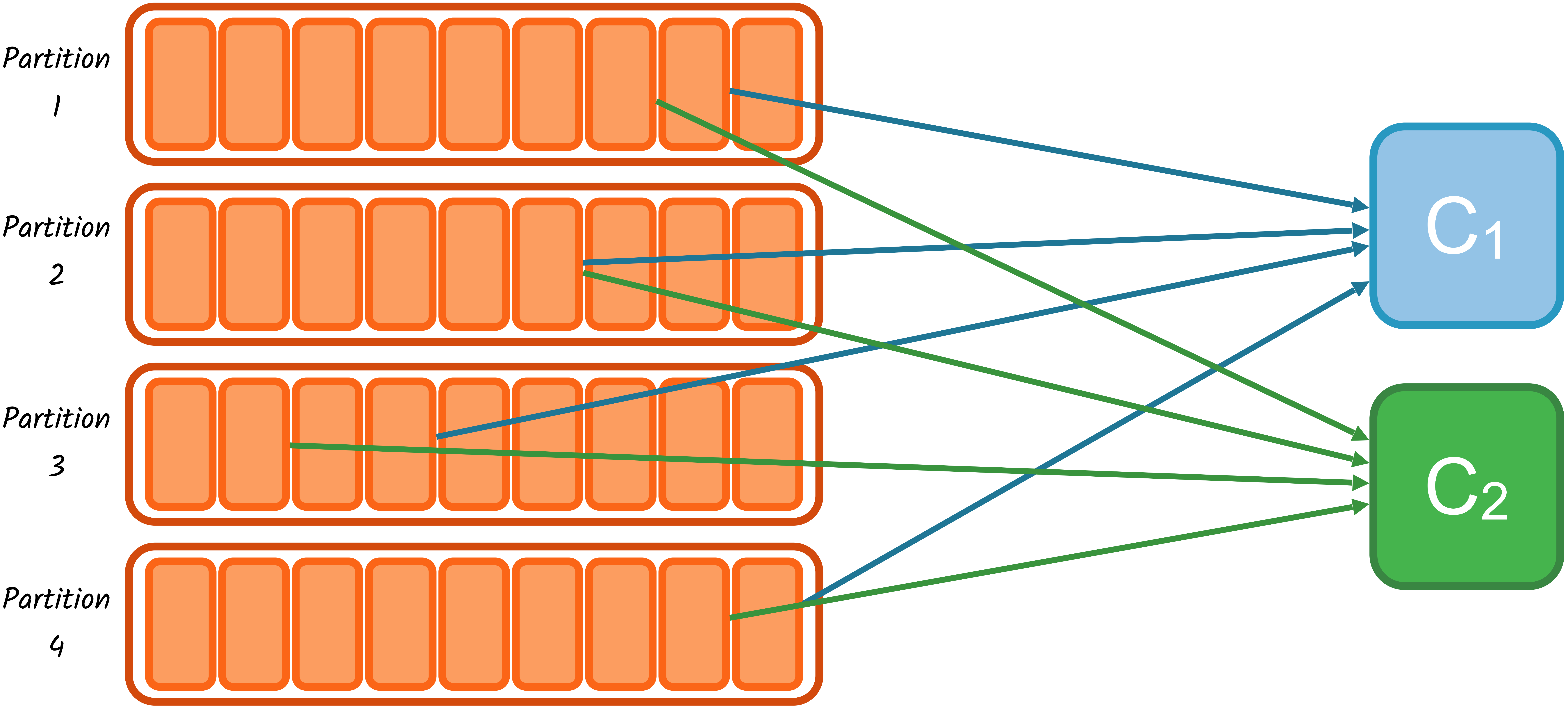
Consumers have a position all of their own



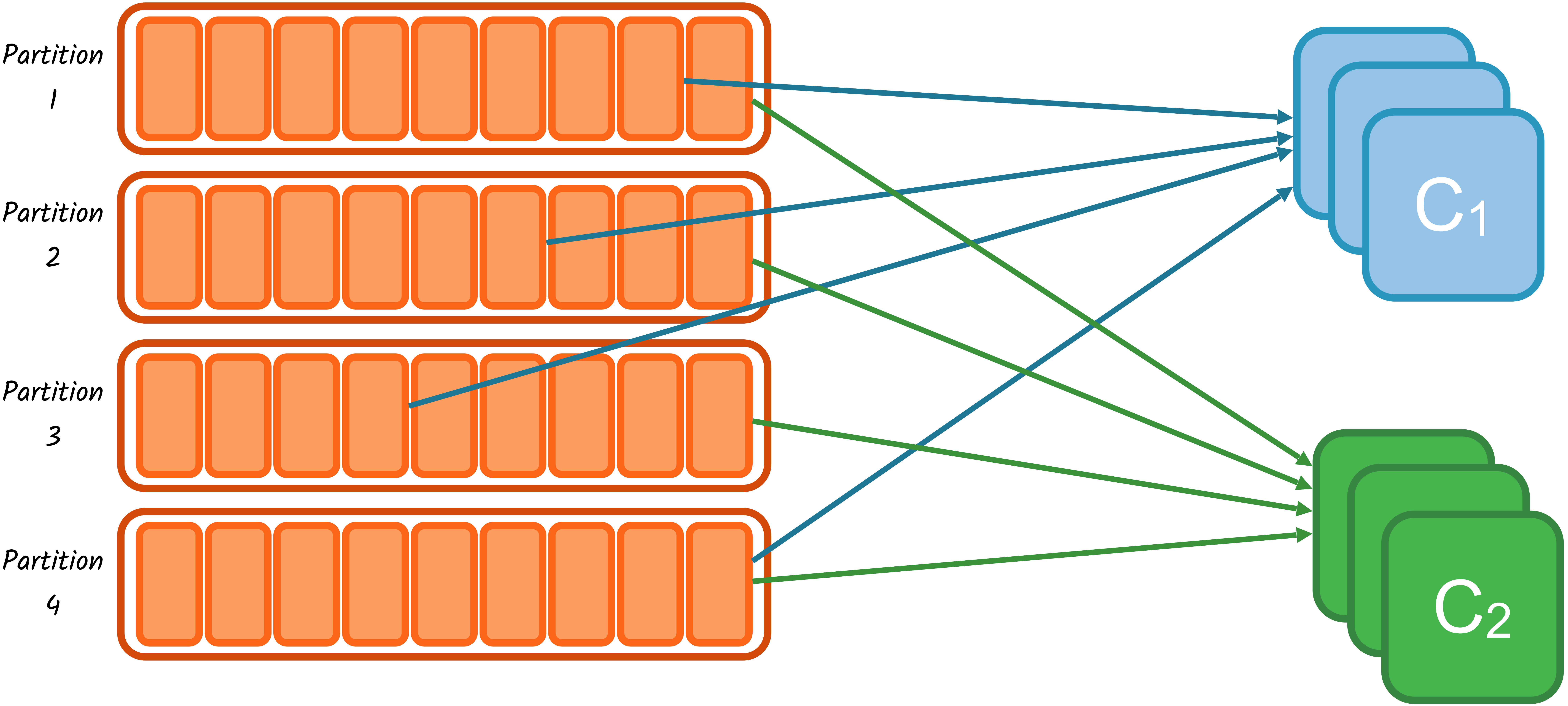
Consuming From Kafka - Single Consumer



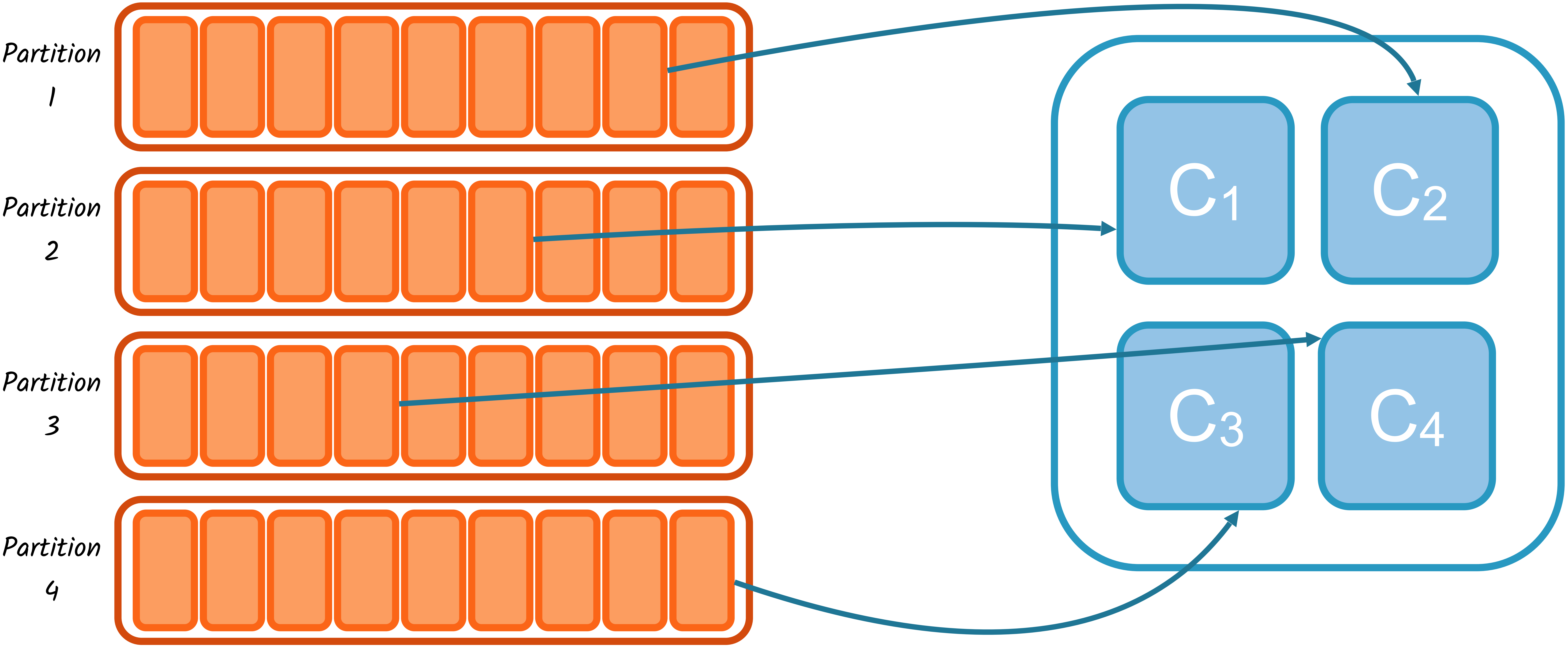
Consuming From Kafka - Multiple Consumers



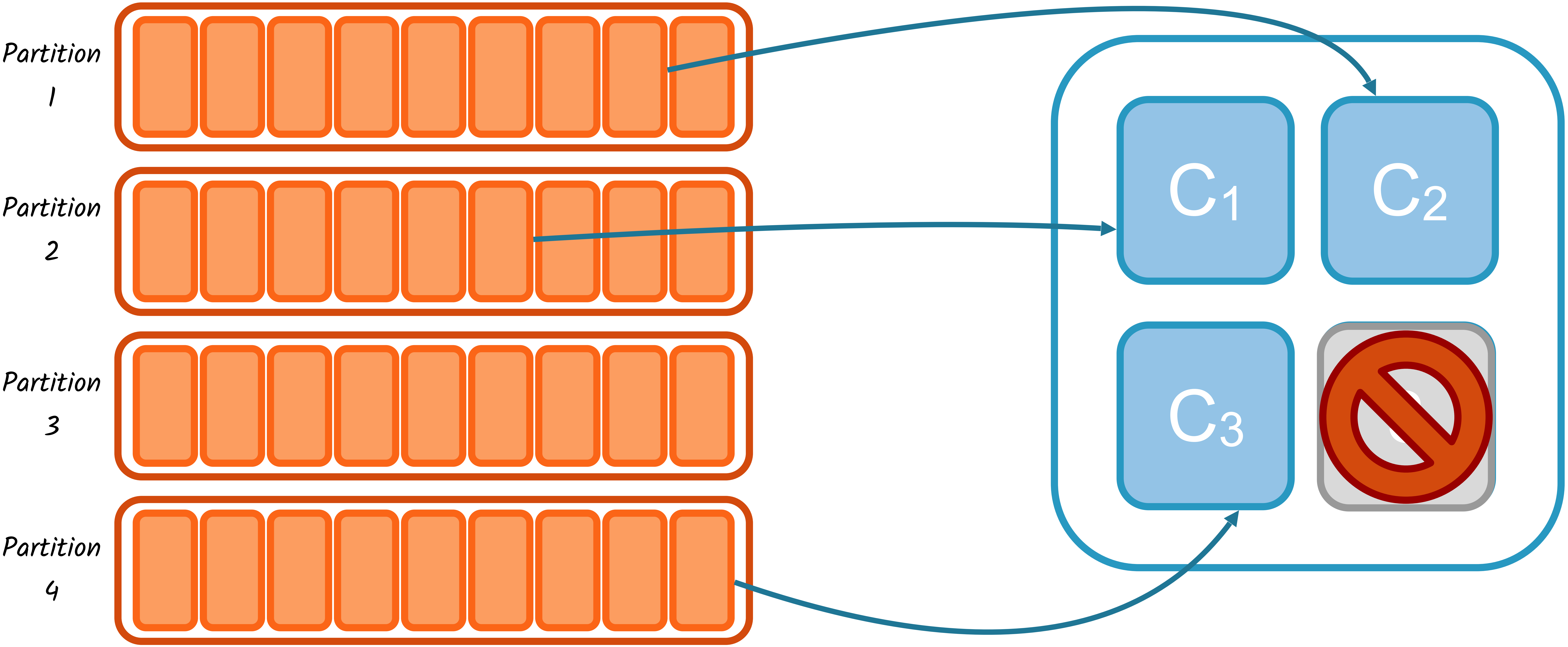
Consuming From Kafka - Grouped Consumers



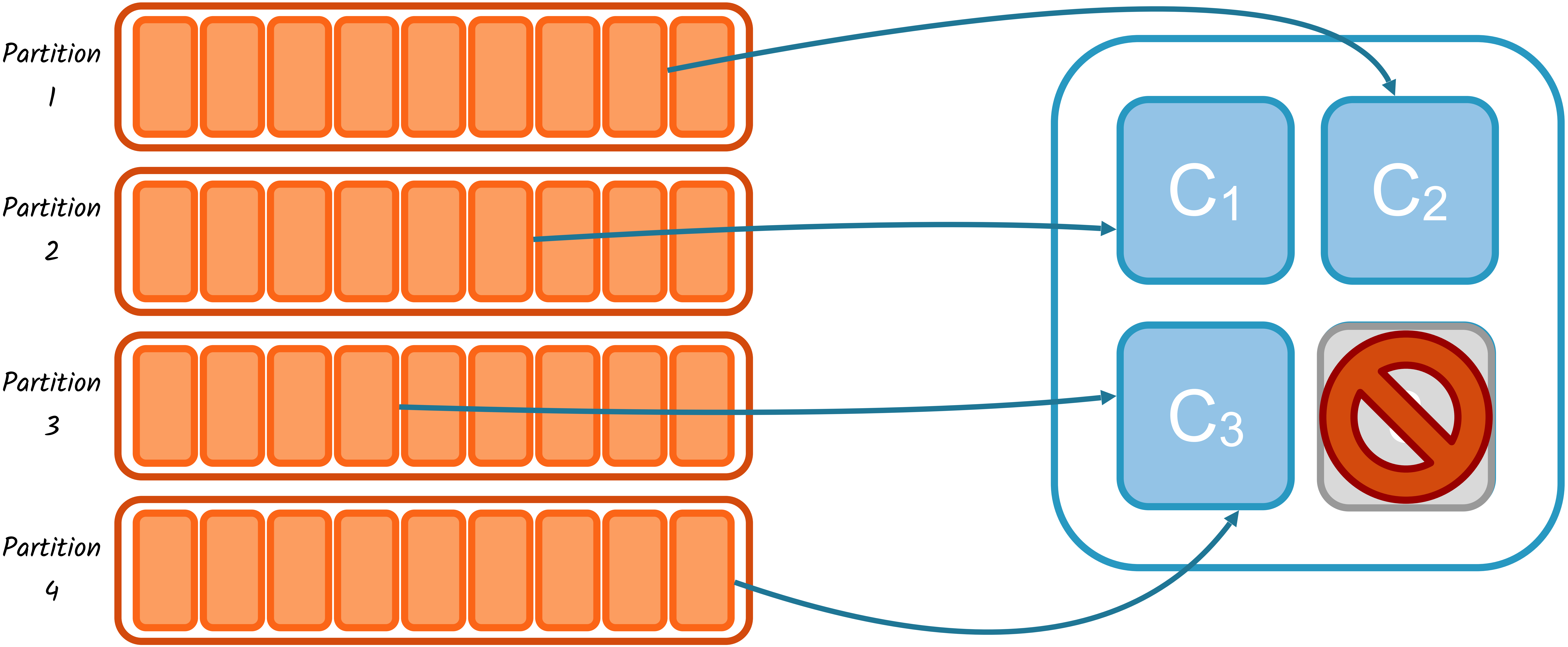
Consuming From Kafka - Grouped Consumers



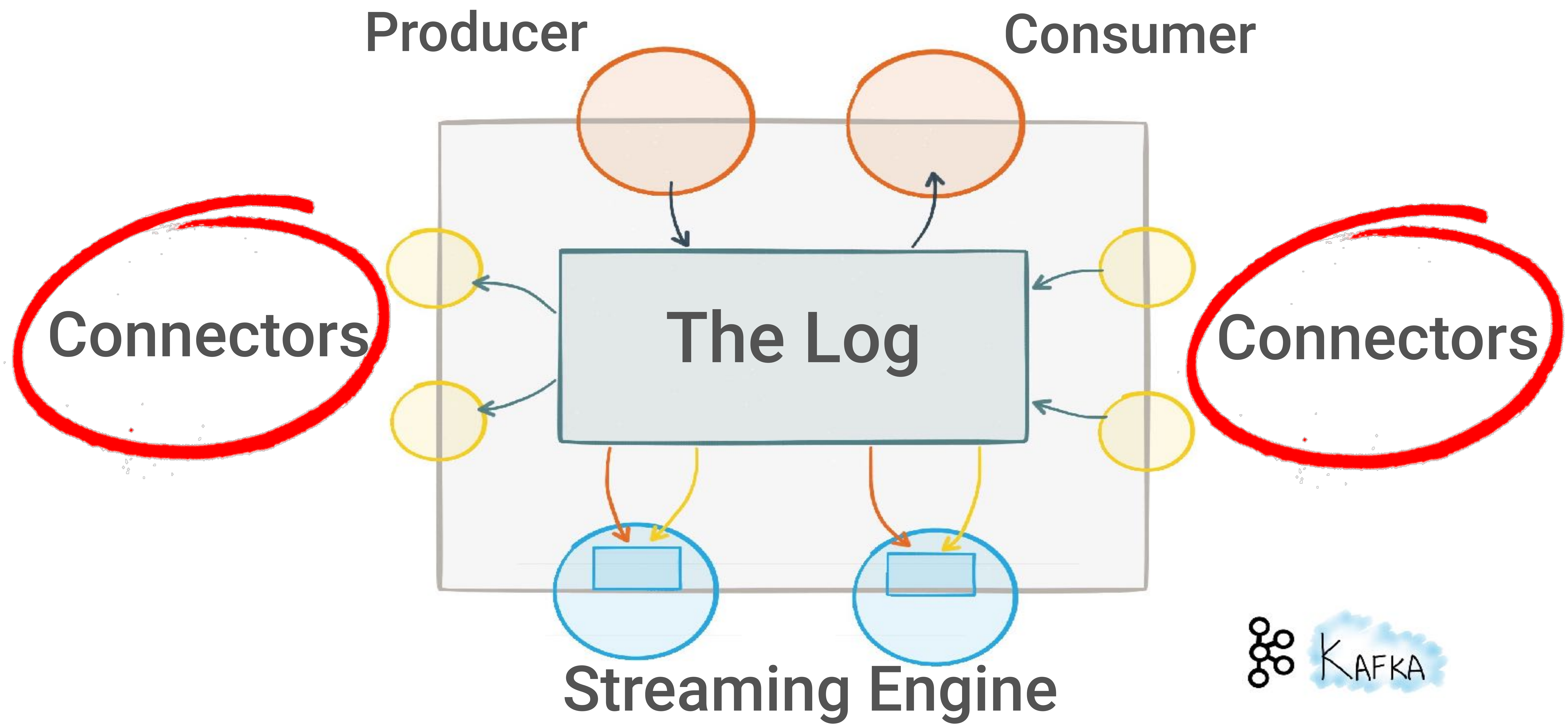
Consuming From Kafka - Grouped Consumers



Consuming From Kafka - Grouped Consumers



The Connect API



Streaming Integration with Kafka Connect

JSON MQTT.ORG

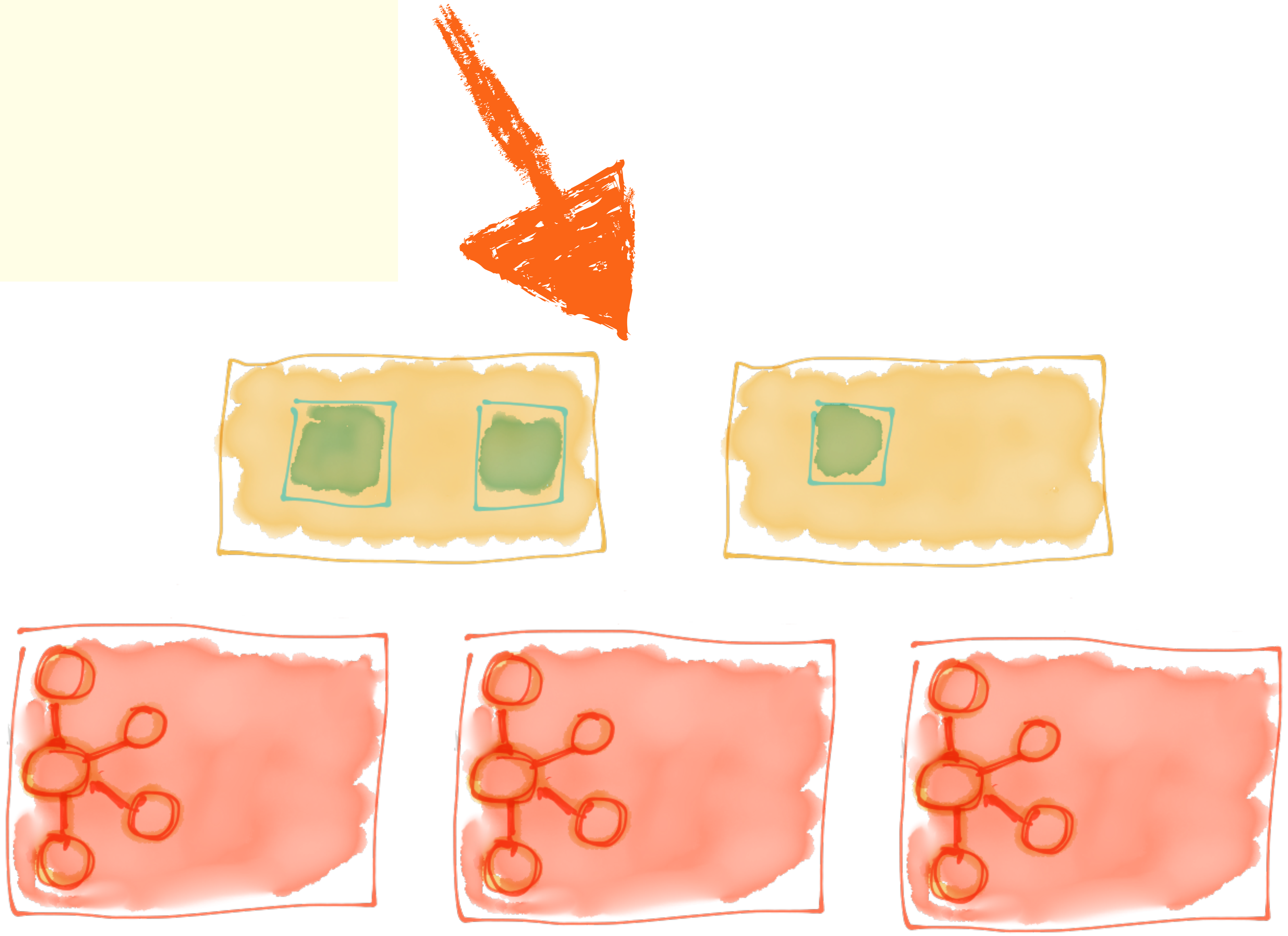
MySQL ORACLE

syslog CSV salesforce

http://

Microsoft SQL Server

Sources

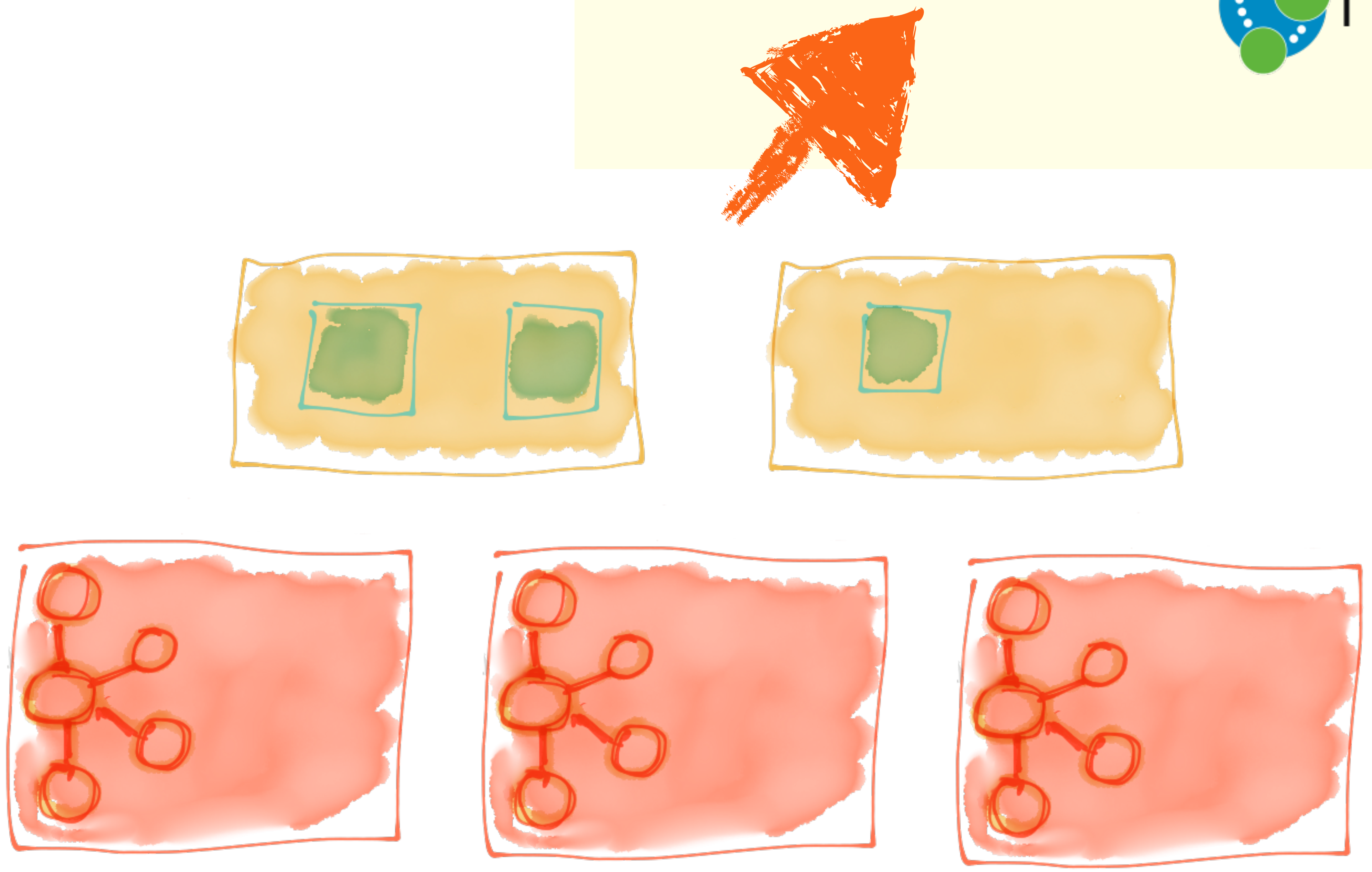


Kafka Connect

Kafka Brokers

Streaming Integration with Kafka Connect

Sinks

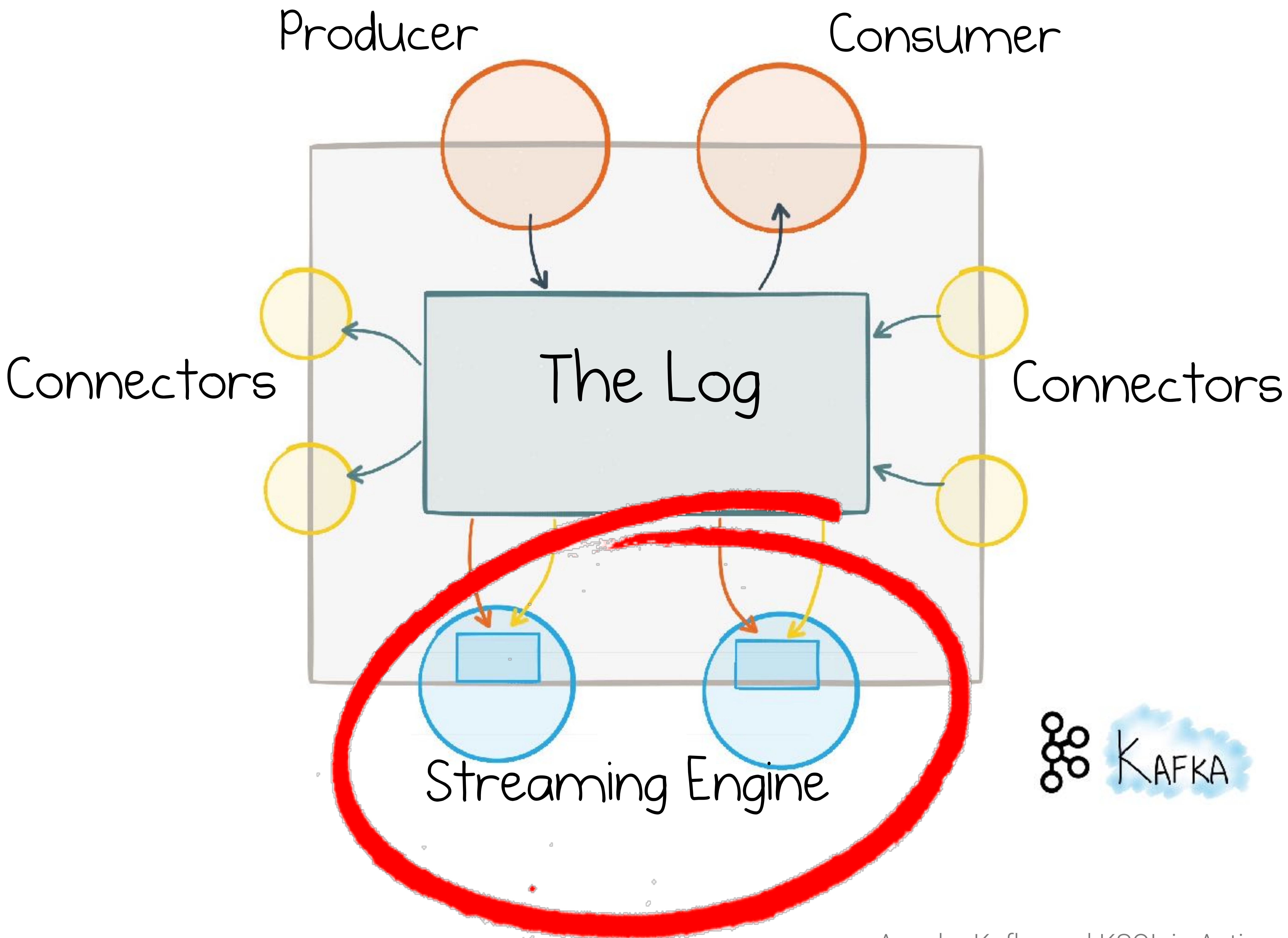


Kafka Connect

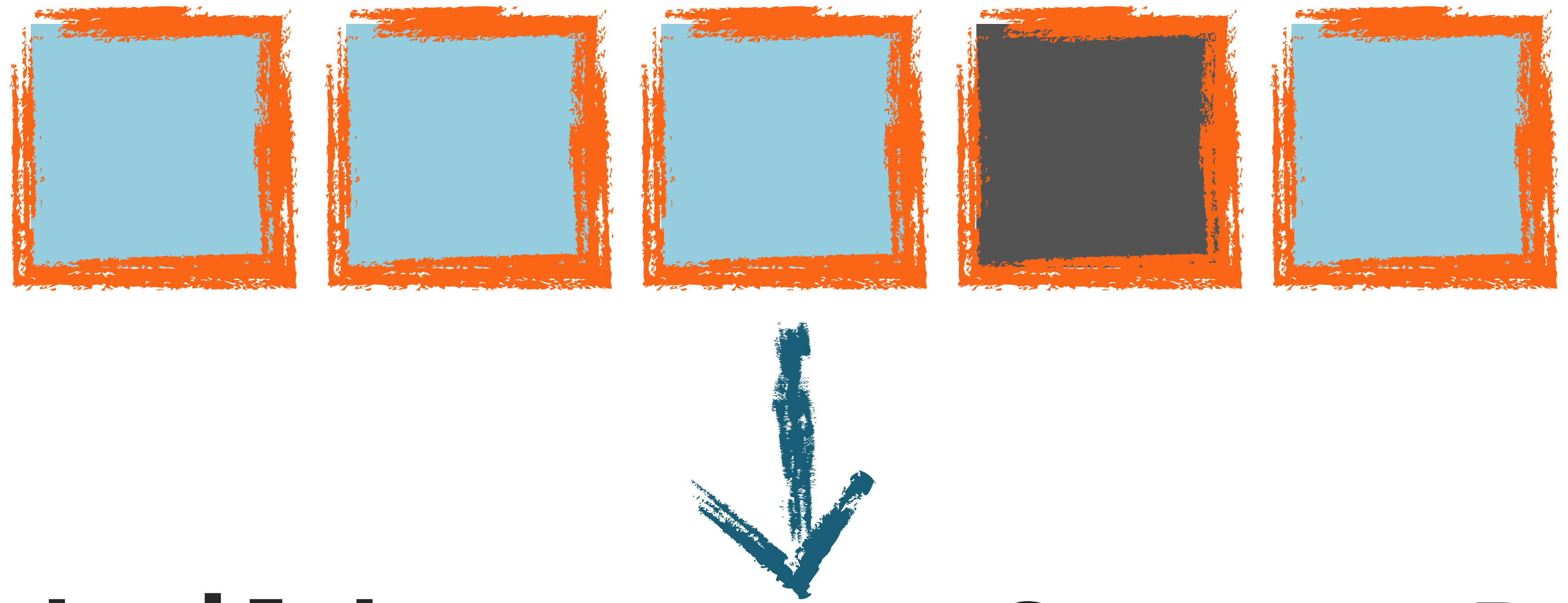
Kafka Brokers

Streaming Integration with Kafka Connect

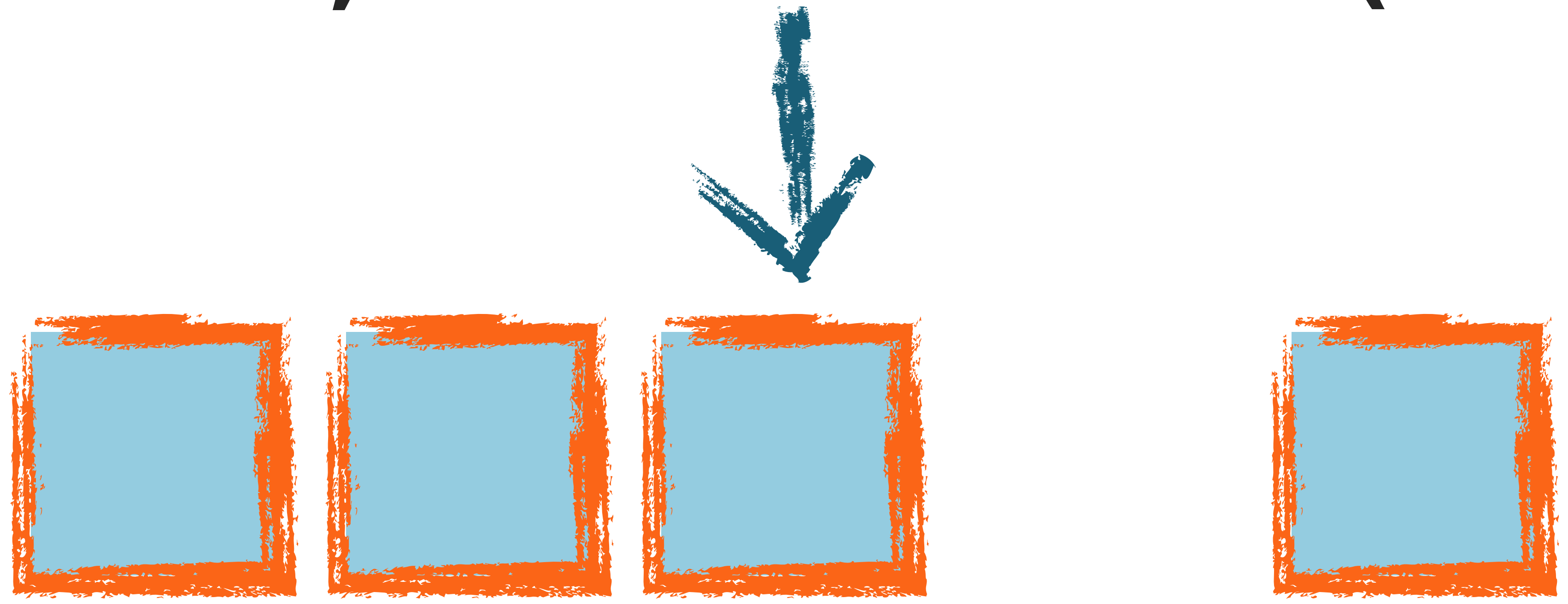
Stream Processing in Kafka



Kafka Streams API

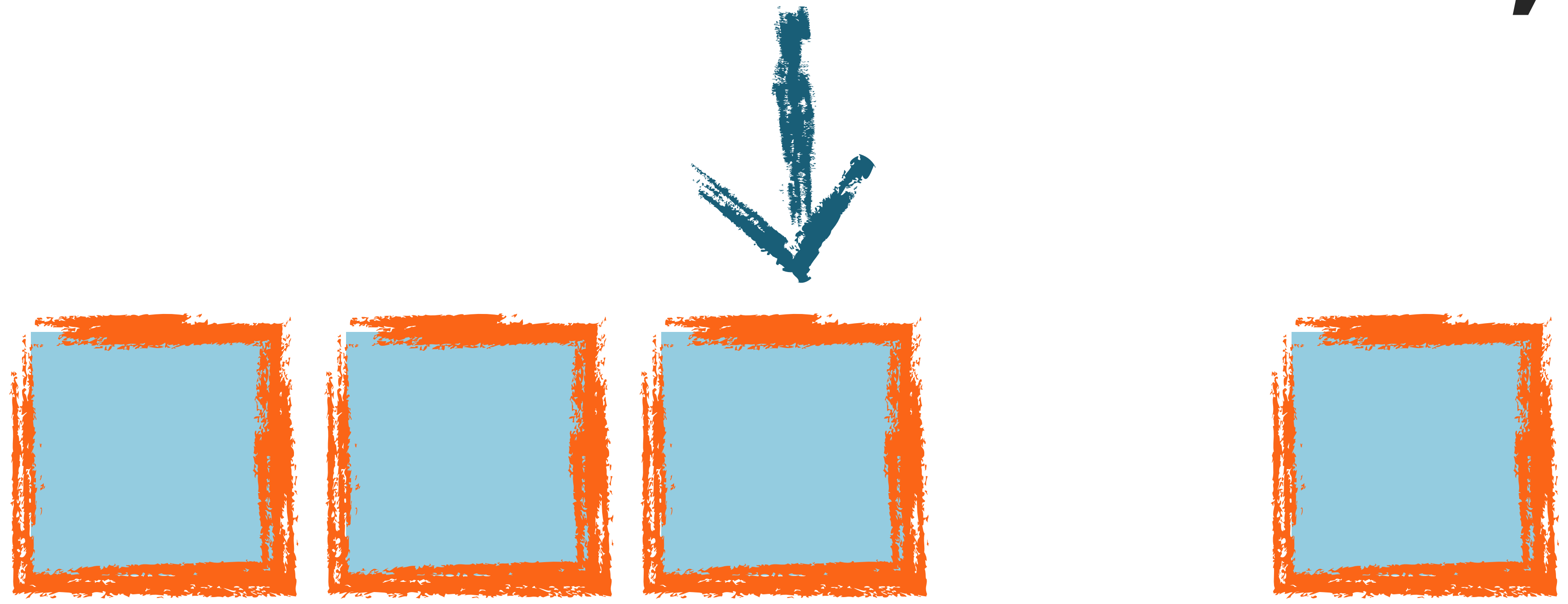


```
final StreamsBuilder builder = new StreamsBuilder()
    .stream("orders", Consumed.with(stringSerde, ordersSerde))
    .filter( (key, order) -> order.getStatus().equals("COMPLETE") )
    .to("complete_orders", Produced.with(stringSerde, ordersSerde));
```

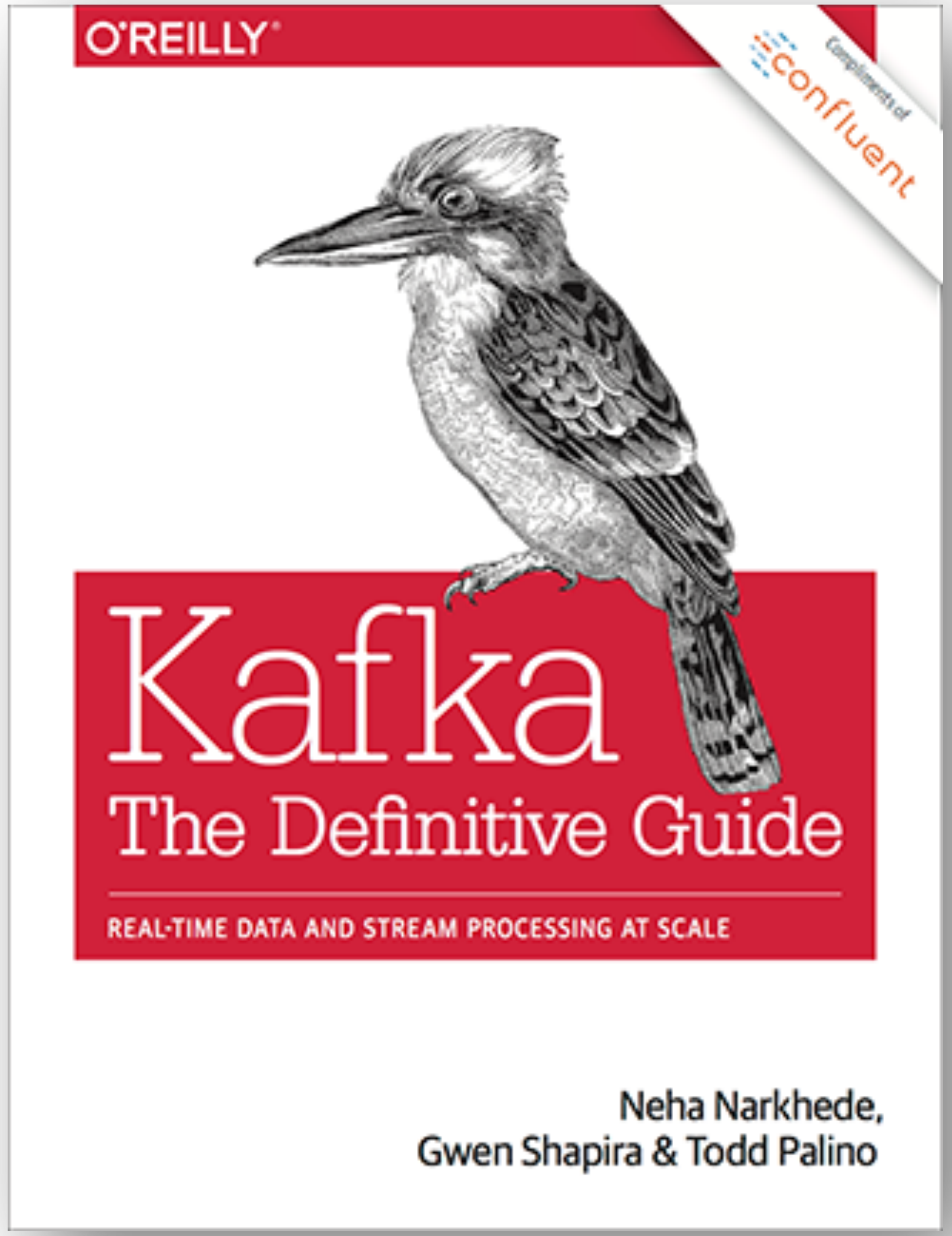


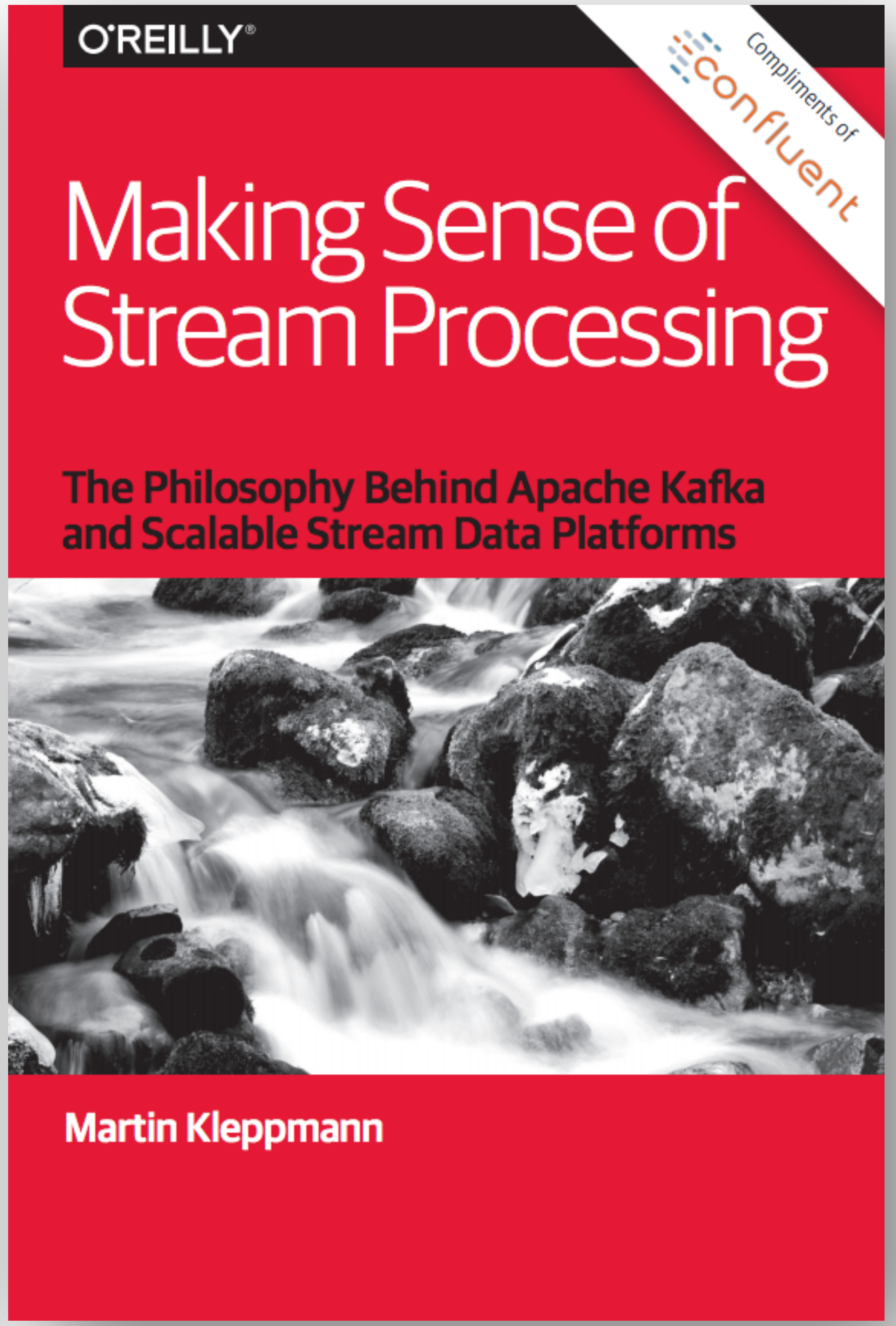
Stream Processing with KSQL

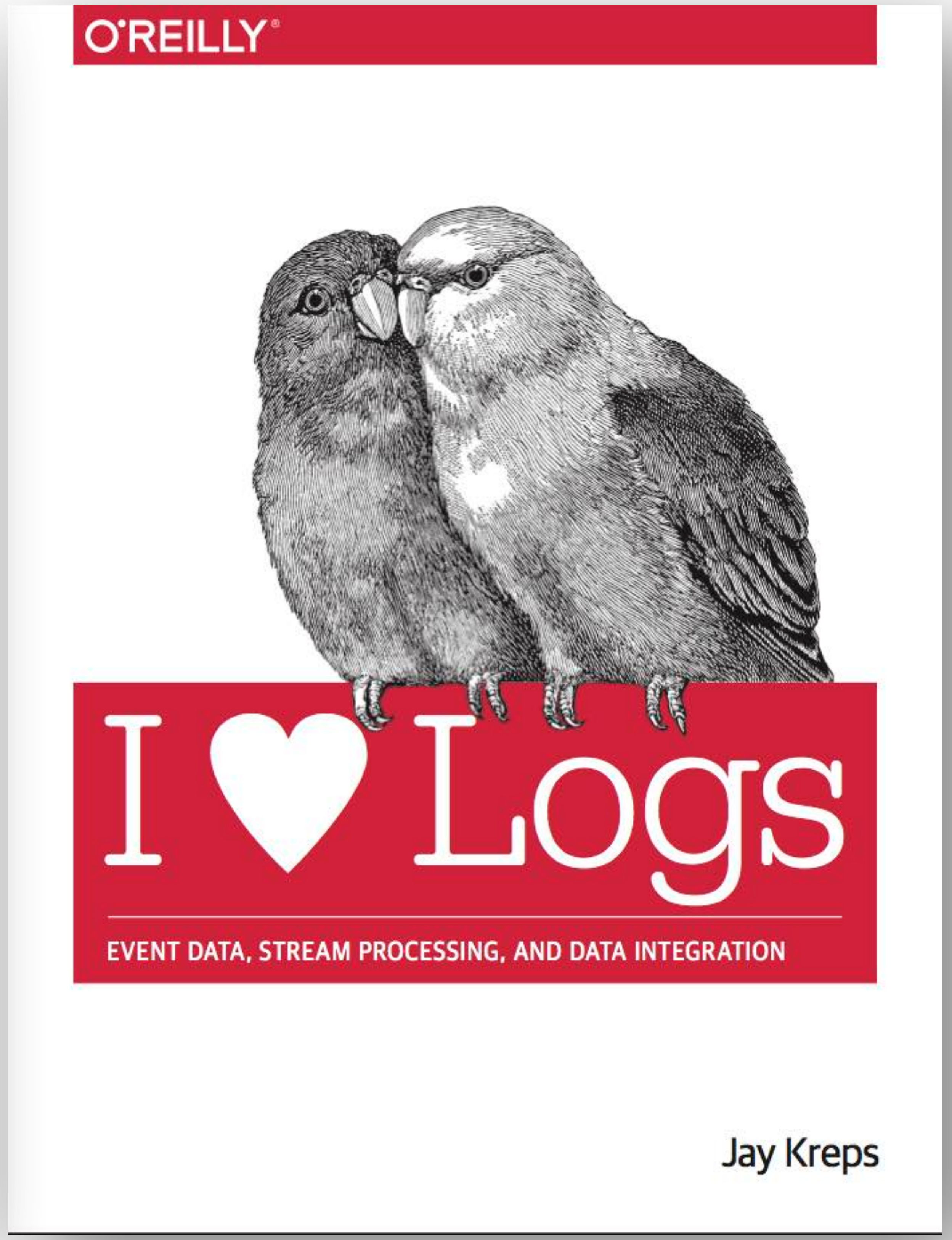
```
CREATE STREAM completedOrders AS
SELECT *
FROM orders
WHERE status='COMPLETE';
```

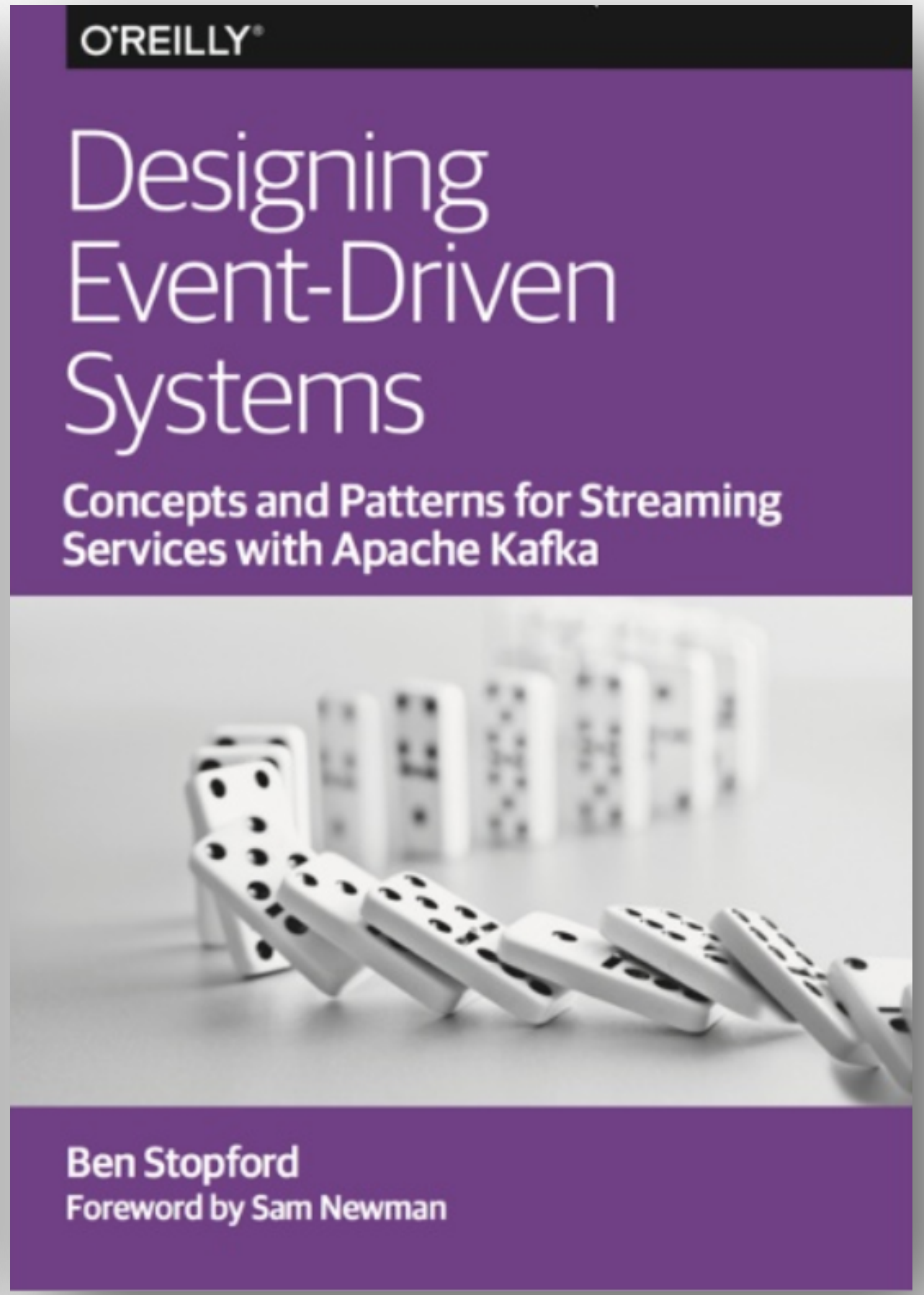


http://cnfl.io/book-bundle

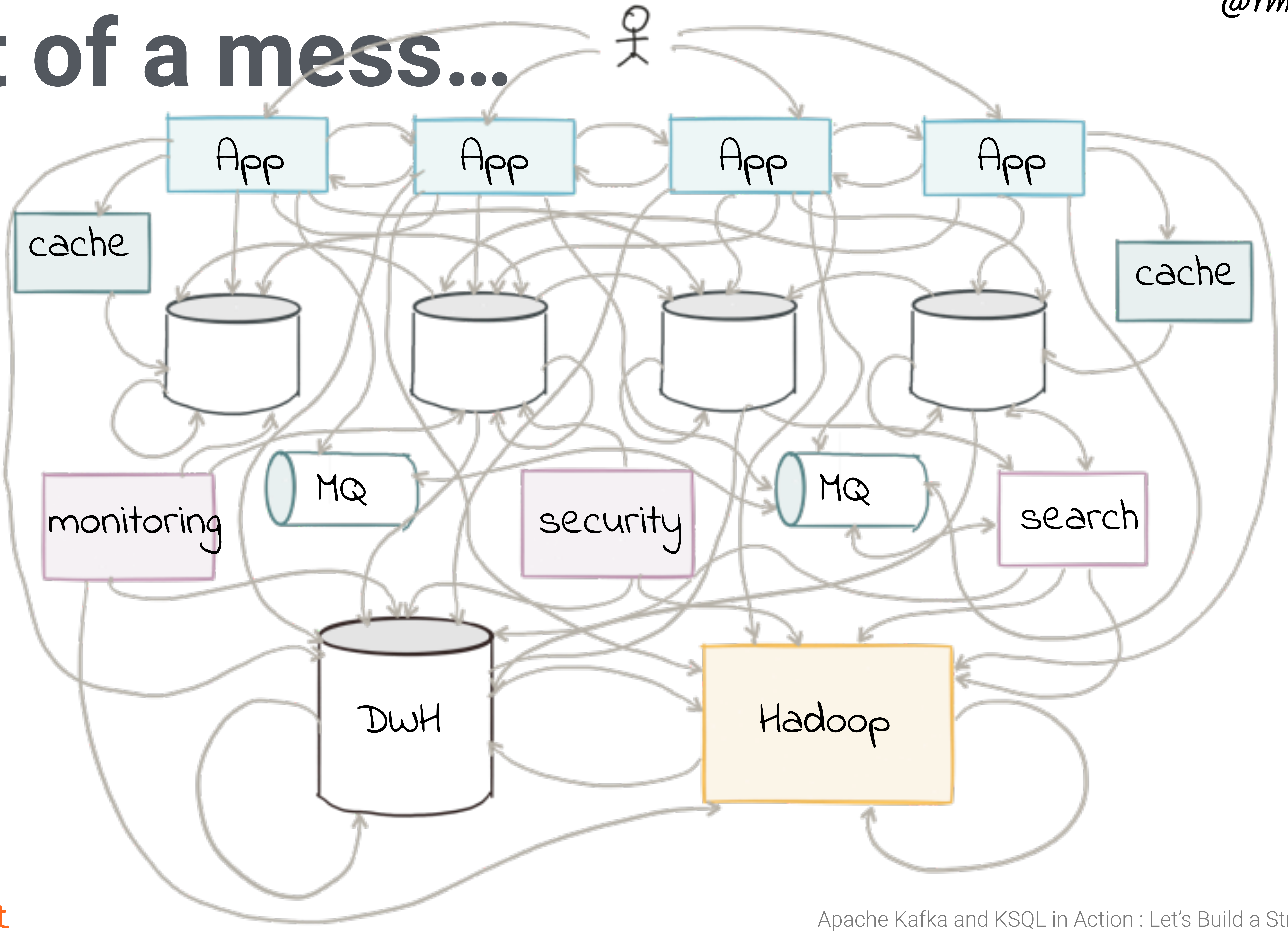




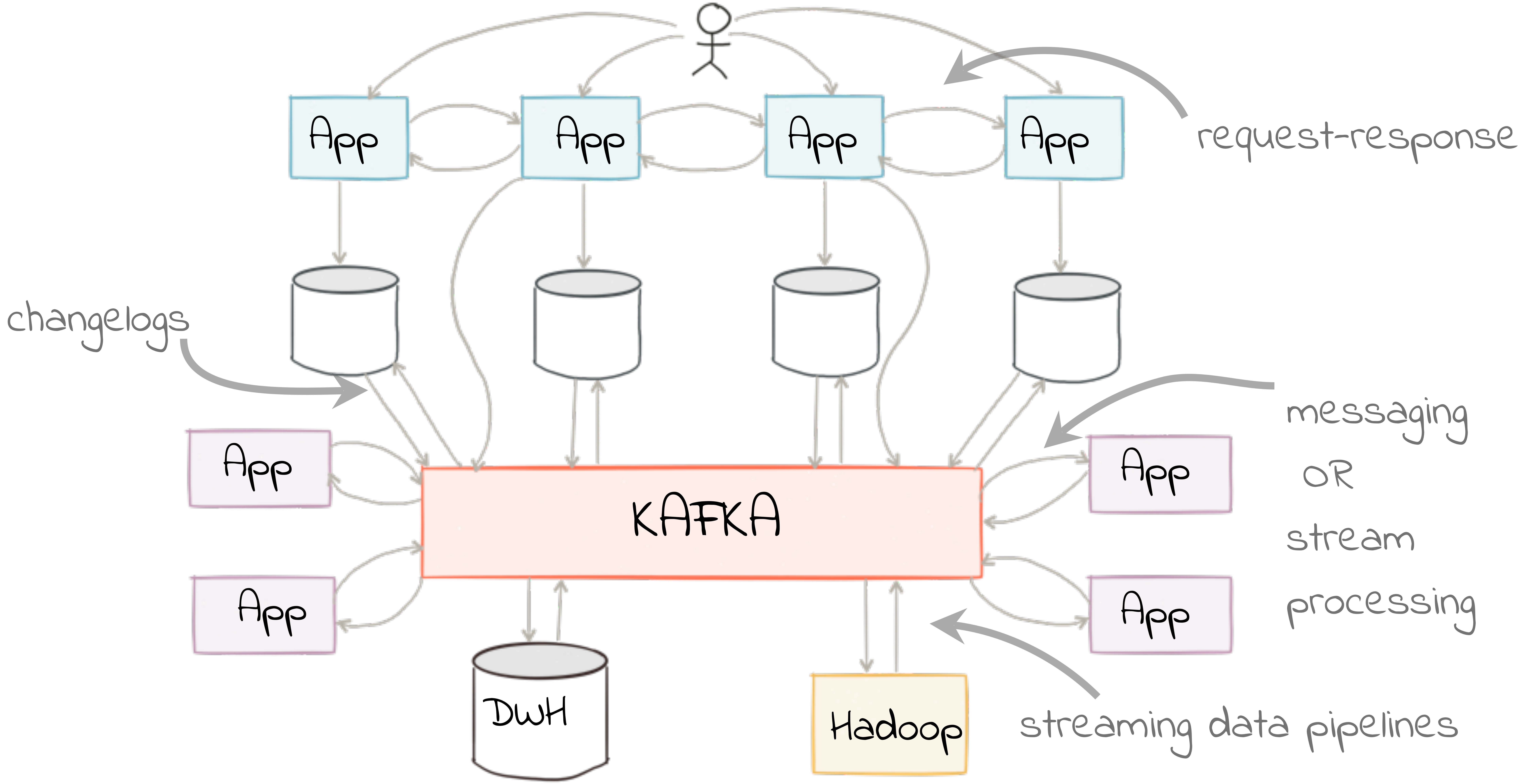




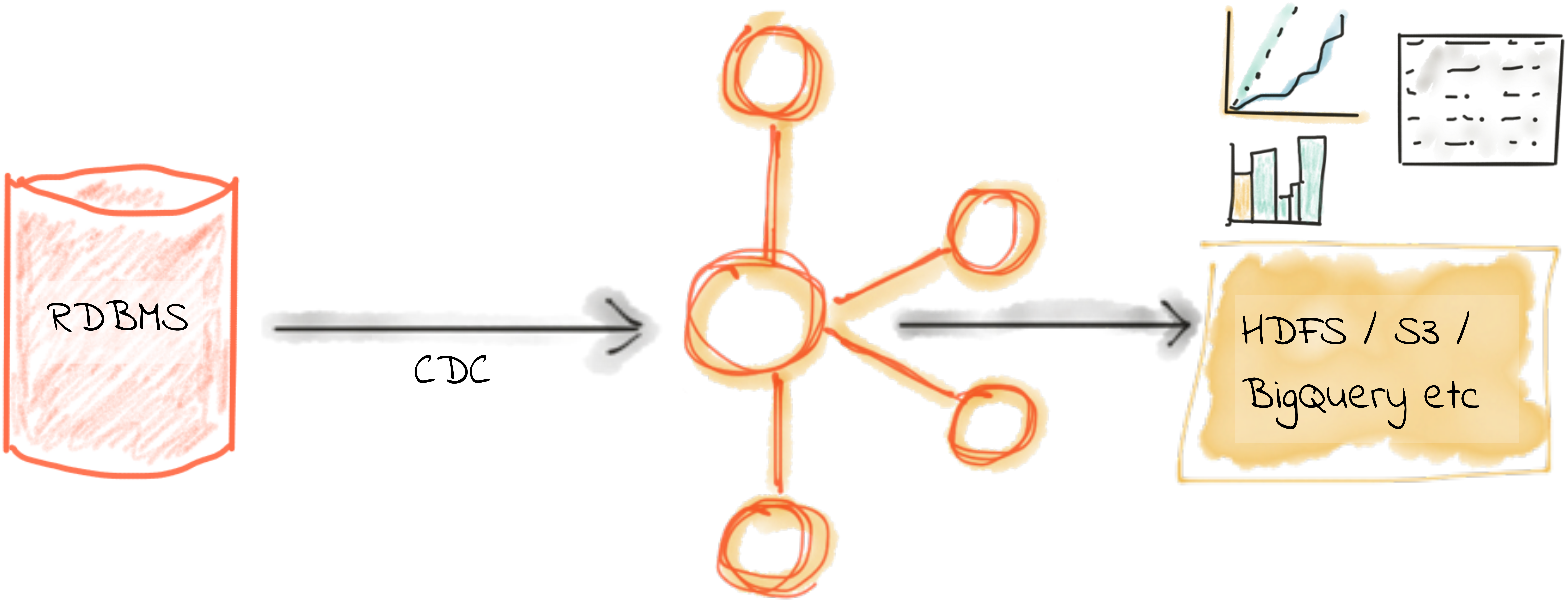
A bit of a mess...



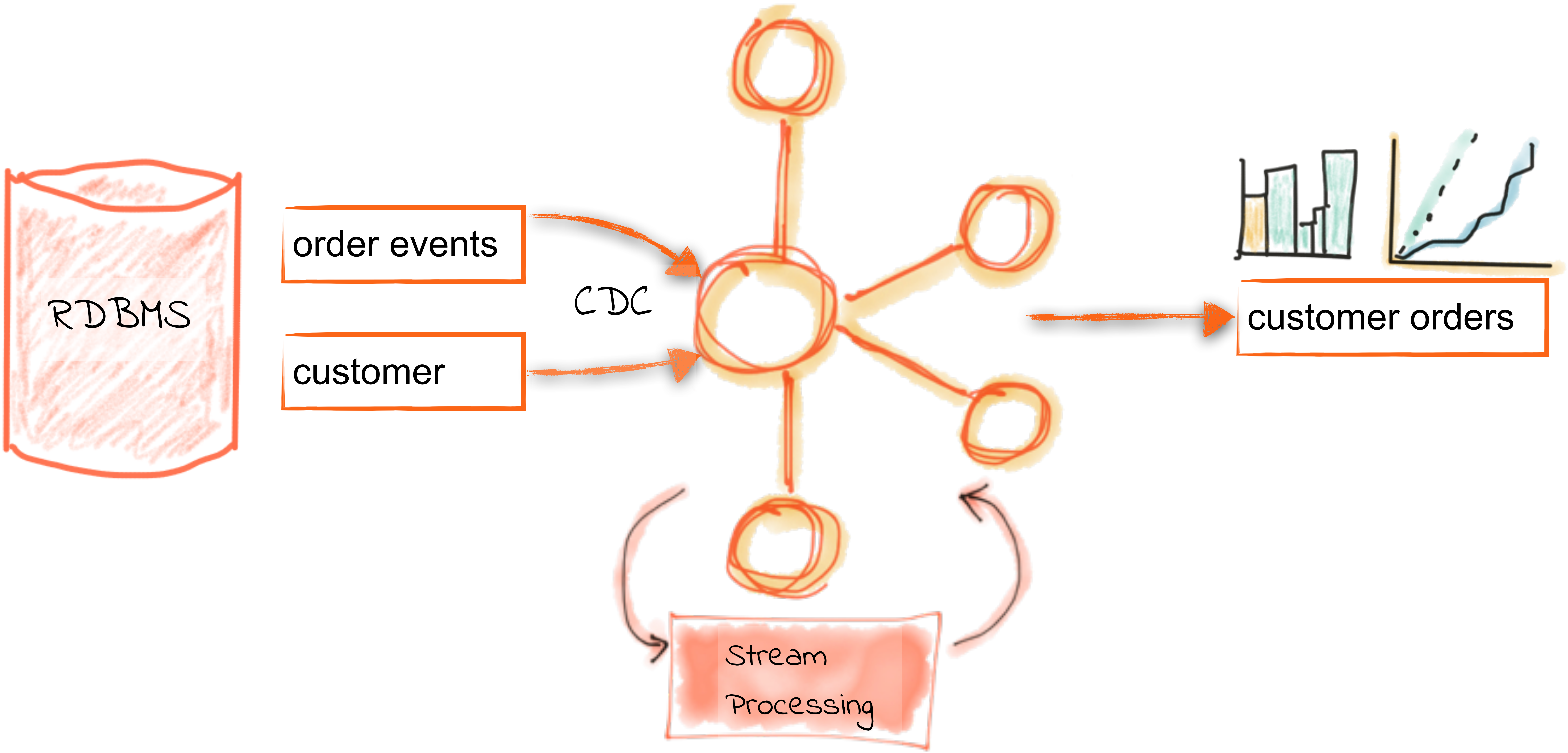
Kafka is a Streaming Platform



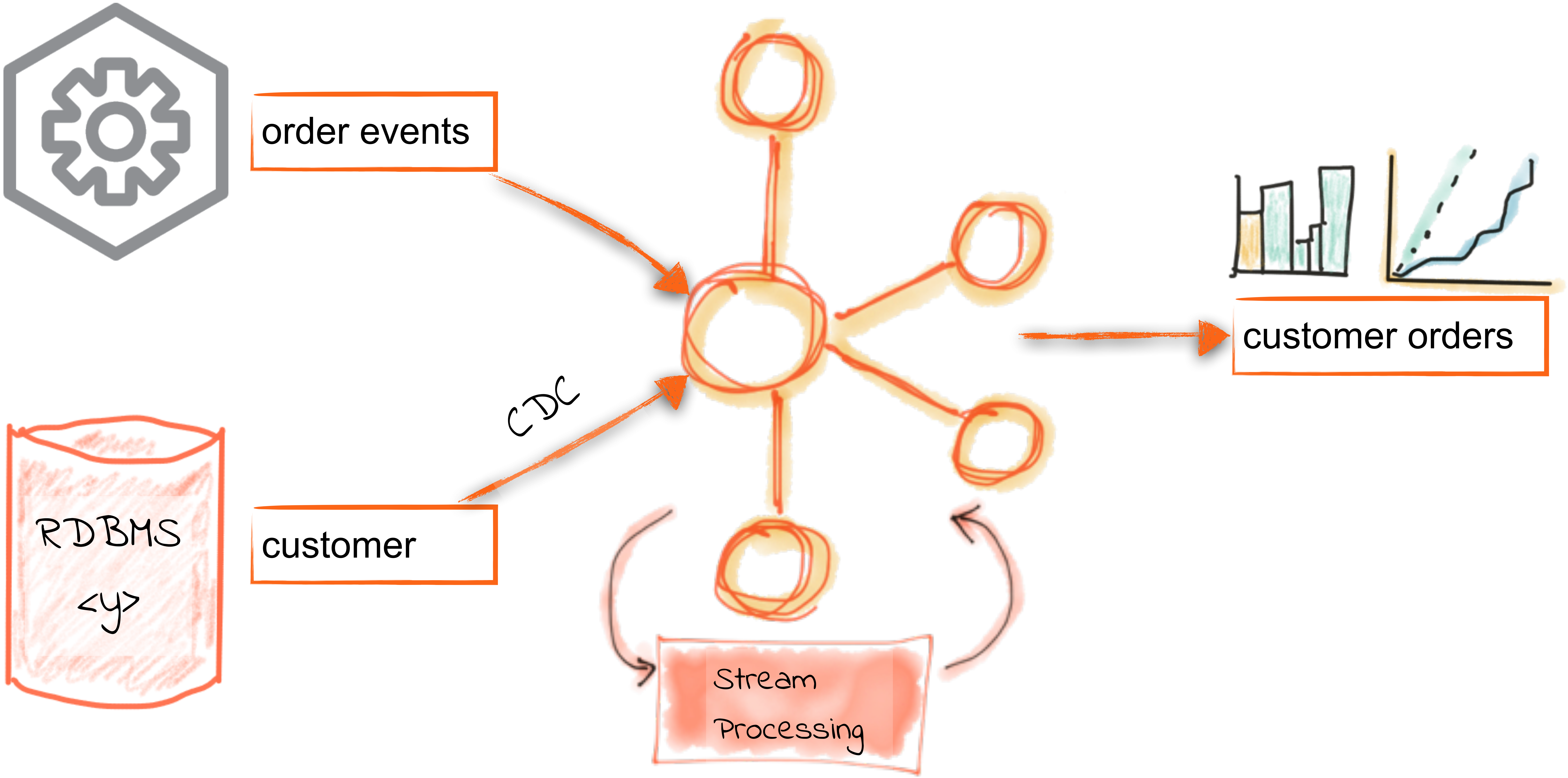
Analytics - Database Offload



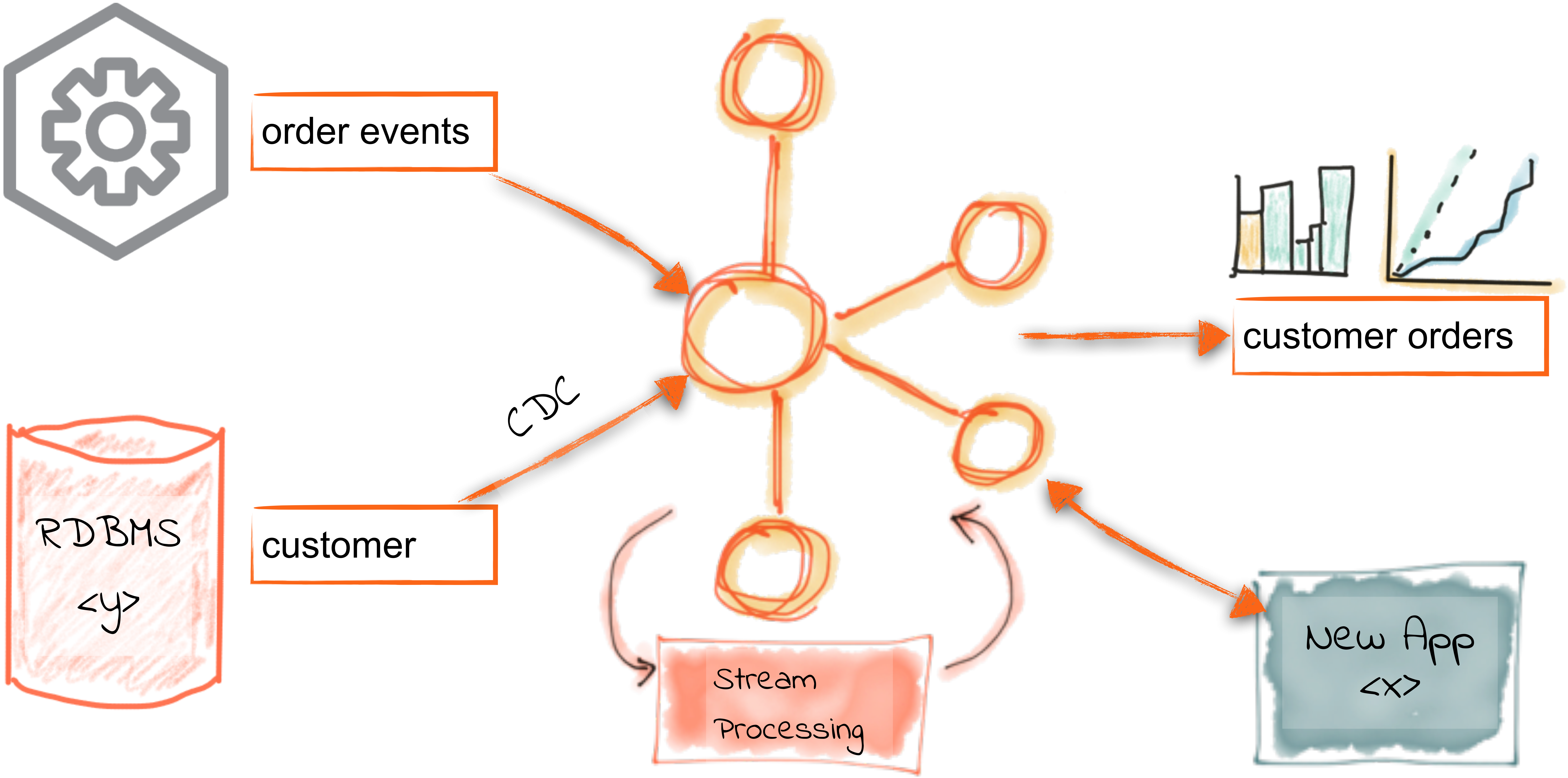
Stream Processing with Apache Kafka and KSQL



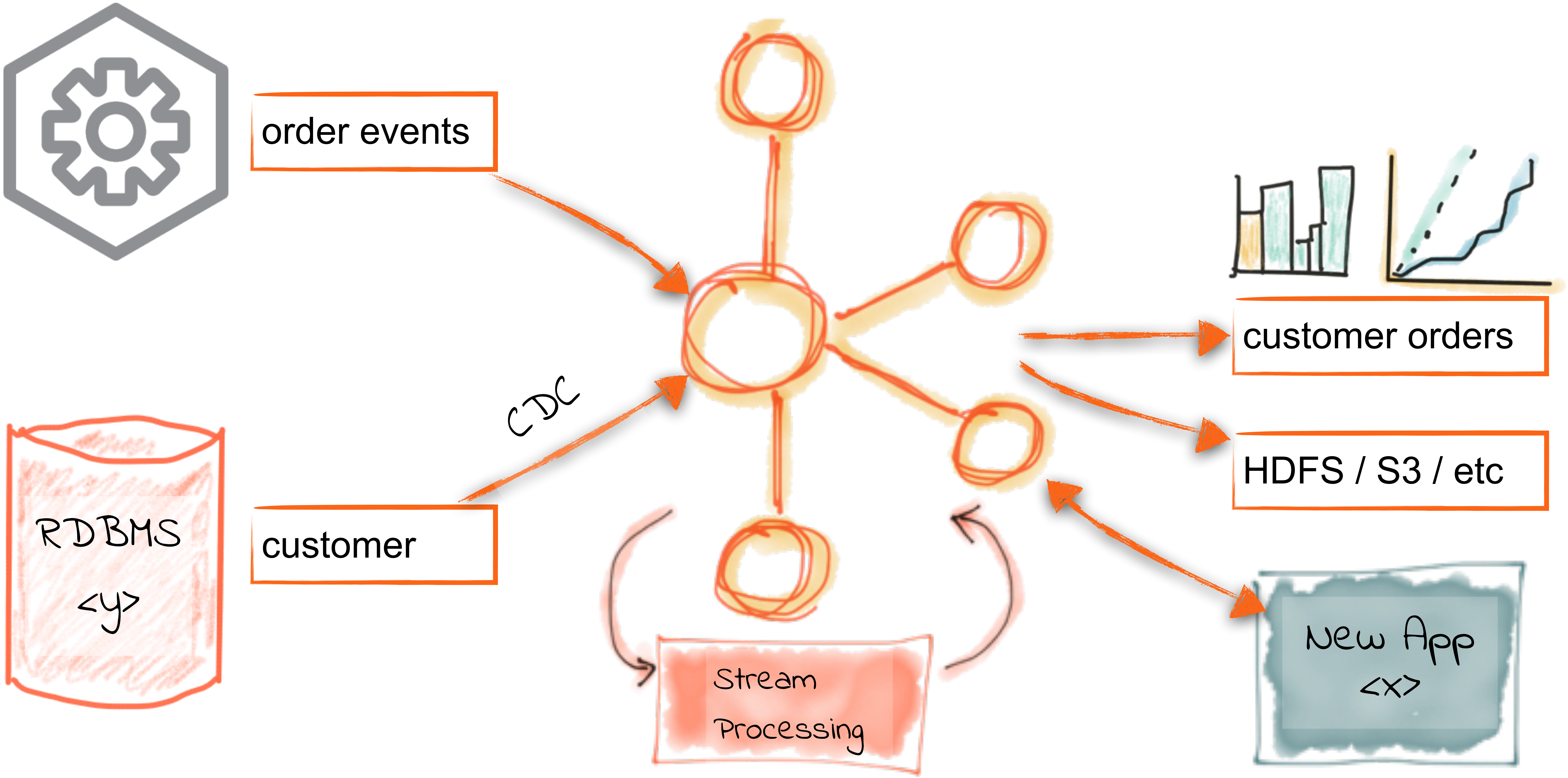
Real-time Event Stream Enrichment



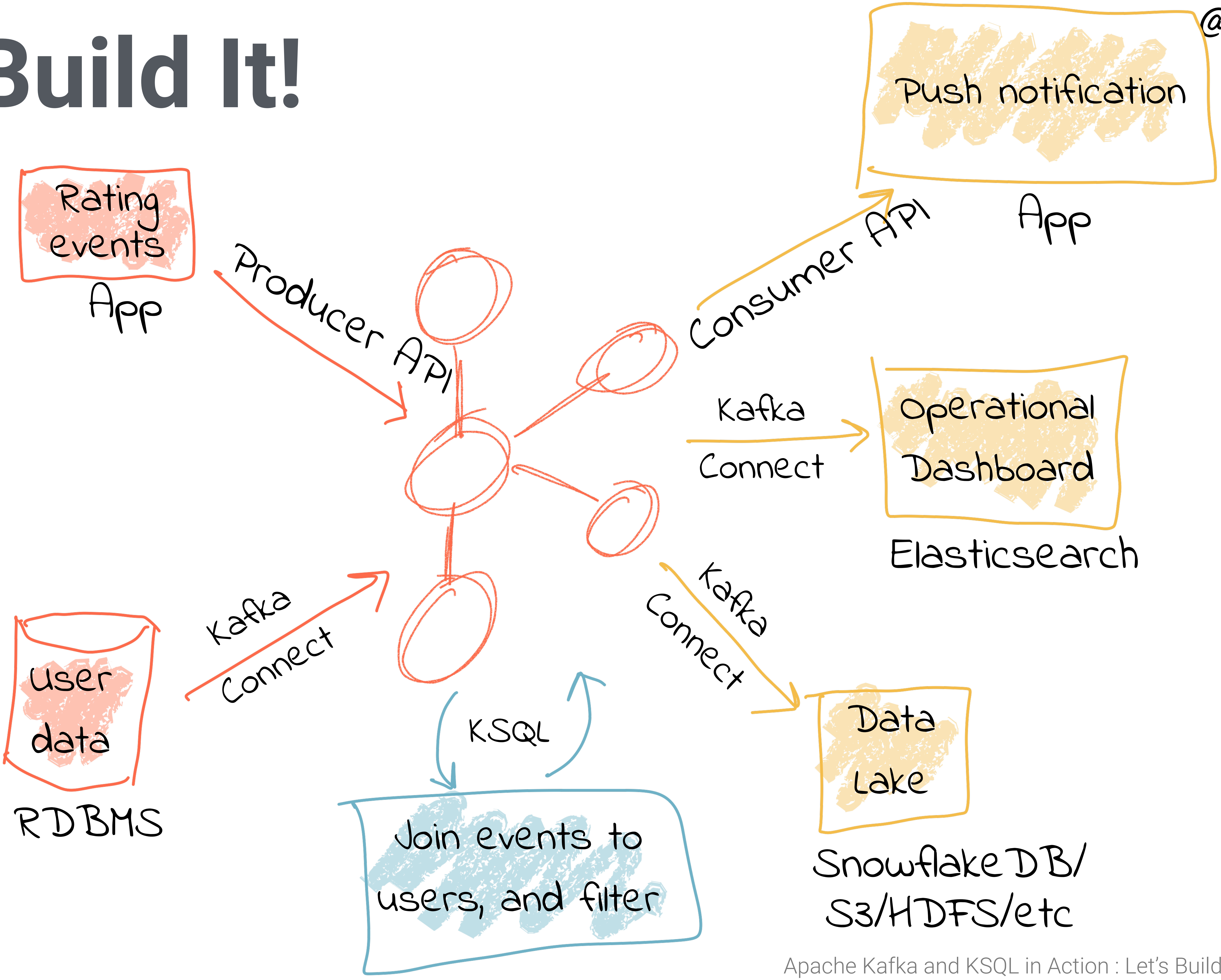
Transform Once, Use Many



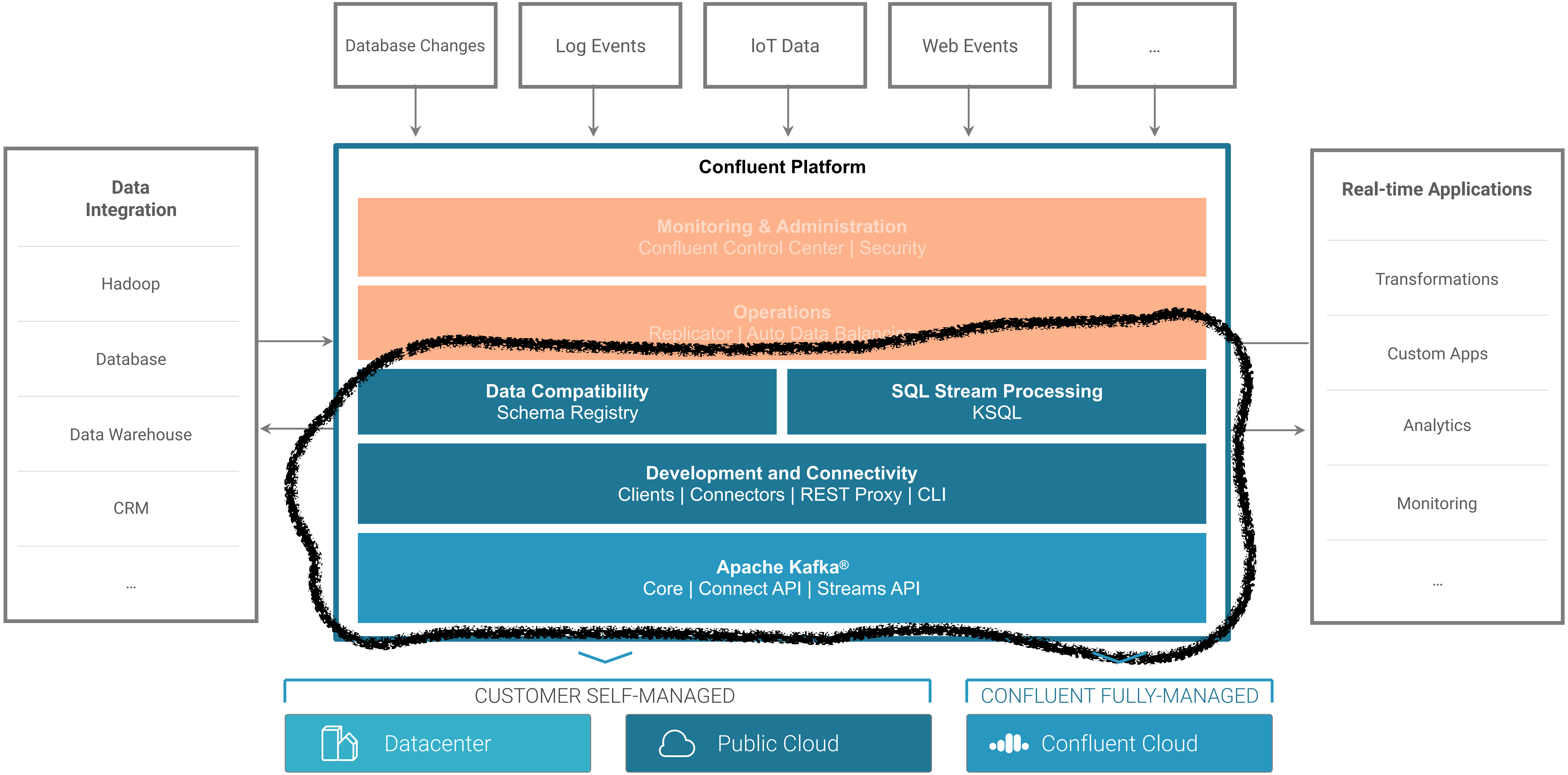
Transform Once, Use Many



Let's Build It!

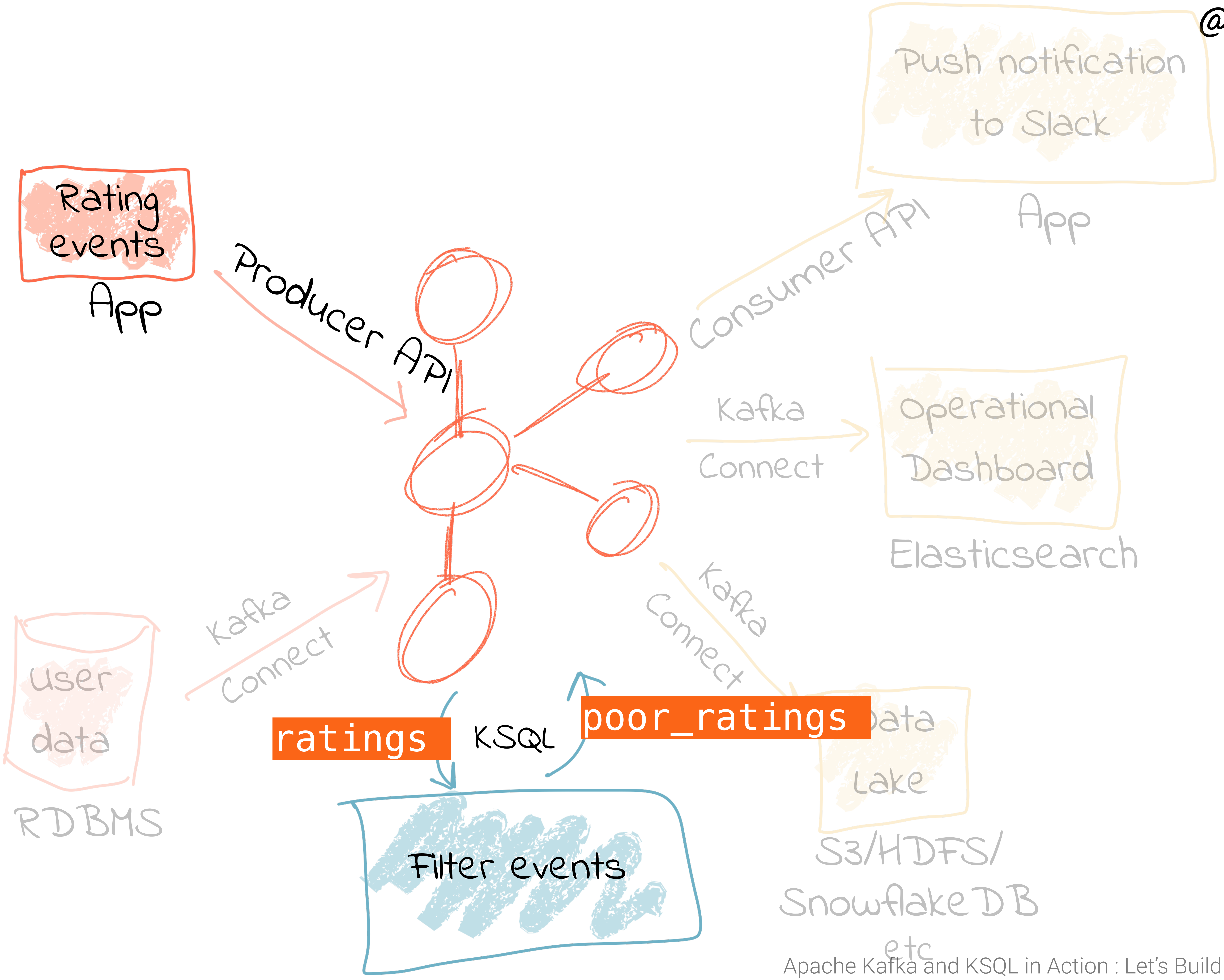


Confluent Community Components
Apache Kafka with a bunch of cool stuff! For free!



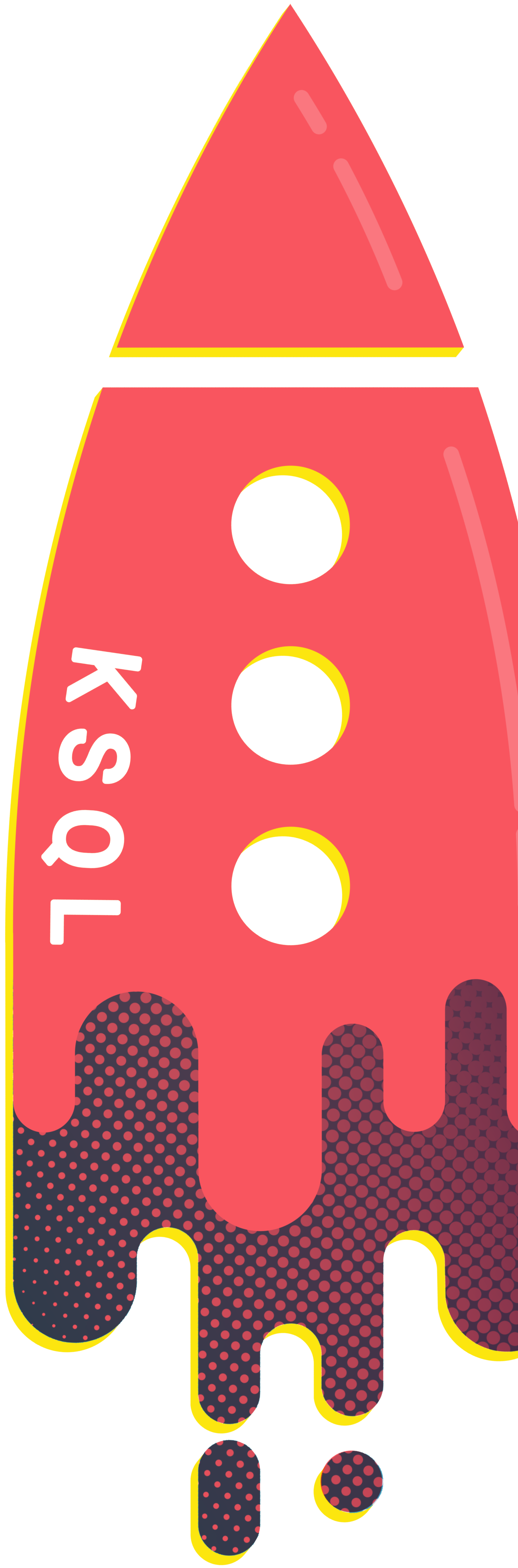
KSQL

@rmoff #NDCOslo



confluent

KSQL
is the
Streaming
SQL Engine
for
Apache Kafka



@rmoff #NDCOslo

Filter messages with KSQL

orders

01, £10.00, COMPLETE

02, £12.33, COMPLETE

04, £5.50, COMPLETE

05, £10.00, PENDING

06, £24.00, COMPLETE

```
CREATE STREAM completedOrders AS
SELECT *
FROM orders
WHERE status='COMPLETE';
```

completedOrders

01, £10.00, COMPLETE

02, £12.33, COMPLETE

04, £5.50, COMPLETE

06, £24.00, COMPLETE

Drop columns with KSQL

customer

```
{"id":1,
"name":"Dana Lidgerton",
"card":"5048370182840140}
```

```
{"id":2,
"name":"Milo Wellsman",
"card":"3557977885537506}
```

```
{"id":3,
"name":"Dolph Cleeton",
"card":"3586303633007251}
```

```
CREATE STREAM customerNoCC AS
SELECT ID, NAME
FROM customer;
```

customerNoCC

```
{"id":1,
"name":"Dana Lidgerton"}
```

```
{"id":2,
"name":"Milo Wellsman"}
```

```
{"id":3,
"name":"Dolph Cleeton"}
```

Stateful aggregation with KSQL

customer

{"id":1,
"name":"Dana Lidgerton",
"country":"UK"}

{"id":2,
"name":"Milo Wellsman",
"country":"UK"}

{"id":3,
"name":"Dolph Cleeton",
"country":"Germany"}

```
CREATE STREAM customersByCountry AS
SELECT country, COUNT(*) AS customerCount
FROM customer WINDOW TUMBLING (SIZE 1 HOUR)
        GROUP BY country;
```

customersByCountry

{"country":"UK",
"customerCount":2}

{"country":"Germany",
"customerCount":1}

KSQL for Anomaly Detection

Identifying 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;
```

KSQL for Data Transformation

Make simple derivations of existing topics from the command line

```
CREATE STREAM pageviews
  WITH (PARTITIONS=4,
        VALUE_FORMAT='AVRO') AS
  SELECT * FROM pageviews_json;
```

KSQL for Streaming ETL

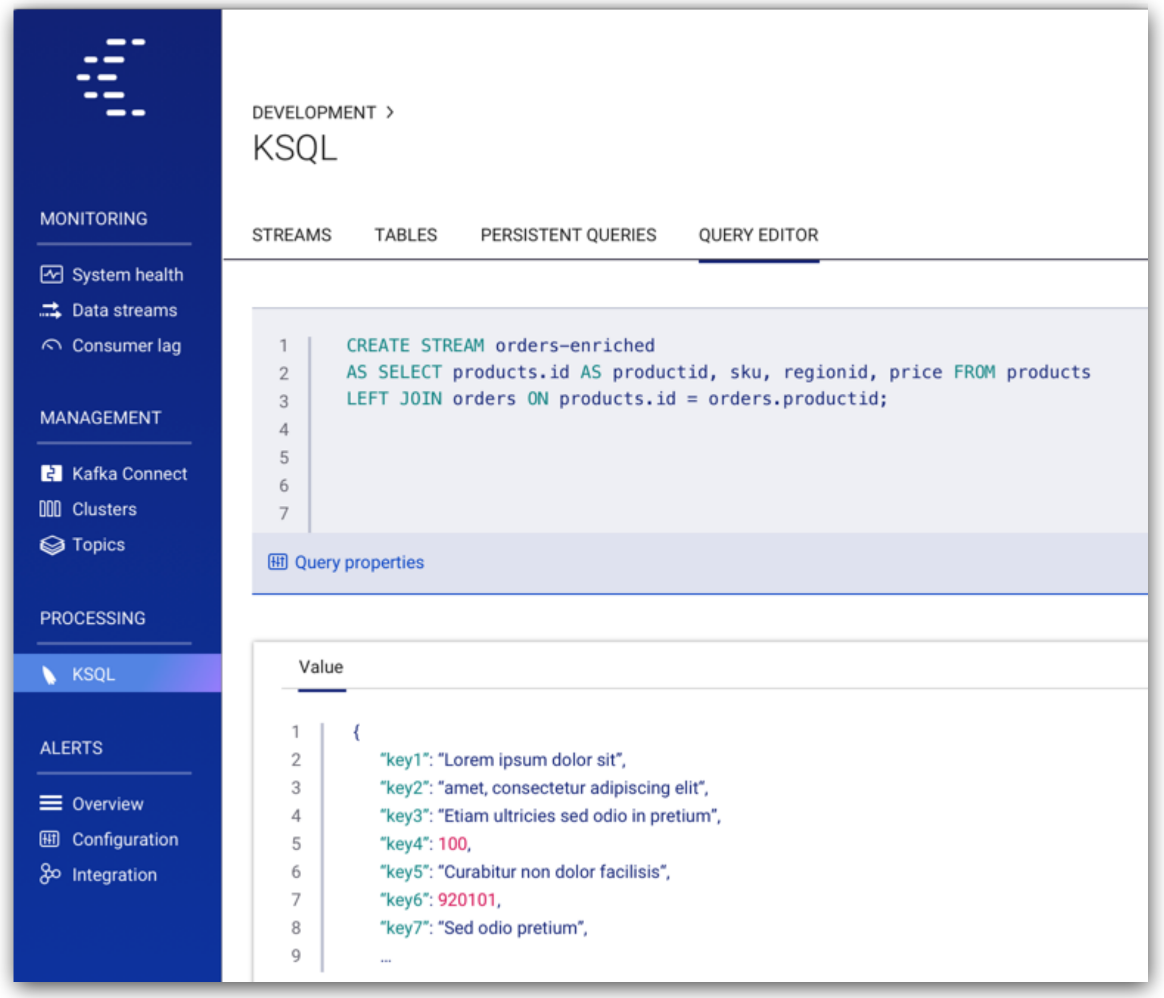
Joining, filtering, and aggregating streams of event data

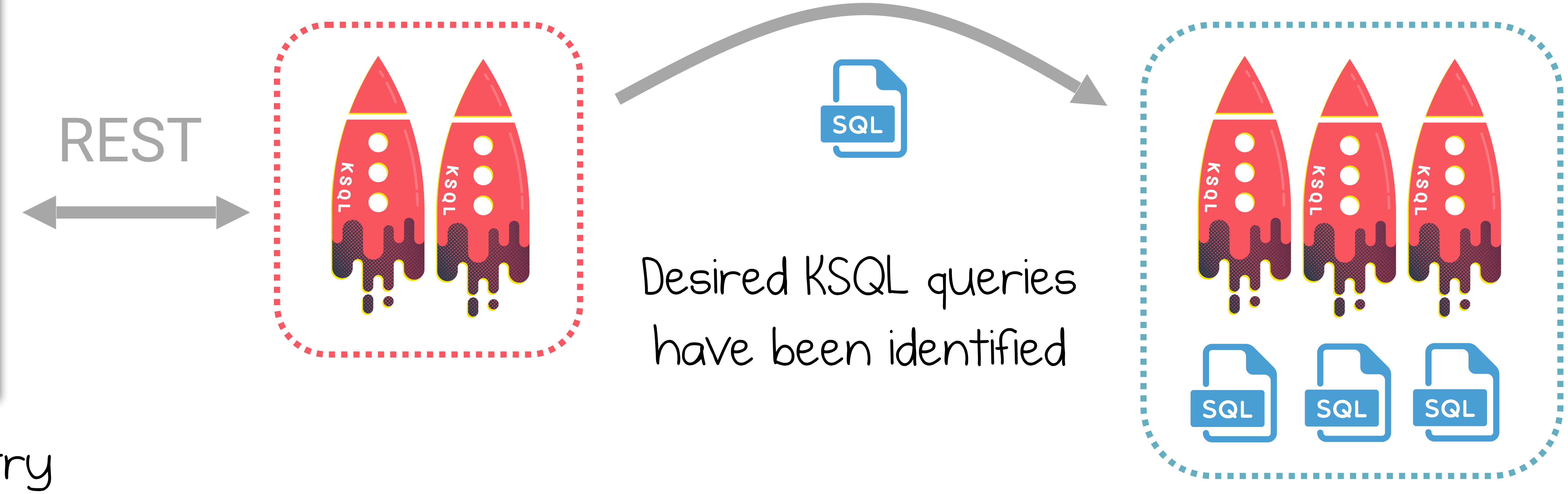
```
CREATE STREAM vip_actions AS
  SELECT userid, page, action
  FROM clickstream c
  LEFT JOIN users u
    ON c.userid = u.user_id
  WHERE u.level = 'Platinum';
```

KSQL in Development and Production

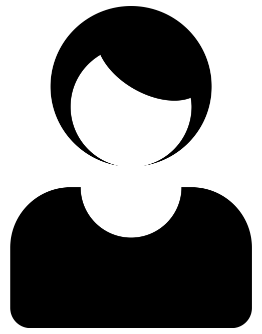
Interactive KSQL
for development and testing

Headless KSQL
for Production



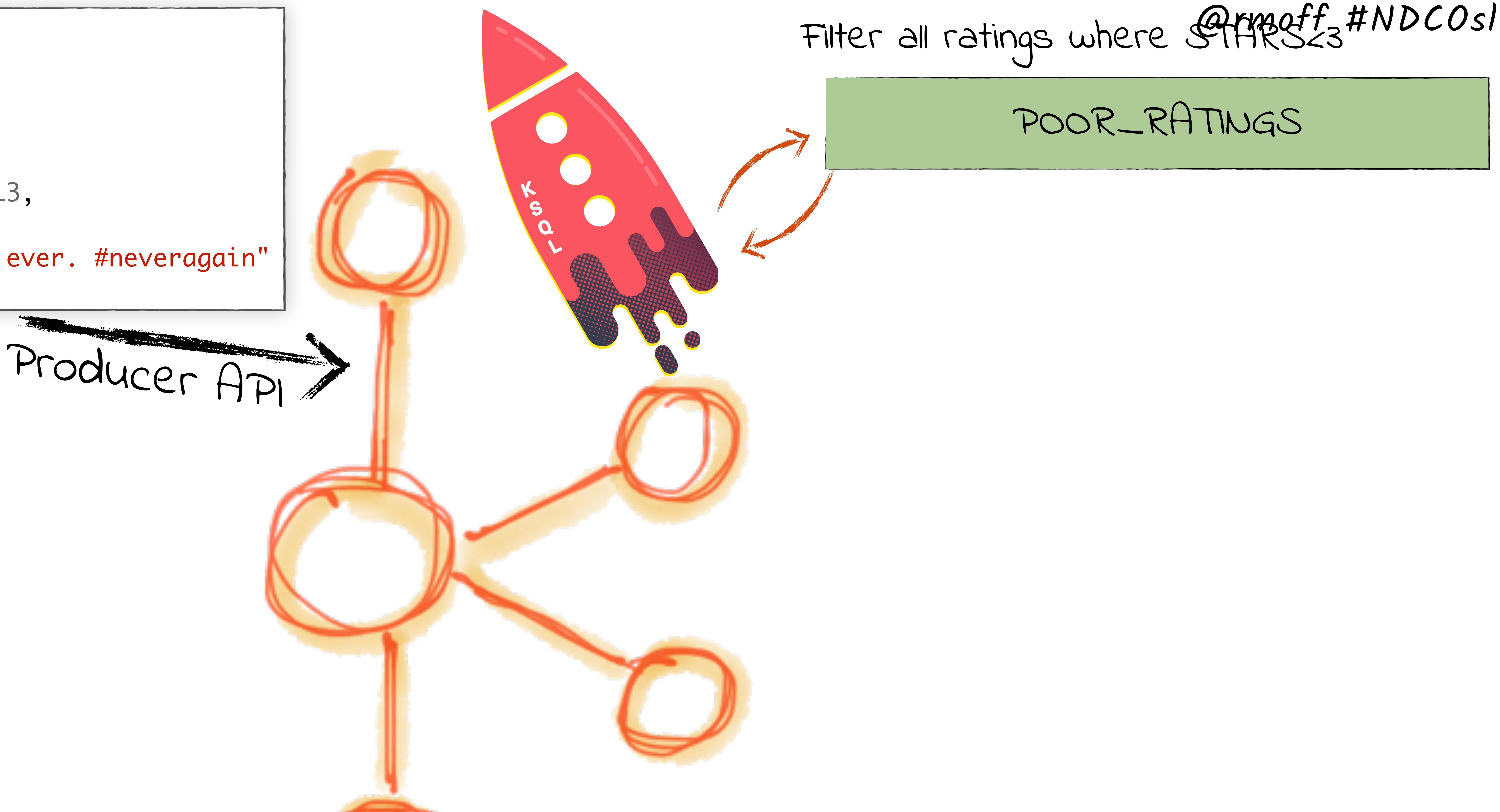


Desired KSQL queries
have been identified



"Hmm, let me try
out this idea..."

```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}
```


```
CREATE STREAM POOR_RATINGS AS
SELECT * FROM ratings WHERE STARS <3
```

`https://cnfl.io/ksql-workshop-prereq`

- **Make sure you allocate Docker >=8GB memory**
  `docker system info | grep Memory`
- **Clone the repo**
- **Pull the git images as instructed in the doc**

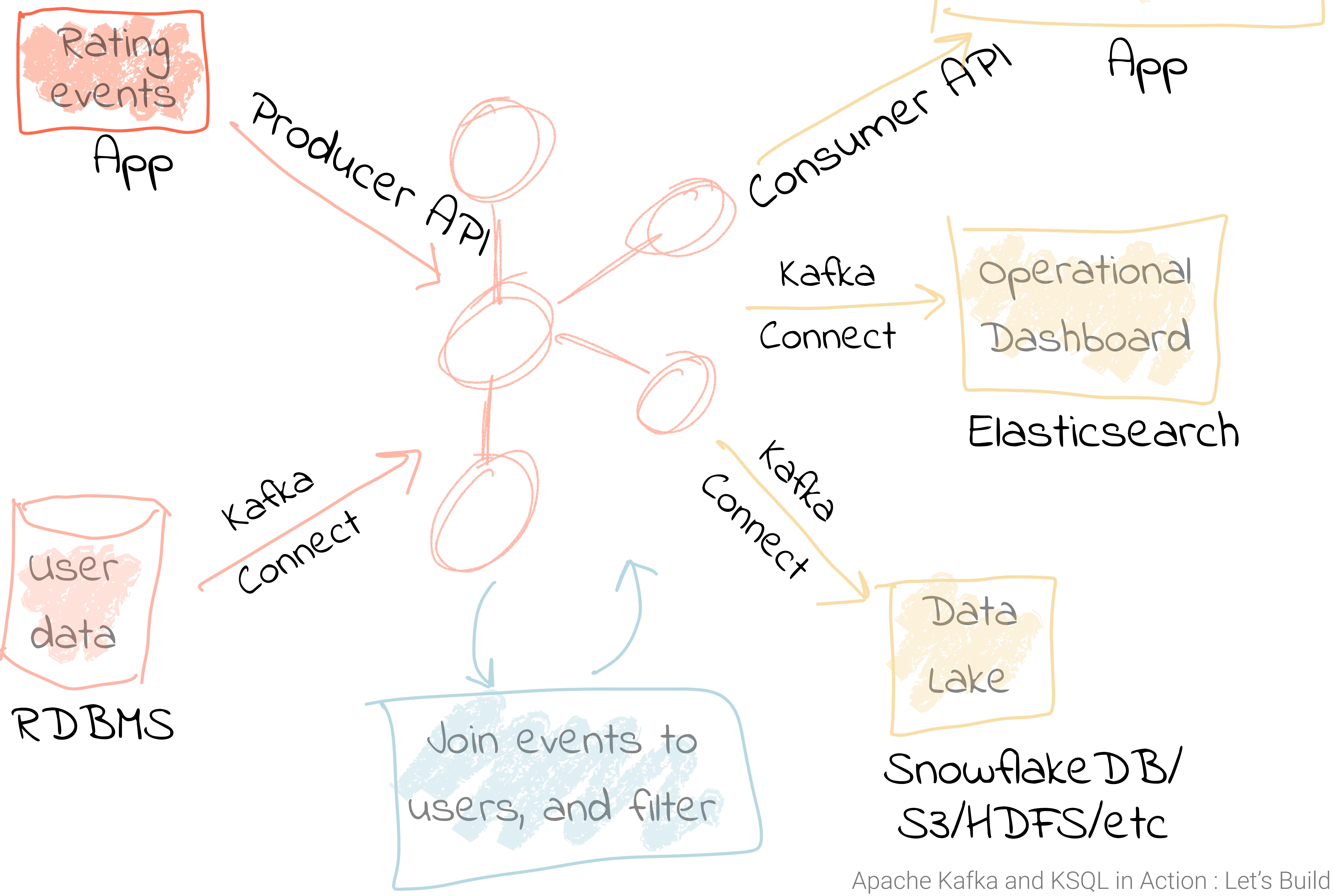
`https://cnfl.io/start-ksql-workshop`

3. Start Confluent Platform

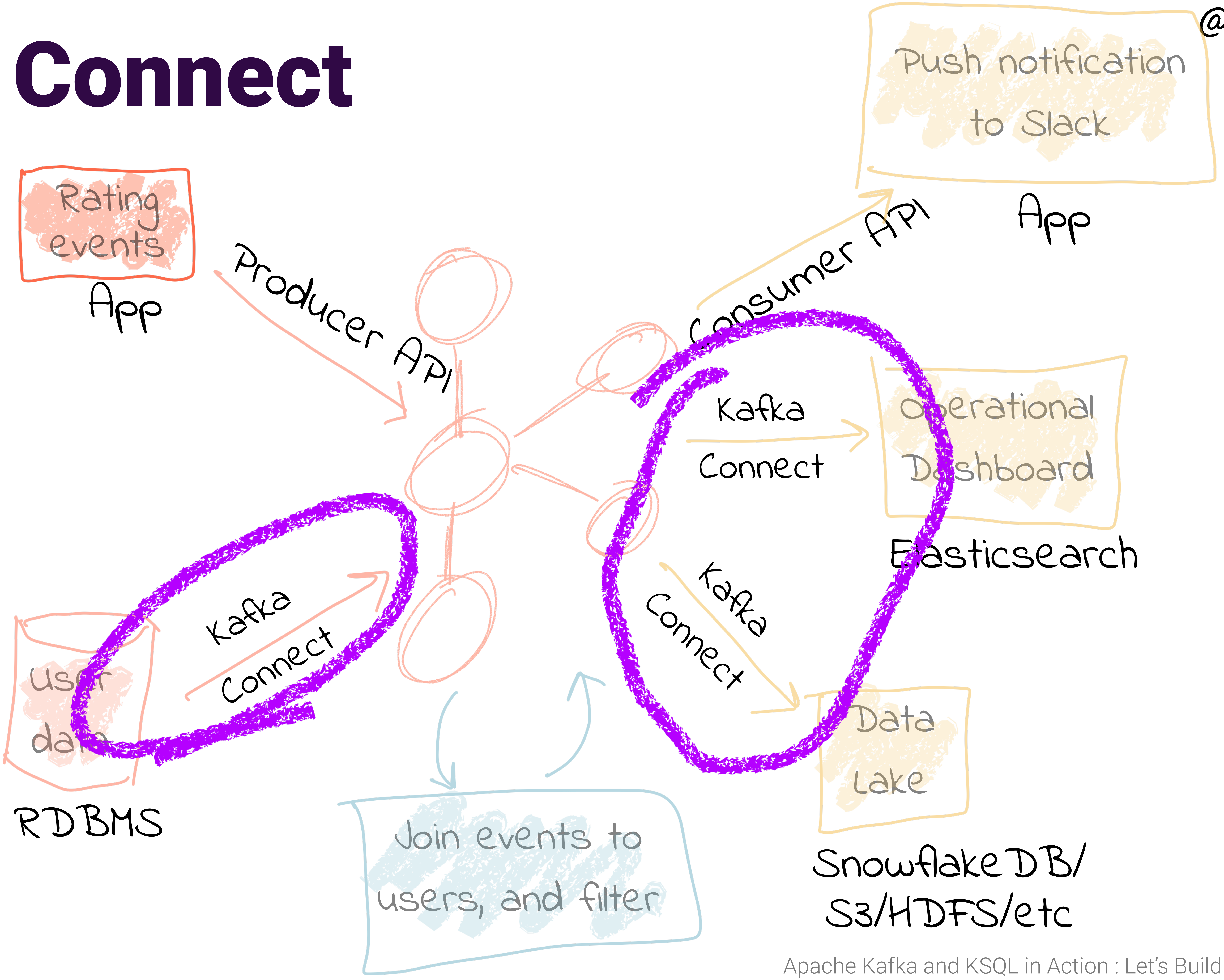
https://cnfl.io/start-ksql-workshop

4. KSQL
5. Querying and filtering streams of data
6. Creating a Kafka topic populated by a filtered stream

Let's Build It!



Kafka Connect



Streaming Integration with Kafka Connect

Kafka Connect

APACHE kafka
A distributed streaming platform

Reliable and scalable integration of Kafka with other systems – no coding required.

- ✓ 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, included in Confluent Platform

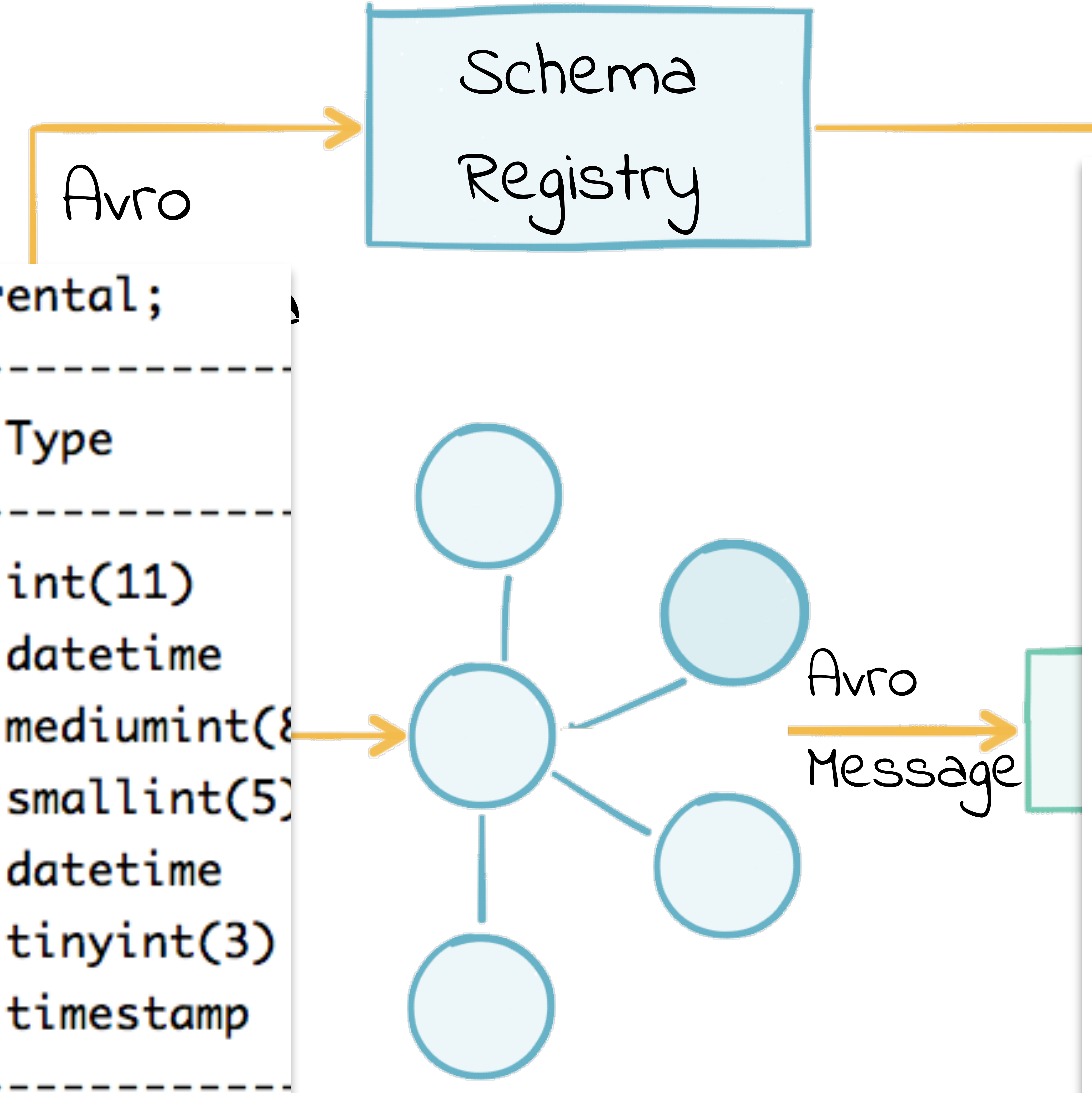
```
{
"connector.class": "io.confluent.connect.jdbc.JdbcSourceConnector",
"connection.url": "jdbc:mysql://localhost:3306/demo?user=rmoff&password=foo",
"table.whitelist": "sales,orders,customers"
}
```

https://docs.confluent.io/current/connect/

Kafka Connect + Schema Registry = WIN

Kafka Connect + Schema Registry = WIN

```
mysql> describe rental;
+--------------+------------
| Field        | Type
+--------------+------------
| rental_id    | int(11)
| rental_date  | datetime
| inventory_id | mediumint(8
| customer_id  | smallint(5)
| return_date  | datetime
| staff_id     | tinyint(3)
| last_update  | timestamp
+--------------+------------
```

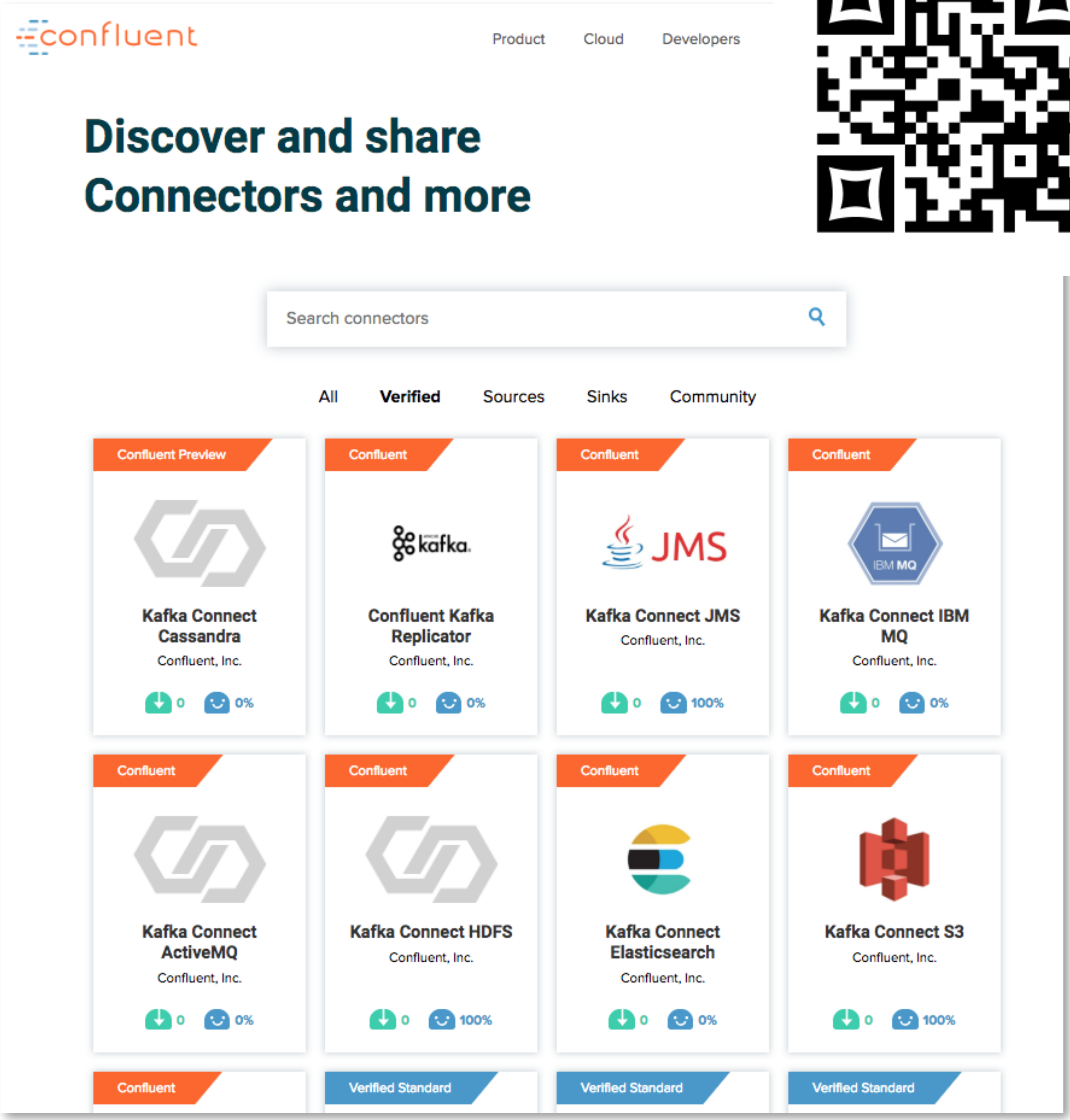



```
"mappings": {
  "type.name=kafka-connect": {
    "dynamic_templates": [],
    "properties": {
      "customer_id": {
        "type": "integer"
      },
      "inventory_id": {
        "type": "long"
      },
      "last_update": {
        "type": "date"
      },
      "rental_date": {
        "type": "date"
      },
      "rental_id": {
        "type": "integer"
      },
      "return_date": {
        "type": "date"
      },
```

Confluent Hub

@

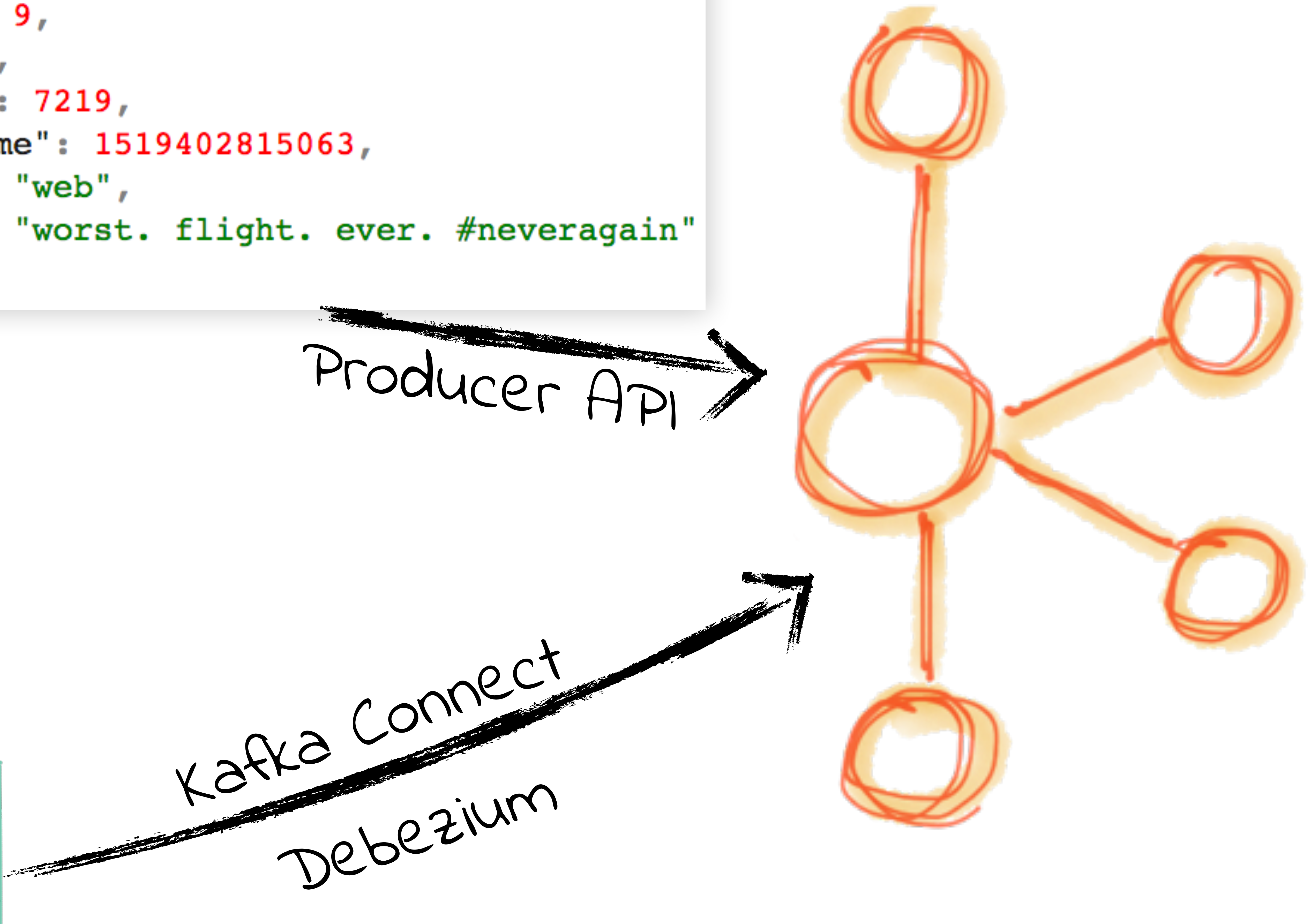
- One-stop place to discover and download :
  - Connectors
  - Transformations
  - Converters



hub.confluent.io

Demo Time!

```
{
  "rating_id": 2133,
  "user_id": 9,
  "stars": 1,
  "route_id": 7219,
  "rating_time": 1519402815063,
  "channel": "web",
  "message": "worst. flight. ever. #neveragain"
}
```




```
{
  "uid": 9,
  "name": "Neha",
  "locale": "en_US",
  "address_city": "Palo Alto",
  "elite": "P"
}
```


Do you think that's a table
you are querying?

The Stream/Table Duality

Stream

Time

Account ID	Amount
12345	+ €50
12345	+ €25
12345	-€60

Table

Account ID	Balance
12345	€50

Account ID	Balance
12345	€75

Account ID	Balance
12345	€15

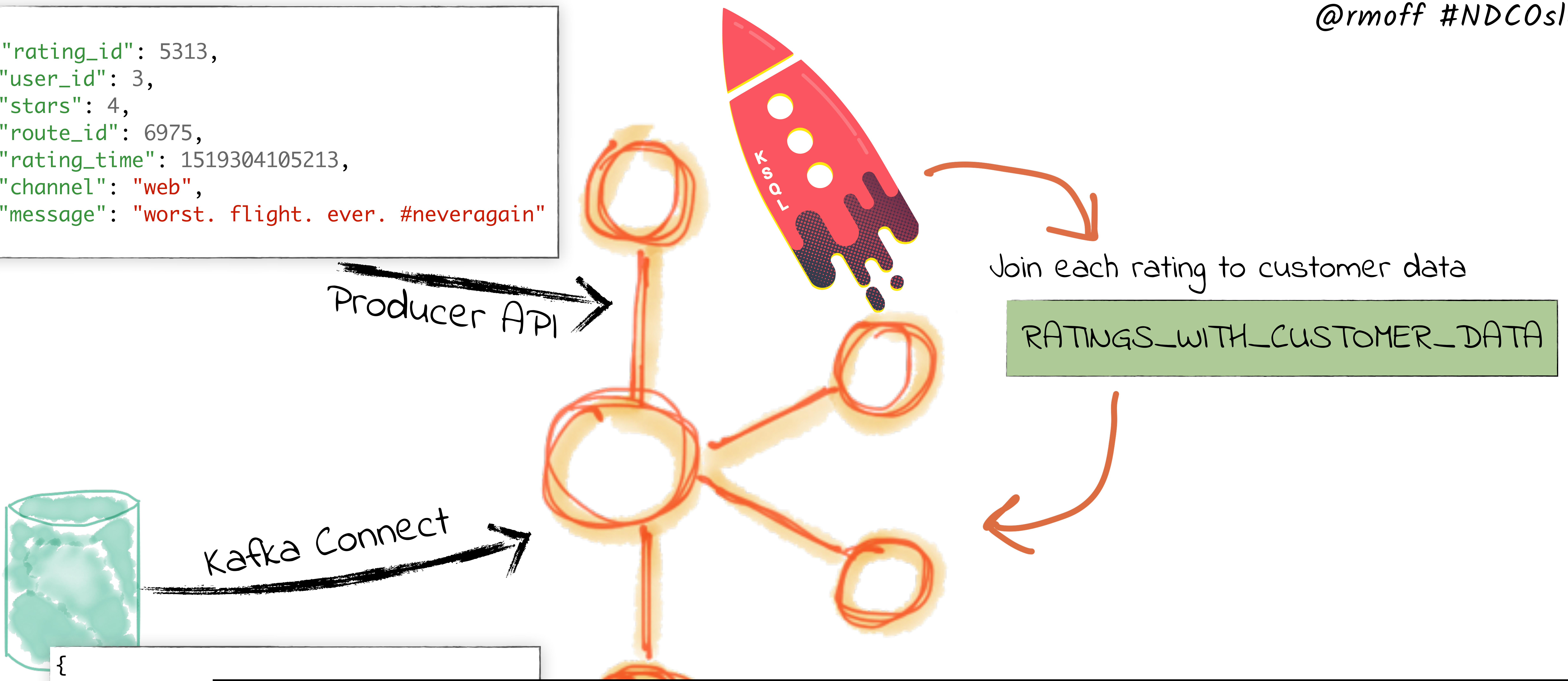
The truth is the log.

The database is a cache of a subset of the log.

—Pat Helland
Immutability Changes Everything
http://cidrdb.org/cidr2015/Papers/CIDR15_Paper16.pdf

Photo by Bobby Burch on Unsplash

{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}

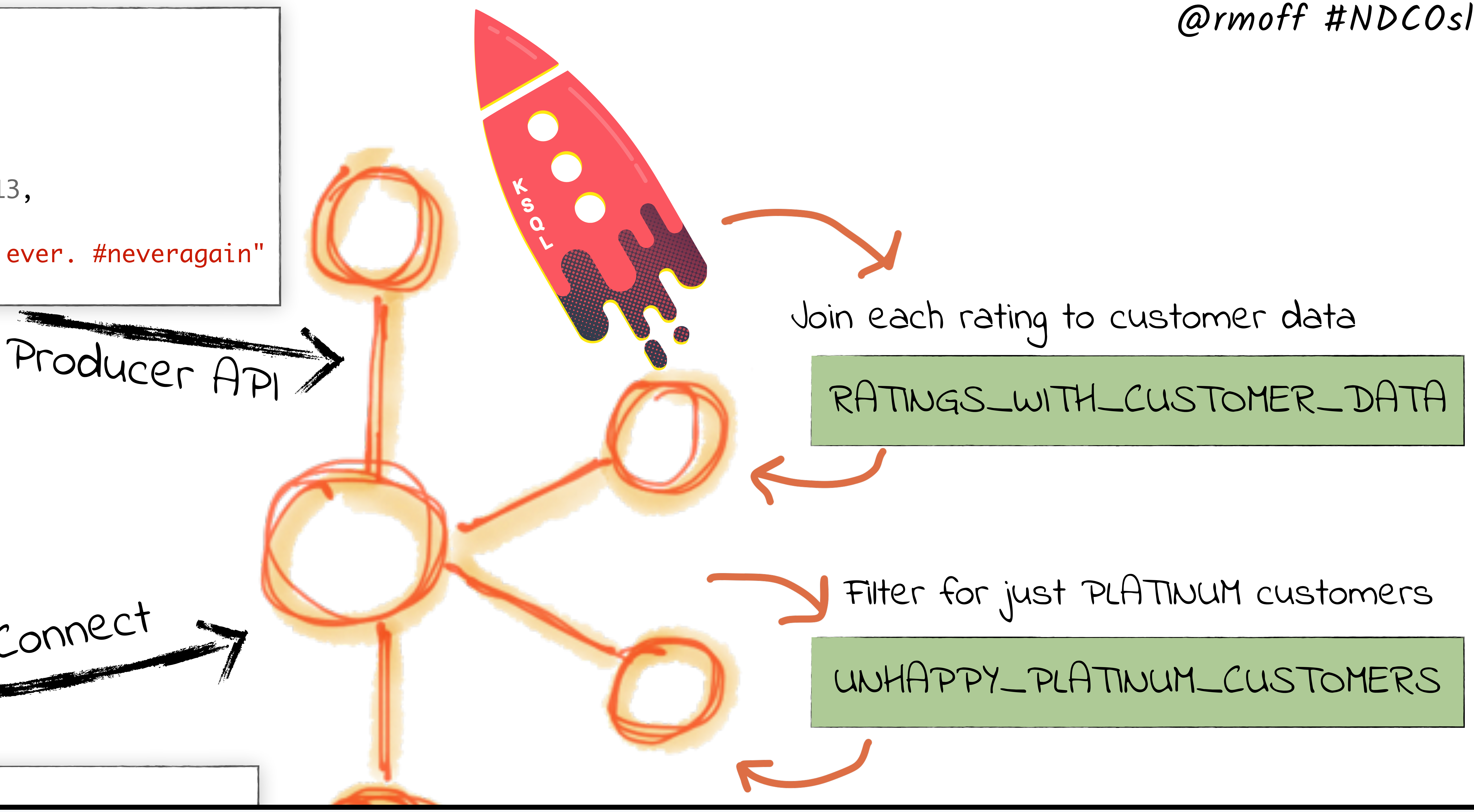


{
 "id": 3,
 "first_name
 "last_name"
 "email": "m
 "gender": "
 "club_statu
 "comments":
}

```
CREATE STREAM RATINGS_WITH_CUSTOMER_DATA AS
SELECT * FROM RATINGS LEFT JOIN CUSTOMERS
ON R.ID=C.ID;
```

@rmoff #NDCOslo


```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}
```

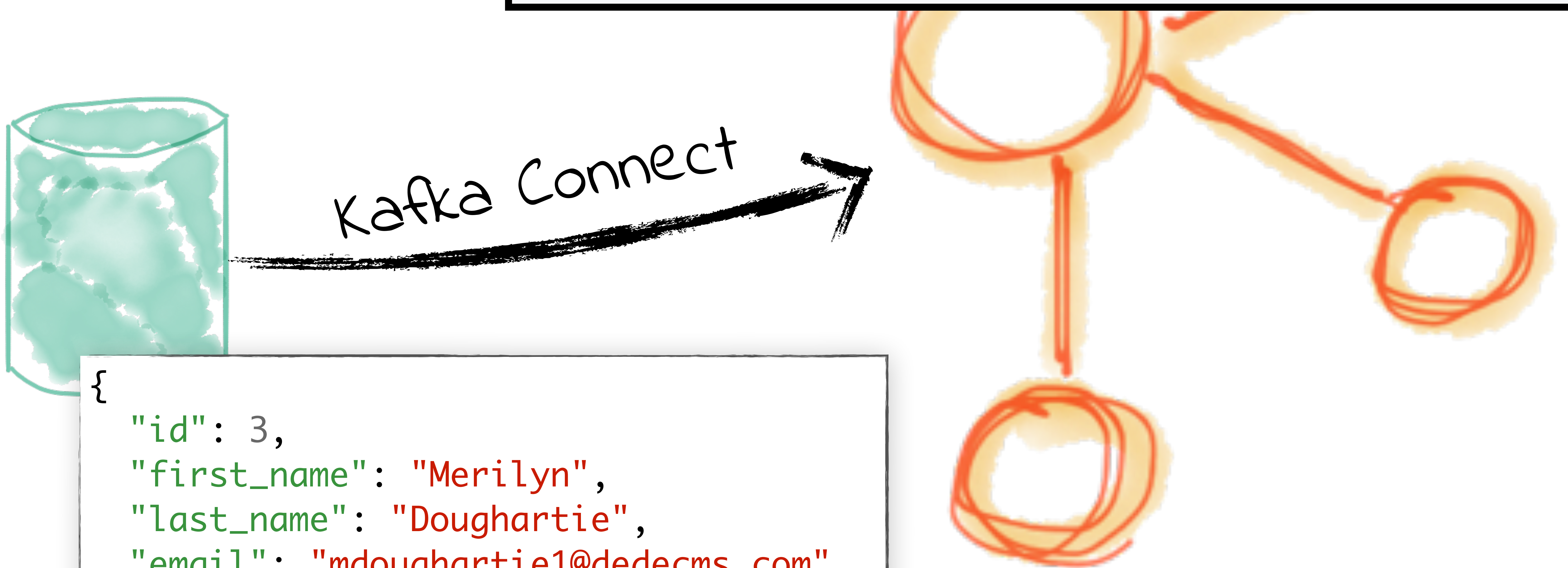




```
{
 "id": 3,
 "first_name
 "last_name"
 "email": "m
 "gender": "
 "club_statu
 "comments":
}
```


```sql
CREATE STREAM UNHAPPY_PLATINUM_CUSTOMERS AS
SELECT * FROM RATINGS_WITH_CUSTOMER_DATA
WHERE  STARS < 3
```

```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 15193041052
 "channel": "web",
 "message": "worst. flight.
}
```


```sql
CREATE TABLE RATINGS_BY_CLUB_STATUS AS
SELECT CLUB_STATUS, COUNT(*)
FROM RATINGS_WITH_CUSTOMER_DATA
     WINDOW TUMBLING (SIZE 1 MINUTES)
GROUP BY CLUB_STATUS;
```

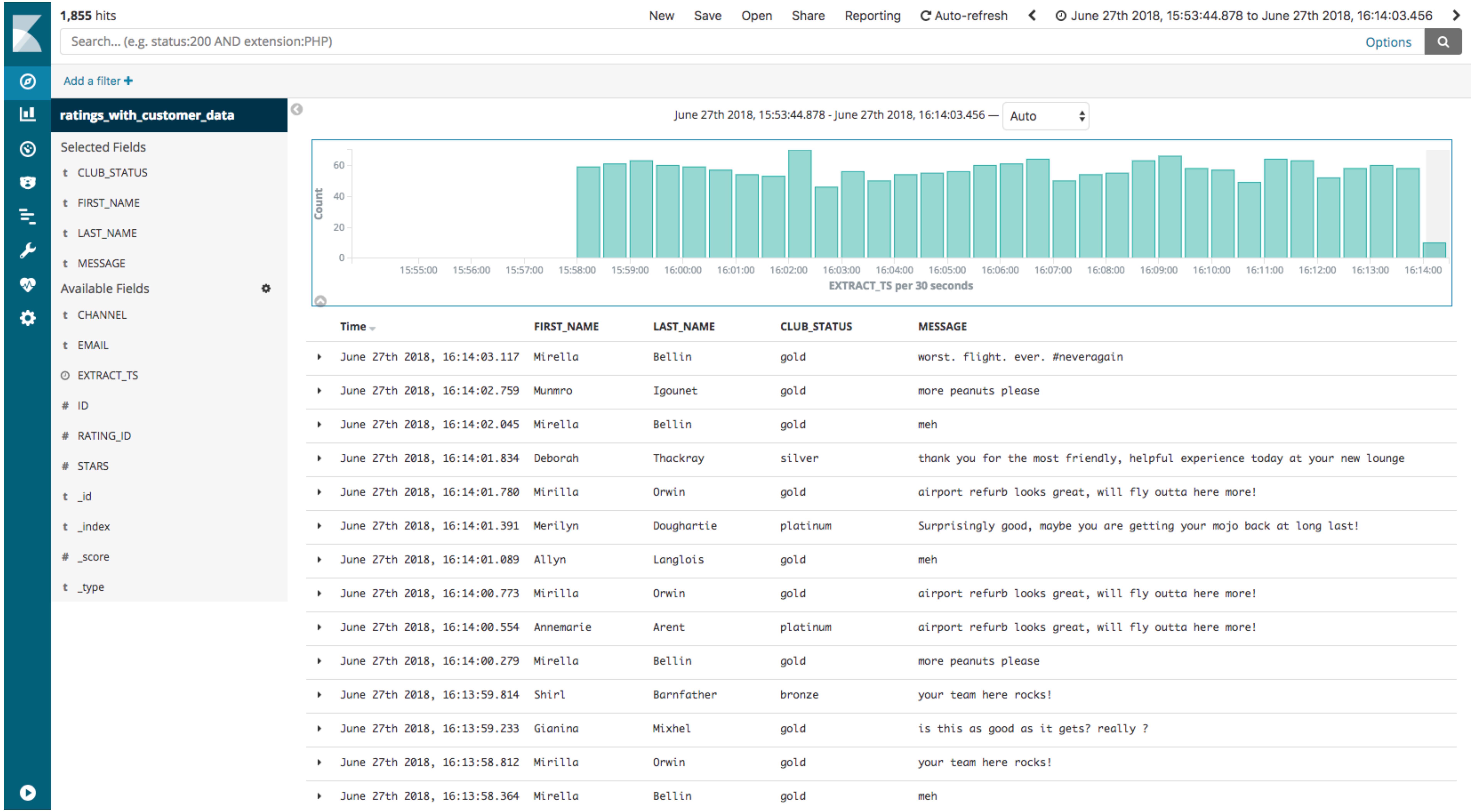



```
{
  "id": 3,
  "first_name": "Merilyn",
  "last_name": "Doughartie",
  "email": "mdoughartie1@dedecms.com",
  "gender": "Female",
  "club_status": "platinum",
  "comments": "none"
}
```


Aggregate per-minute by CLUB_STATUS

RATINGS_BY_CLUB_STATUS_1MIN

Stream to Elasticsearch



https://cnfl.io/start-ksql-workshop

7. Kafka Connect / Integrating Kafka with a database
8. The Stream/Table duality
9. Joining Data in KSQL
10. Streaming Aggregates
11. Optional: Stream data to Elasticsearch


https://www.confluent.io/ksql
http://cnfl.io/demo-scene
http://cnfl.io/book-bundle
http://cnfl.io/slack
@rmoff

Related Talks

- The Changing Face of ETL: Event-Driven Architectures for Data Engineers
  - 📖 Slides
- ATM Fraud detection with Kafka and KSQL
  - 📖 Slides
  - 👾 Code
  - 📽 Recording
- Embrace the Anarchy: Apache Kafka's Role in Modern Data Architectures
  - 📖 Slides
  - 📽 Recording
- Apache Kafka and KSQL in Action : Let's Build a Streaming Data Pipeline!
  - 📖 Slides
  - 👾 Code
  - 📽 Recording
- No More Silos: Integrating Databases and Apache Kafka
  - 📖 Slides
  - 👾 Code (MySQL)
  - 👾 Code (Oracle)
  - 📽 Recording

#EOF