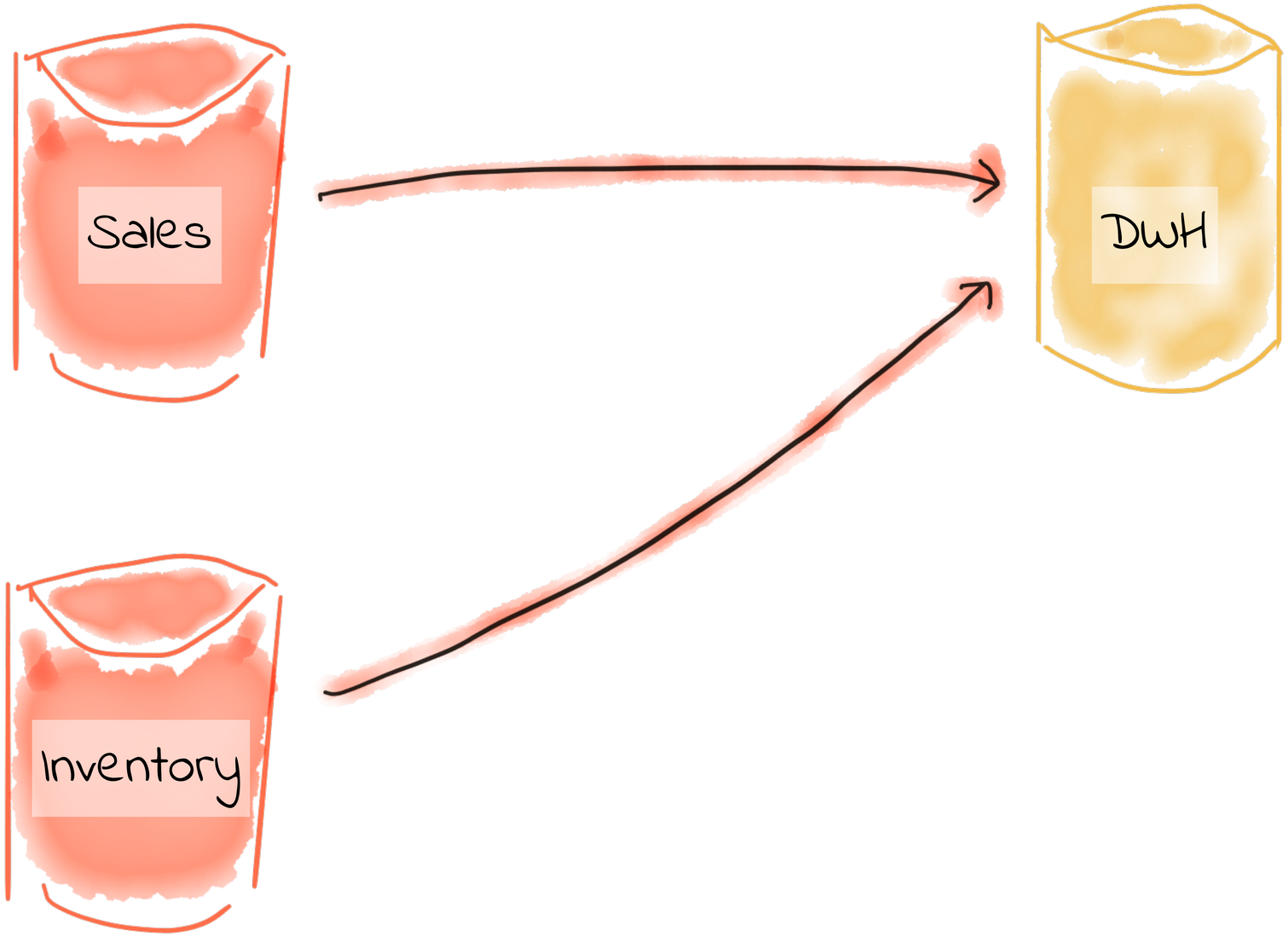
Streaming ETL in Practice with Oracle, Apache Kafka, and KSQL

@rmoff

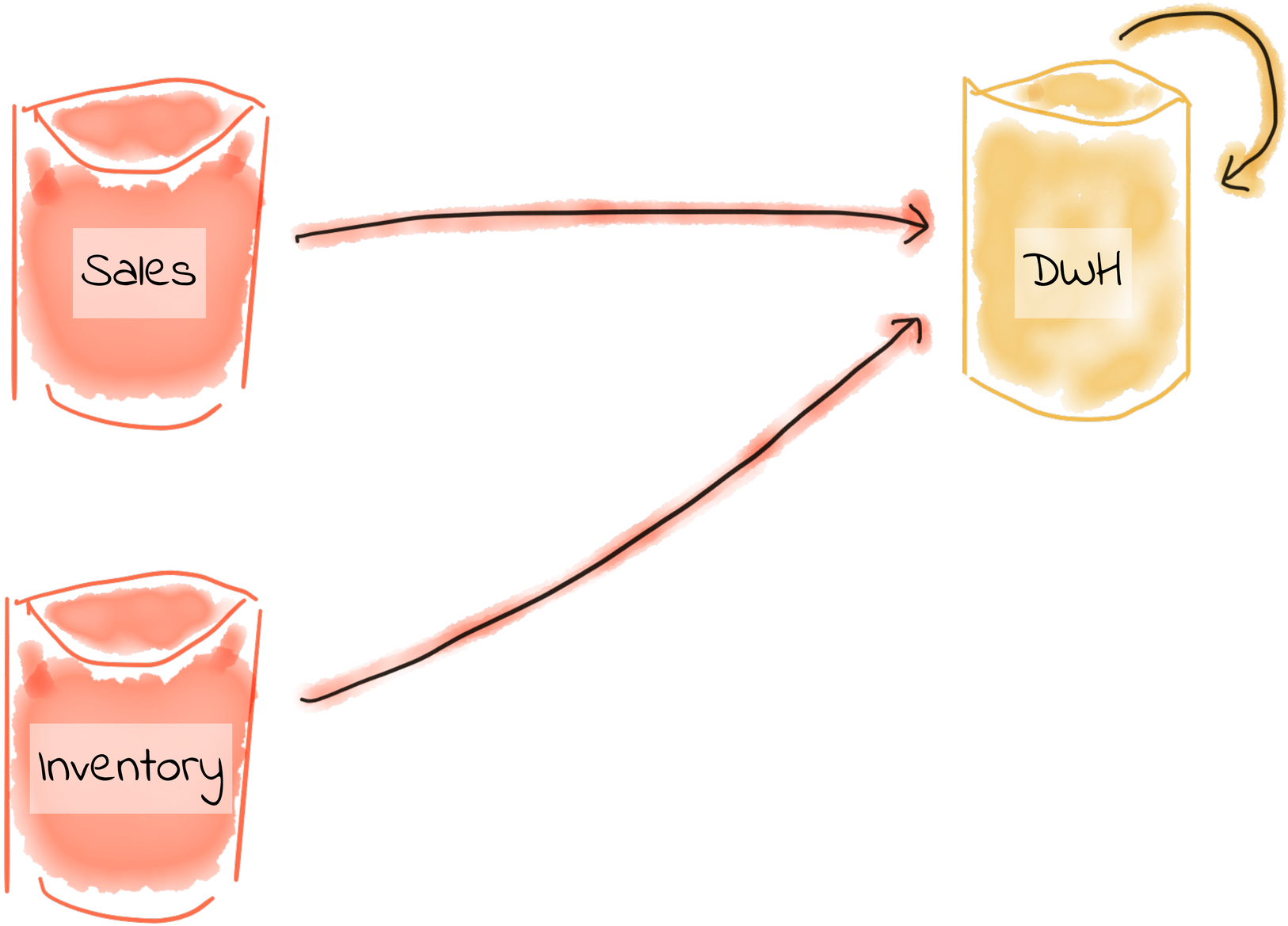
#KScope19

Analytics—In the beginning...

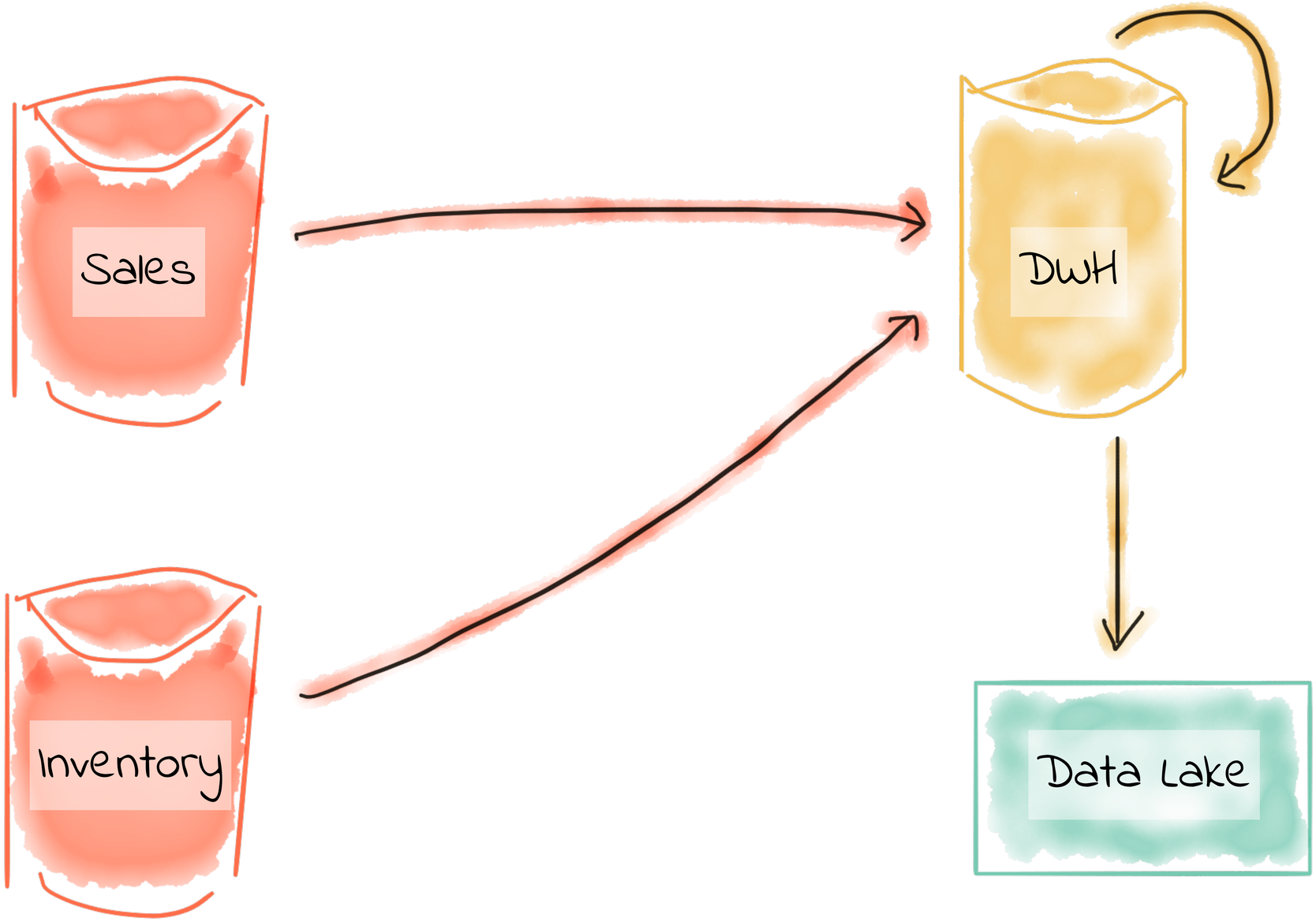
And then there were more data sources...



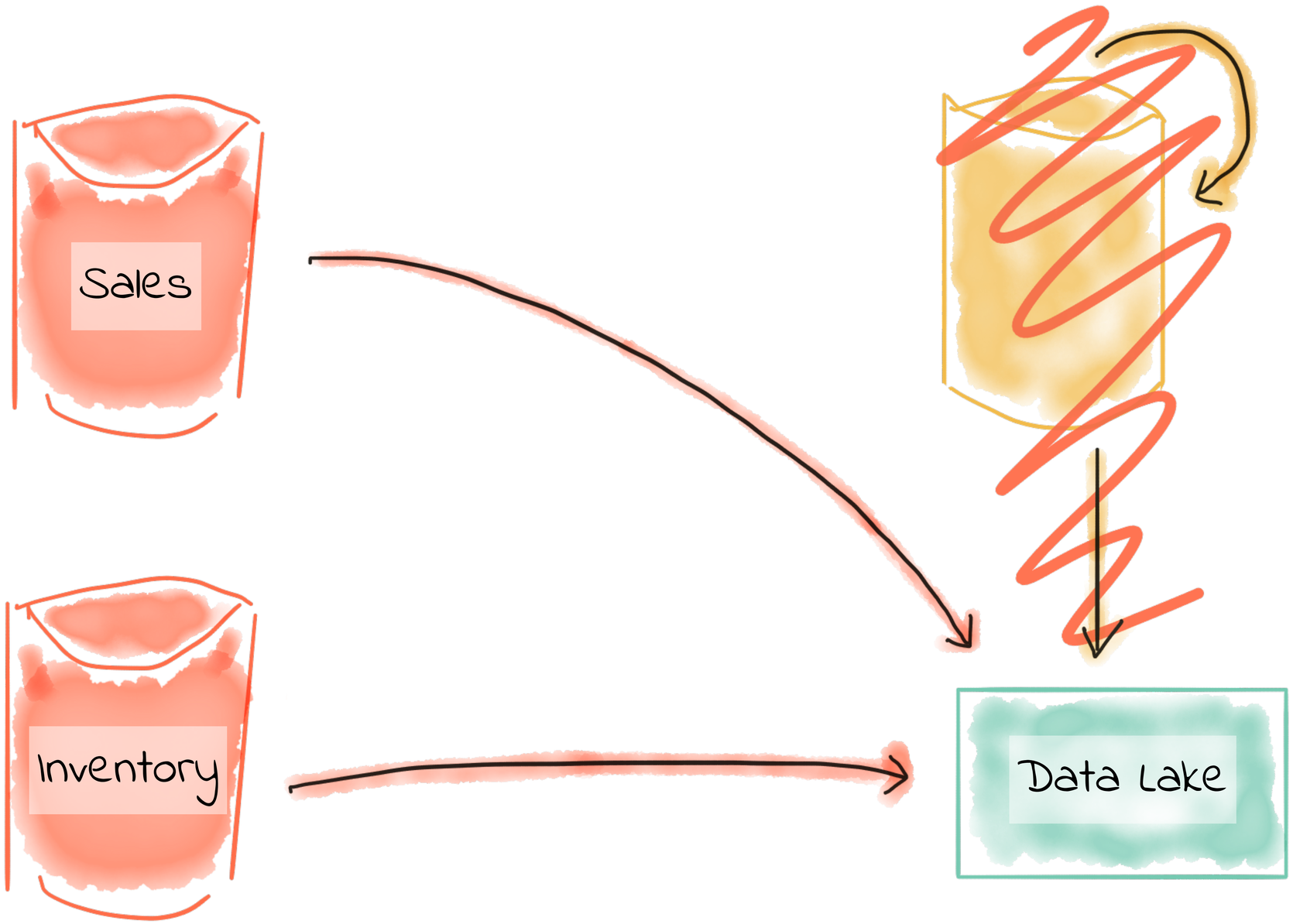
Batch Transformations … (ETL / ELT)



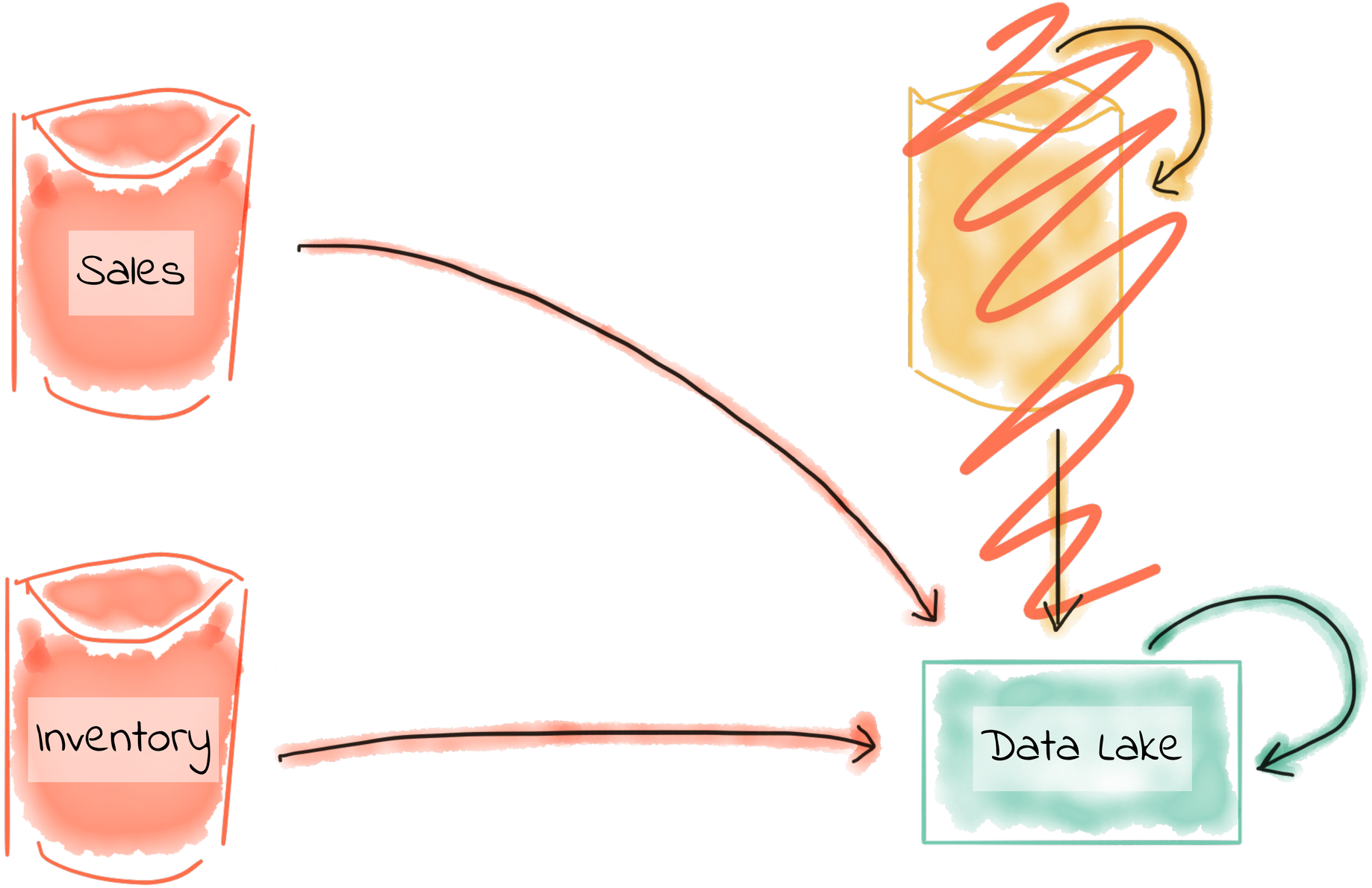
Add a Data Lake...

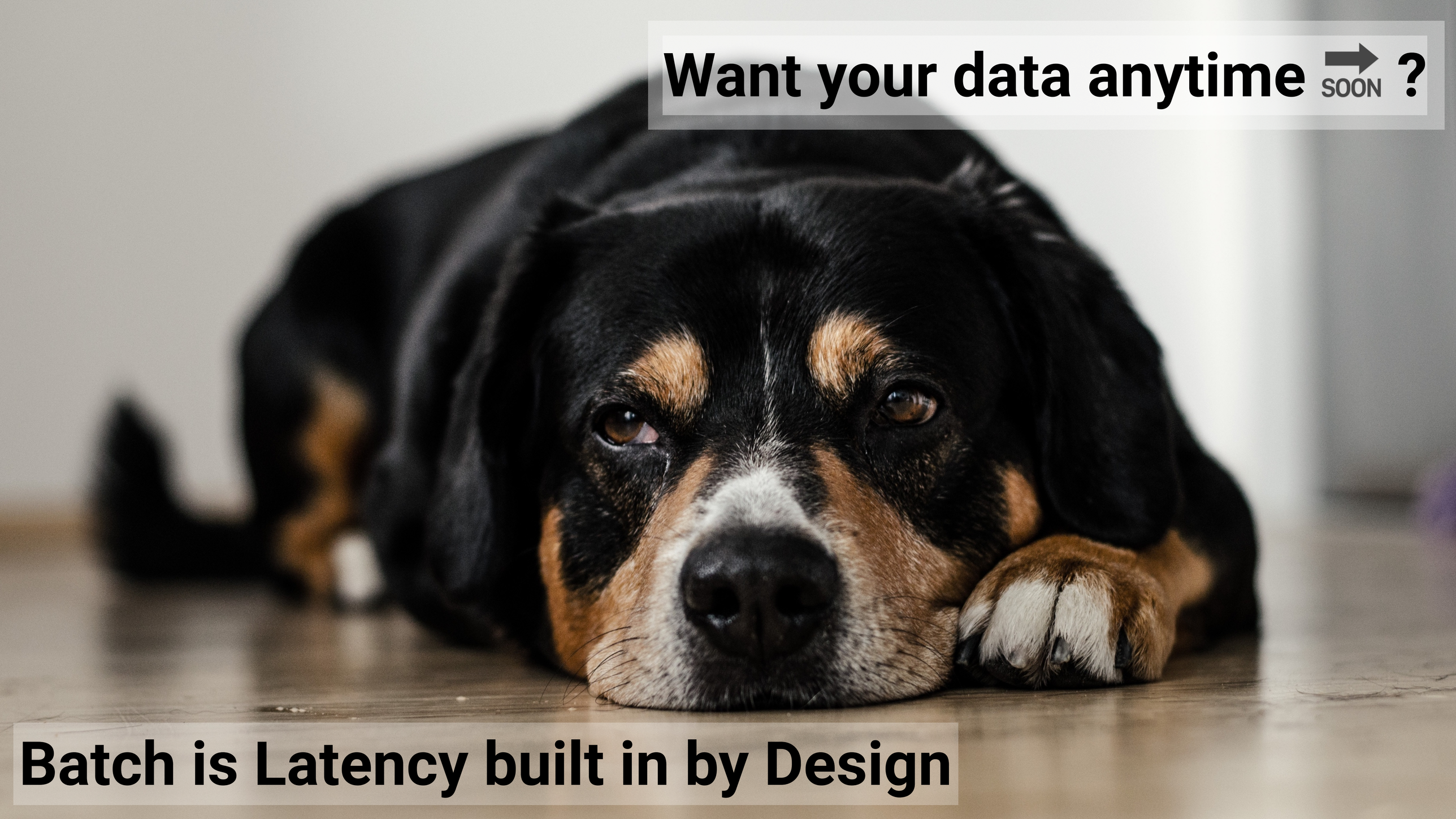


...or Replace the Data Warehouse



Still need to do Batch transformations...




Want your data anytime SOON ?
Batch is Latency built in by Design

The World has Changed

Internet of Things

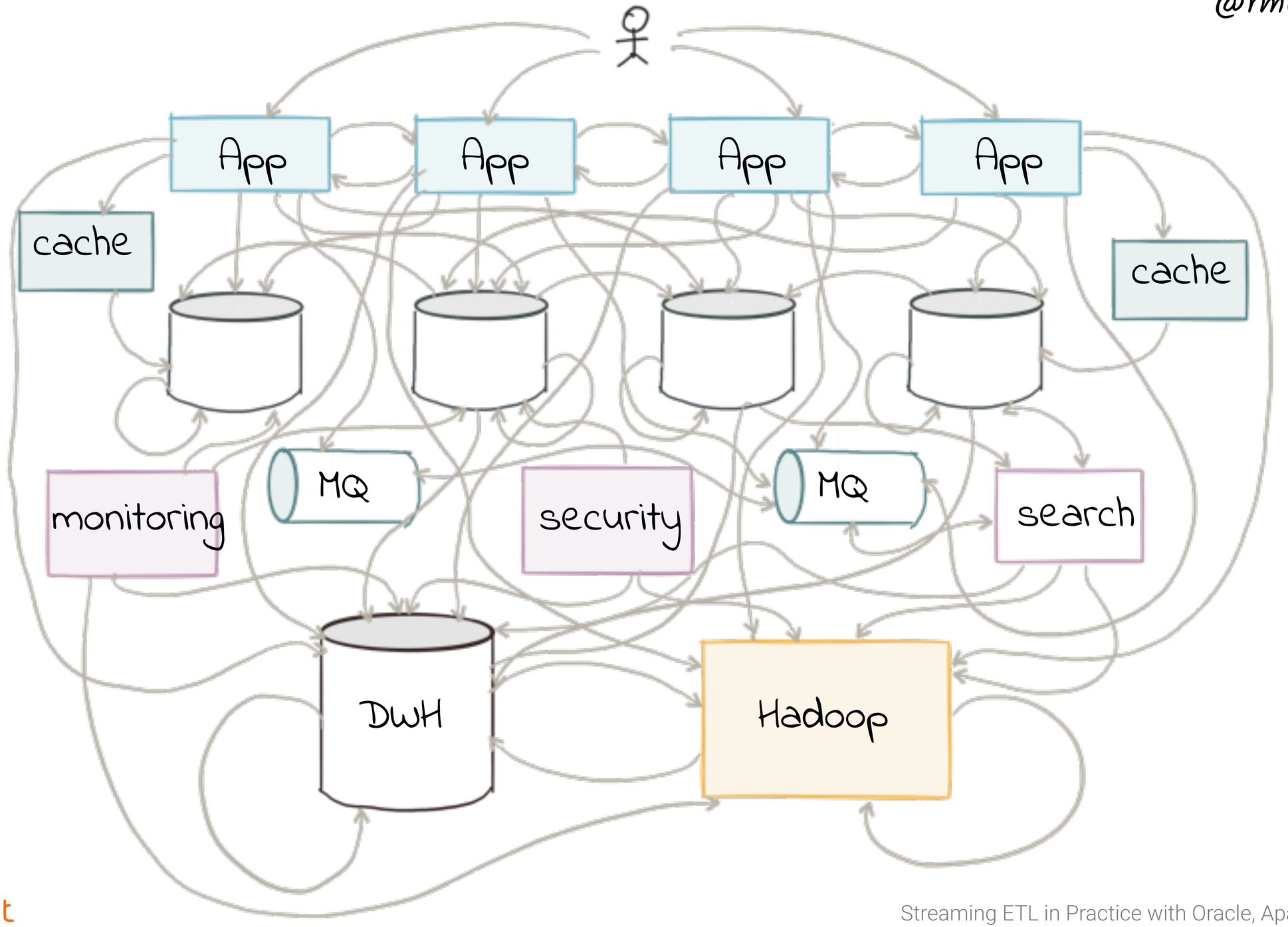
Microservices

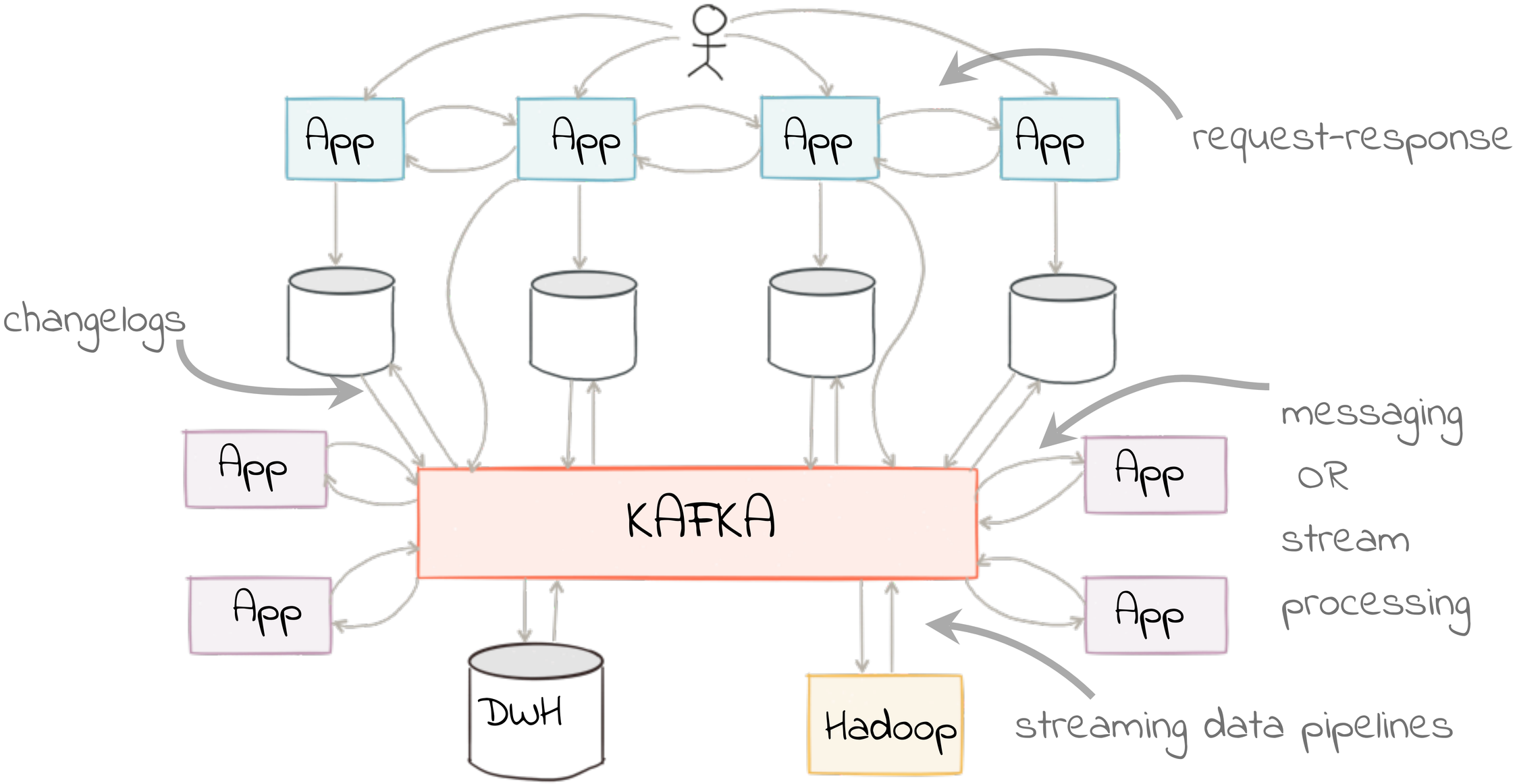
Mobile

Machine Learning

Photo by Denys Nevozhai on Unsplash

Photo by Rosie Fraser on Unsplash


App
App
App
App
cache
cache
monitoring
MQ
security
MQ
search
DWH
Hadoop


App
App
App
App
request-response
changelogs
App
KAFKA
App
App
App
messaging
OR
stream
processing
DWH
Hadoop
streaming data pipelines

"But streaming...I've just got data in a database...right?

confluent

"Bold claim: all your data is event streams

confluent

A Customer Experience

A Sale

A Sensor Reading

An Application Log Entry

Databases

The Stream/Table Duality

Stream

Time

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

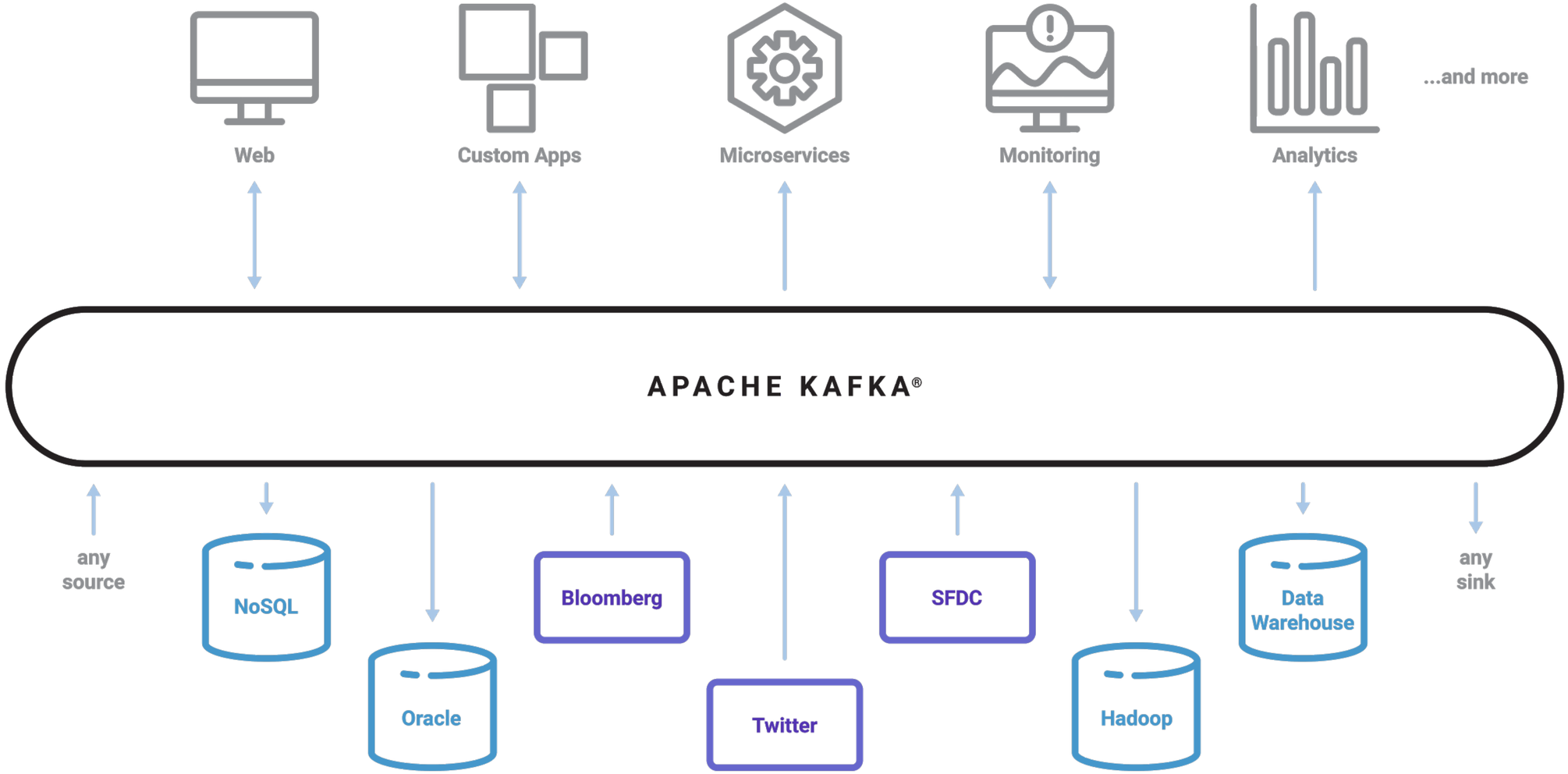
Table

Account ID	Balance
12345	€50

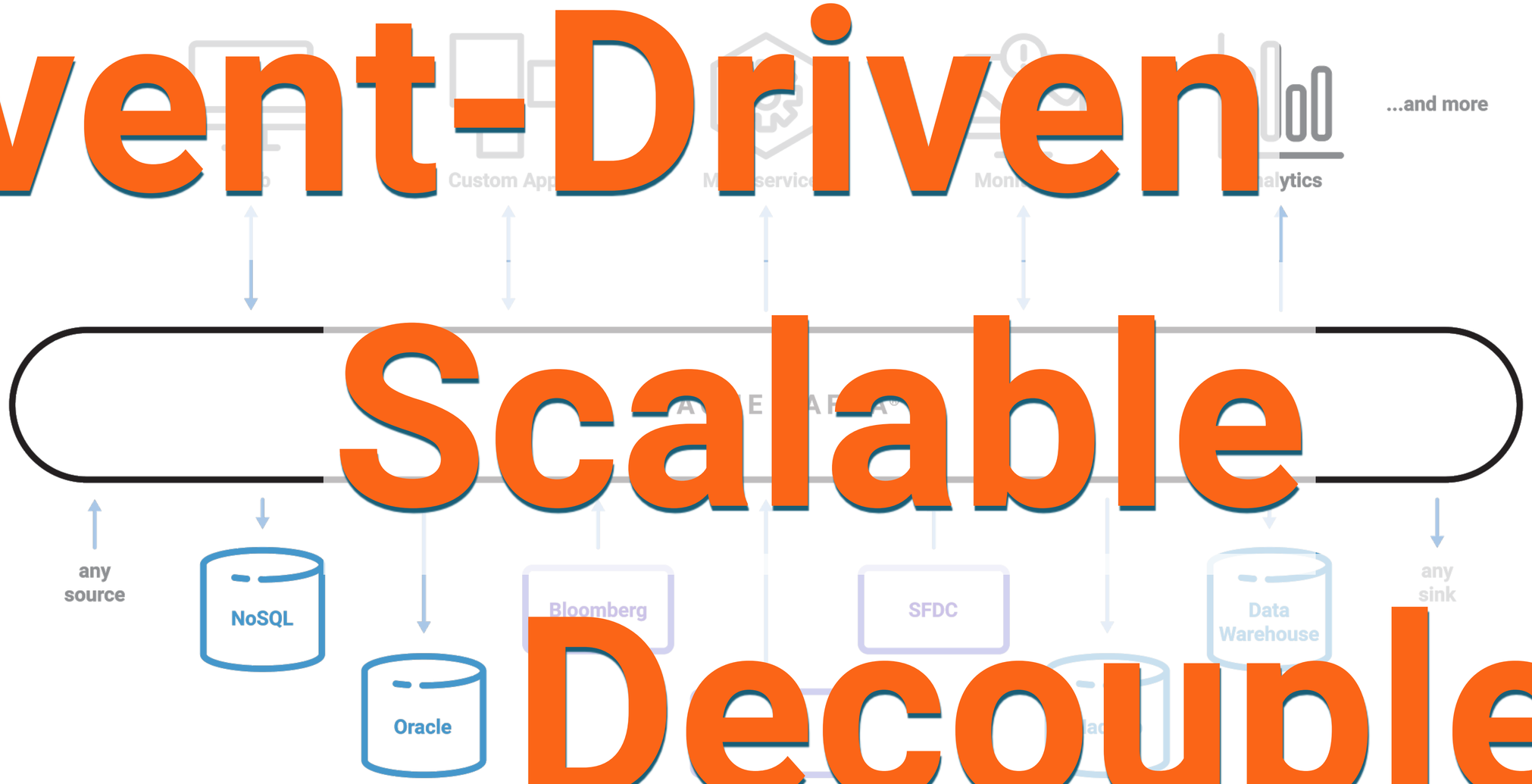
Account ID	Balance
12345	€75

Account ID	Balance
12345	€15

Streaming Platform Vision



Streaming Platform Vision

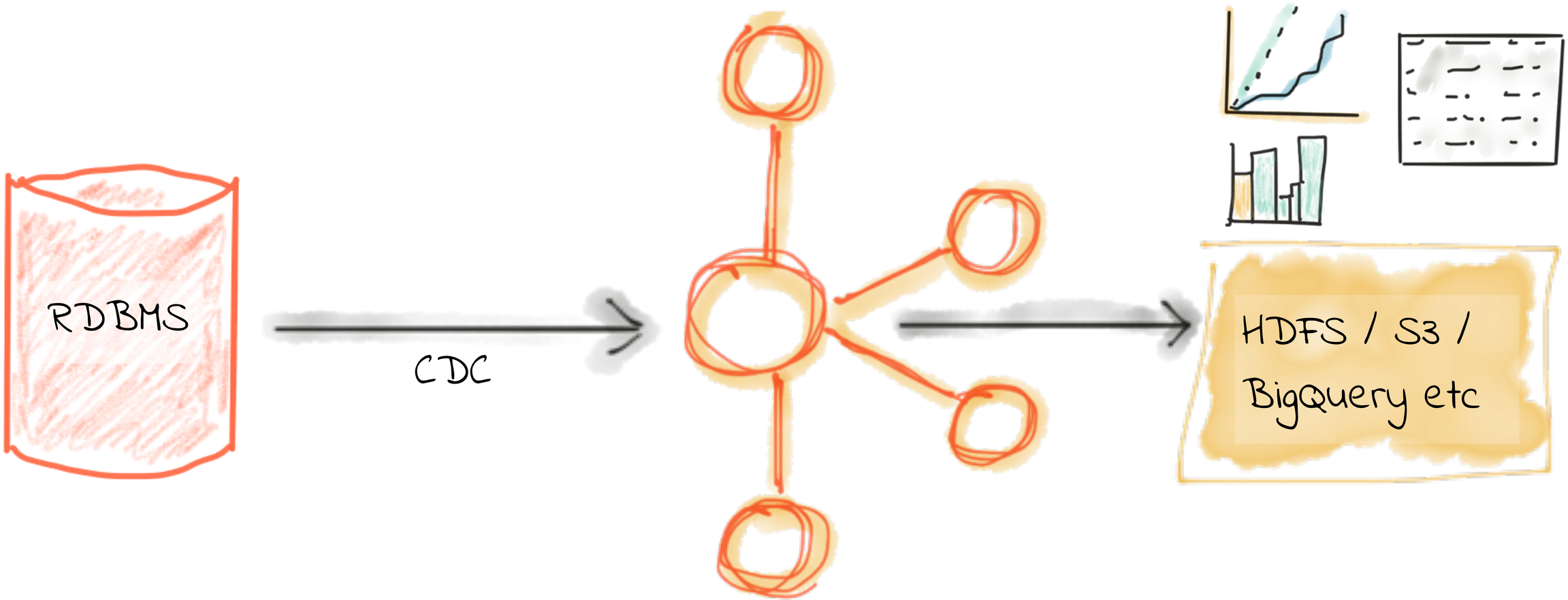


Event-Driven

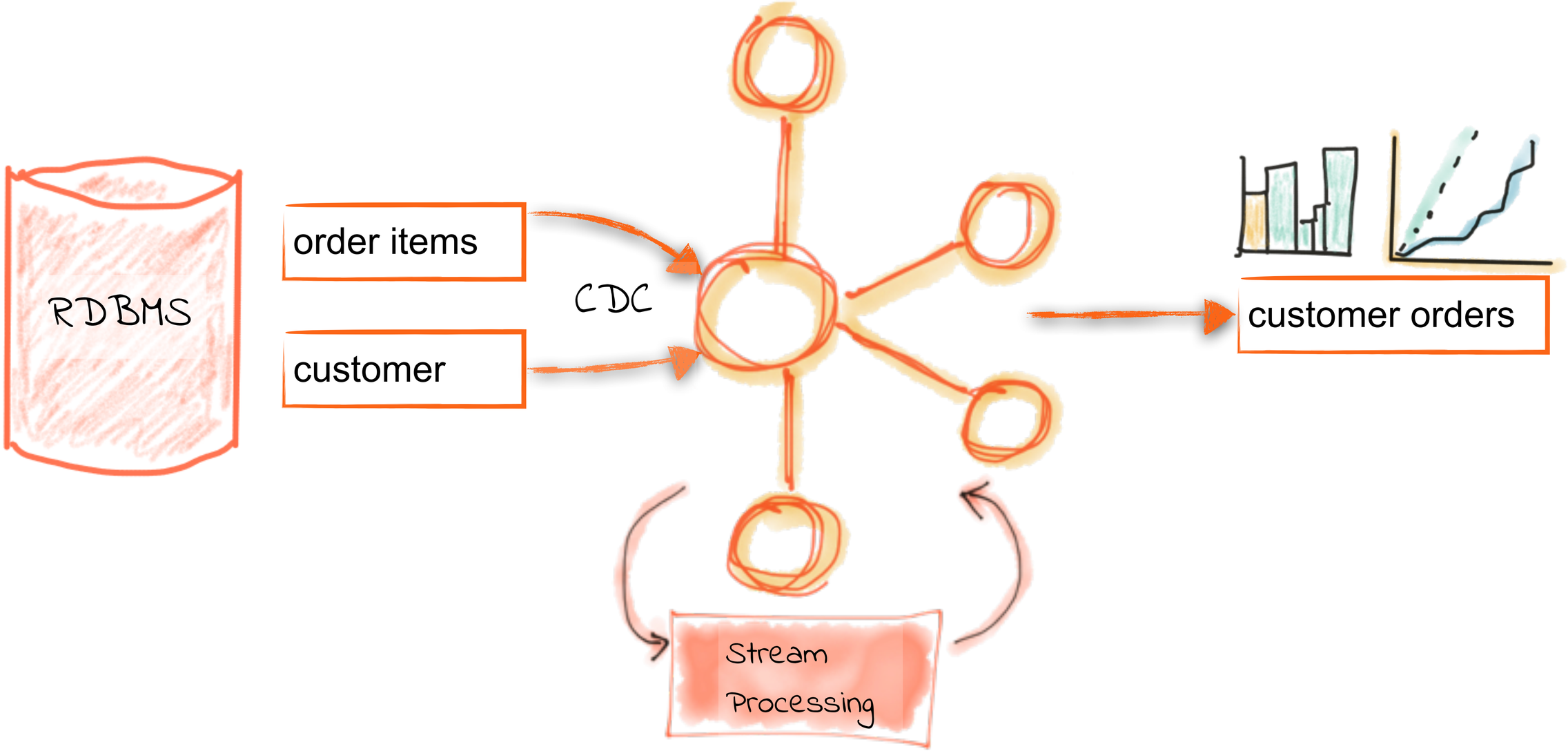
Scalable

Decoupled

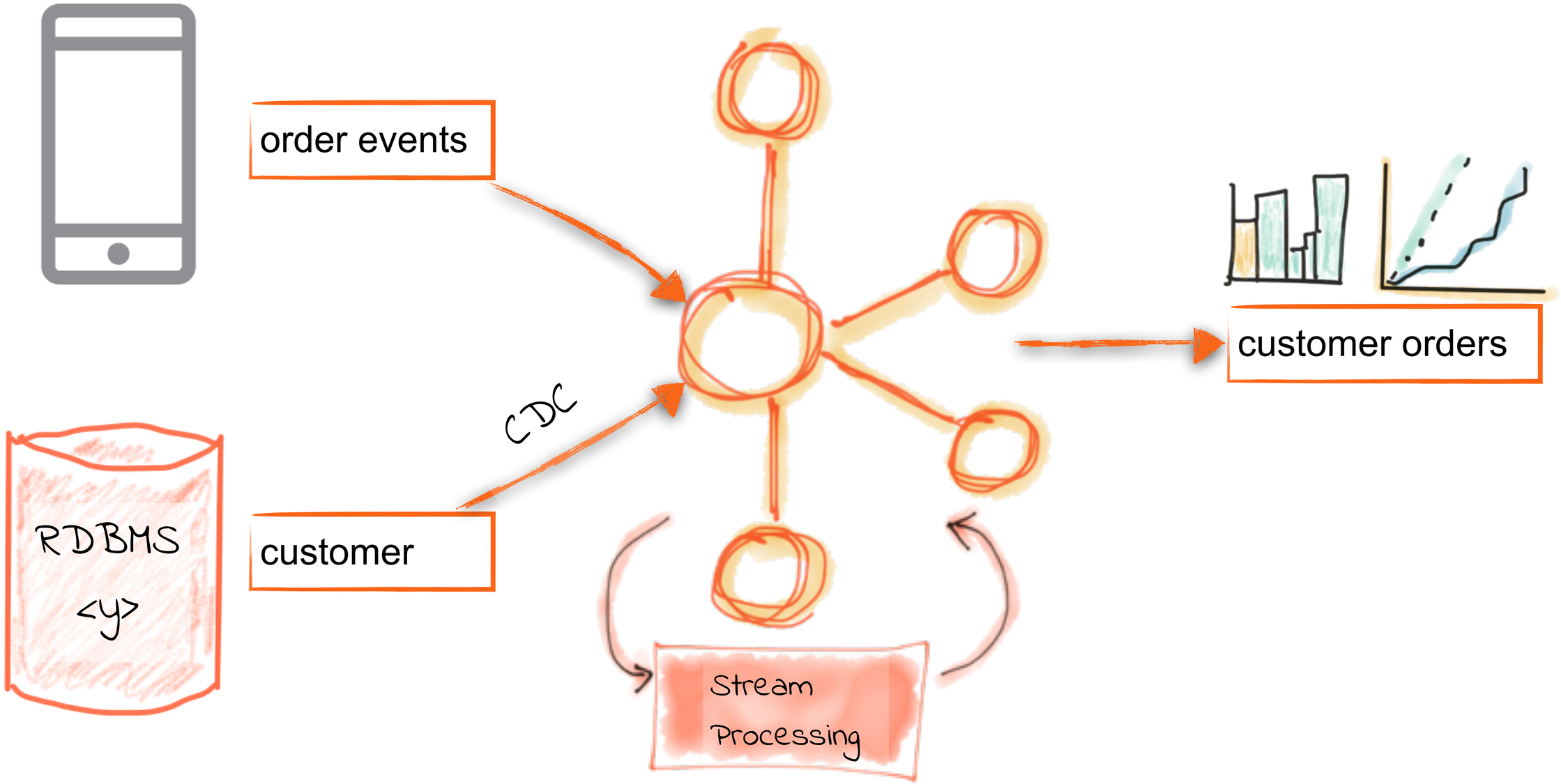
Database offload→Hadoop/Object Storage/Cloud DW for Analytics



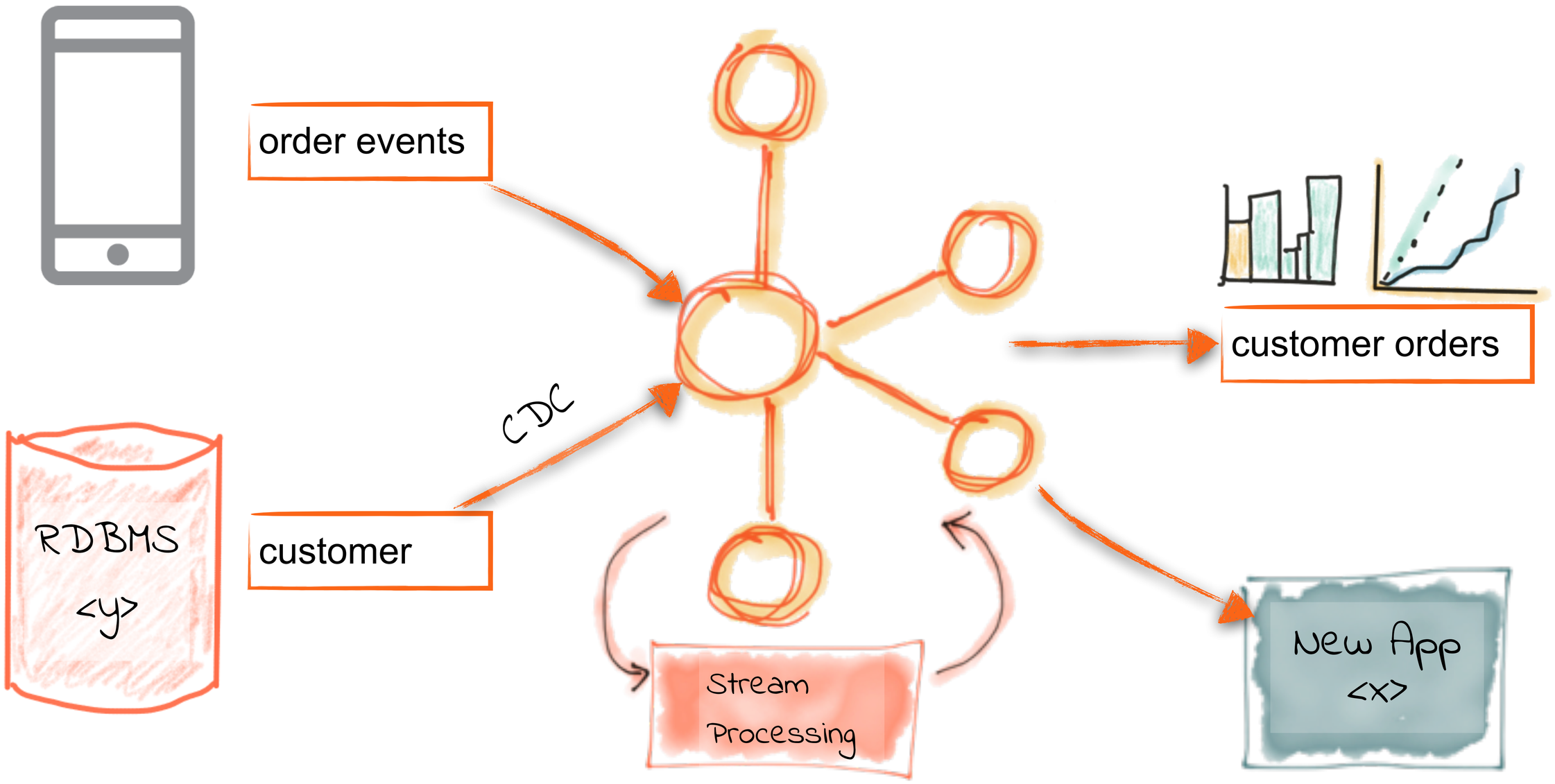
Streaming ETL with Apache Kafka and KSQL



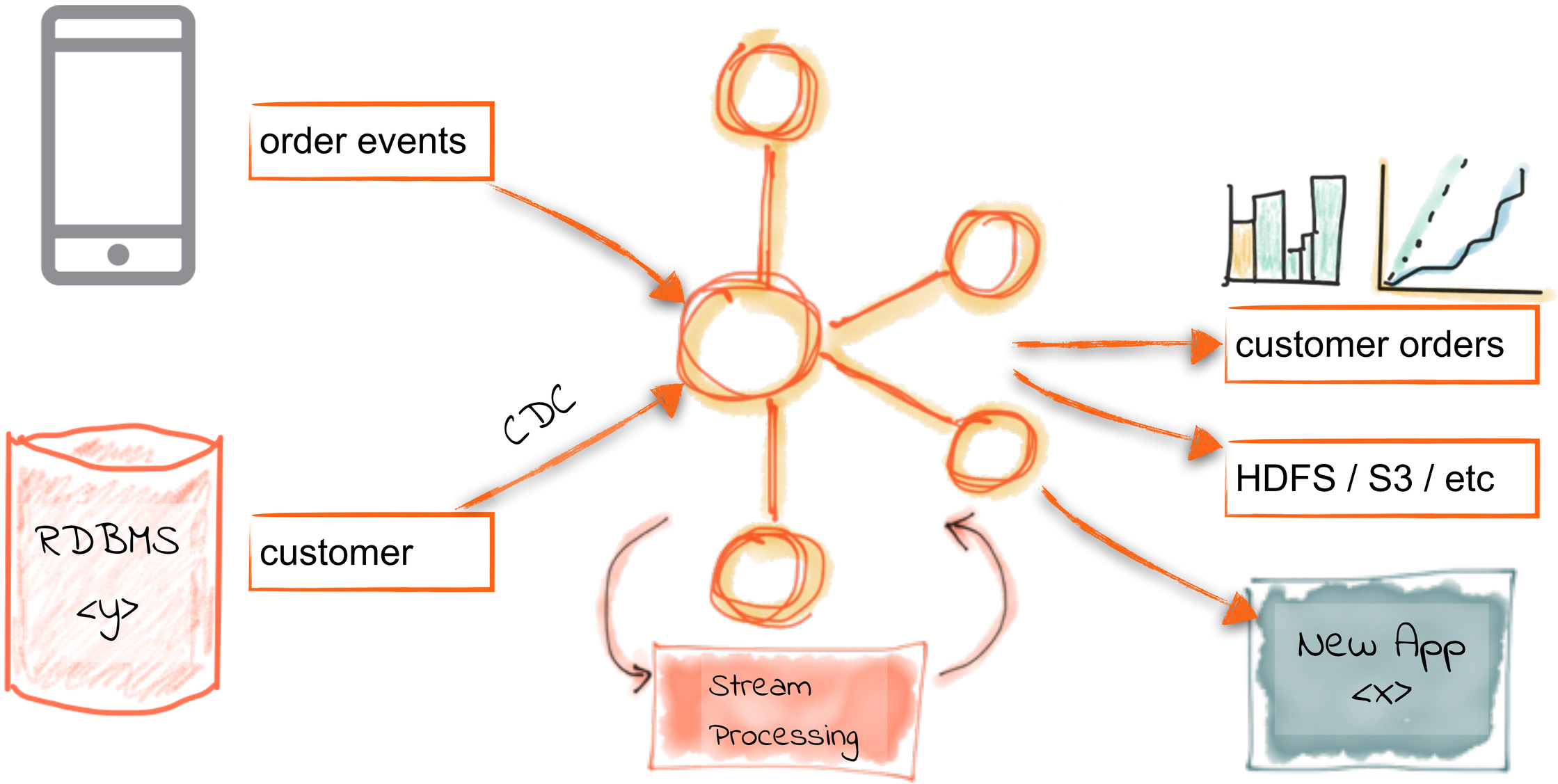
Real-time Event Stream Enrichment with Apache Kafka and KSQL



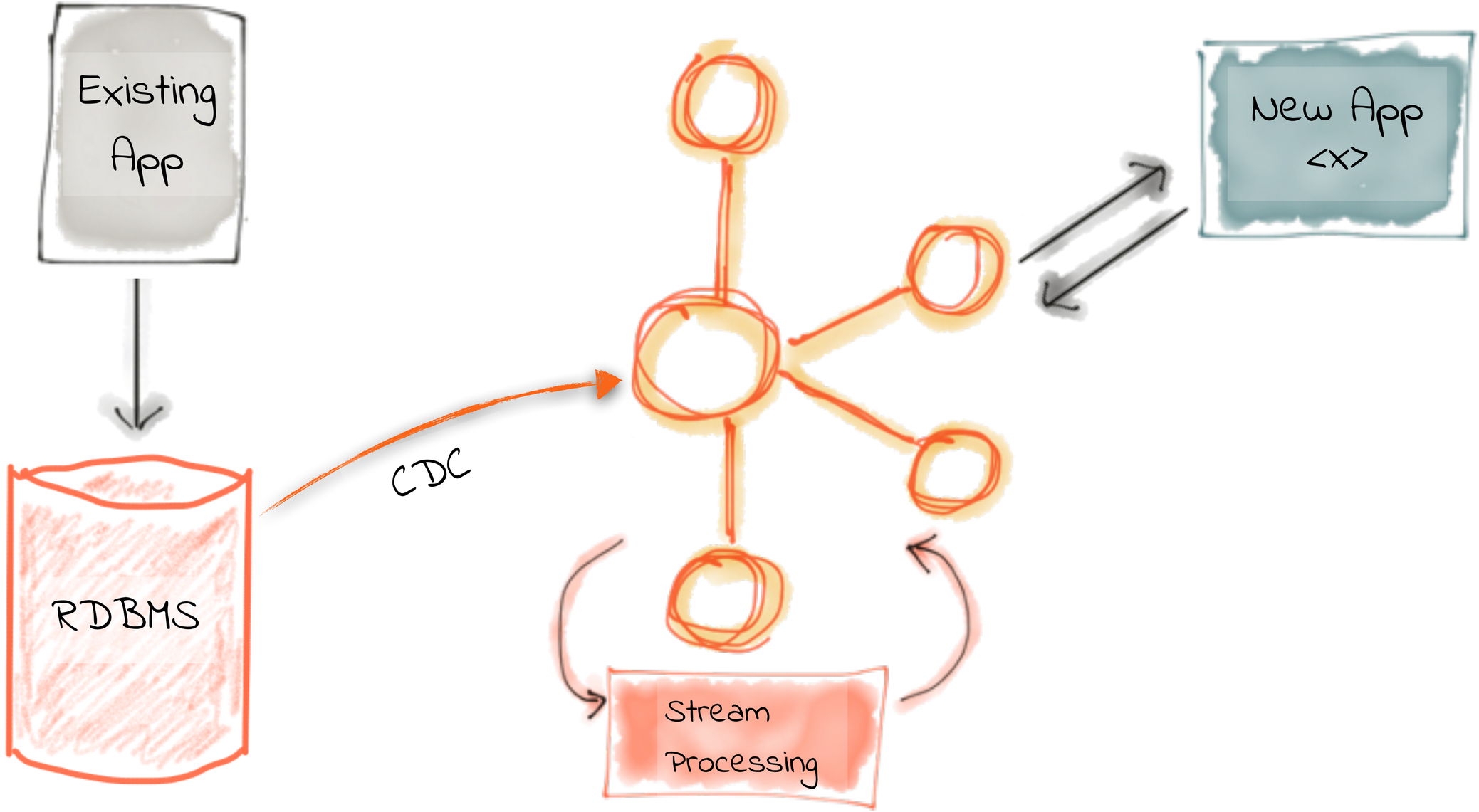
Transform Once, Use Many



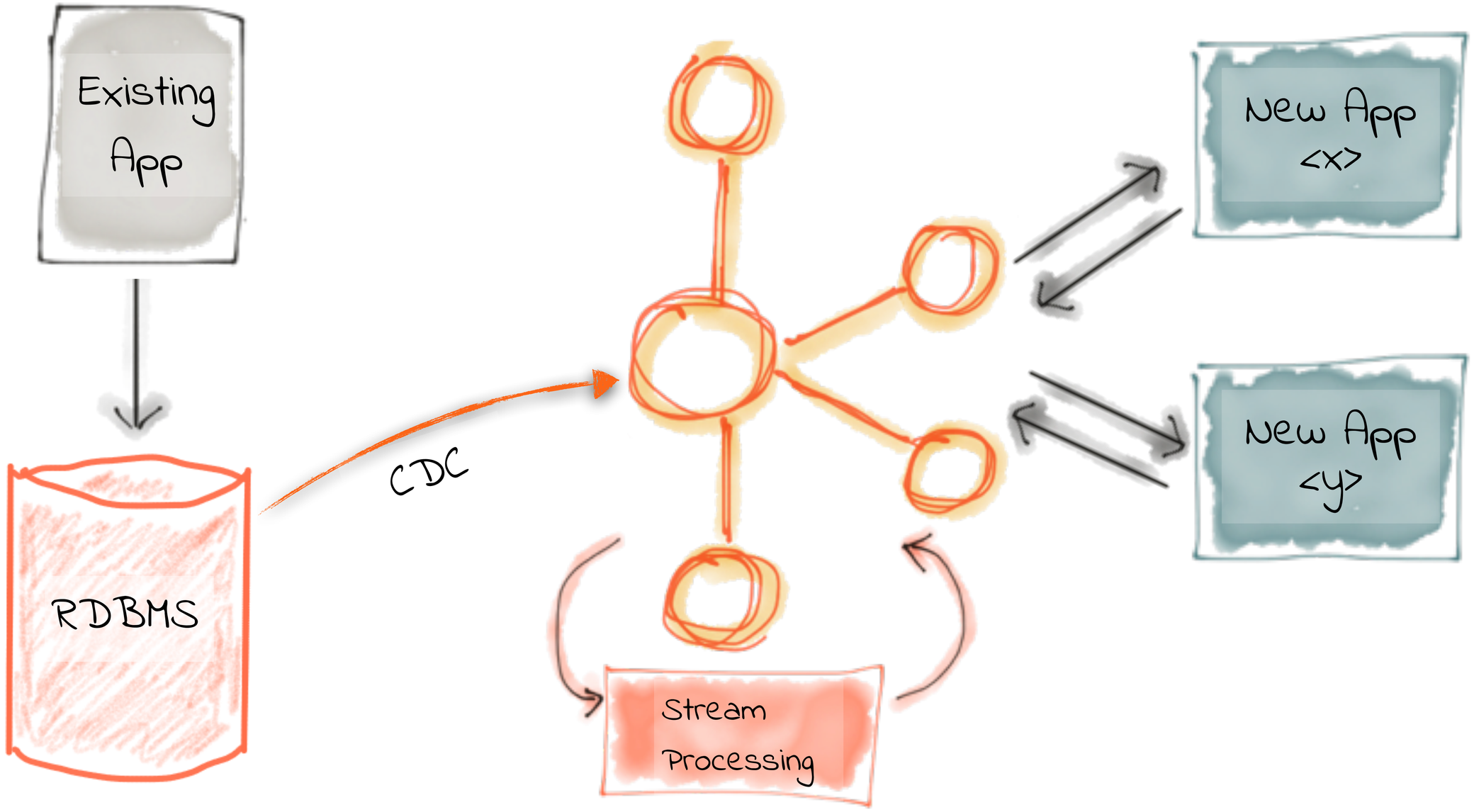
Transform Once, Use Many



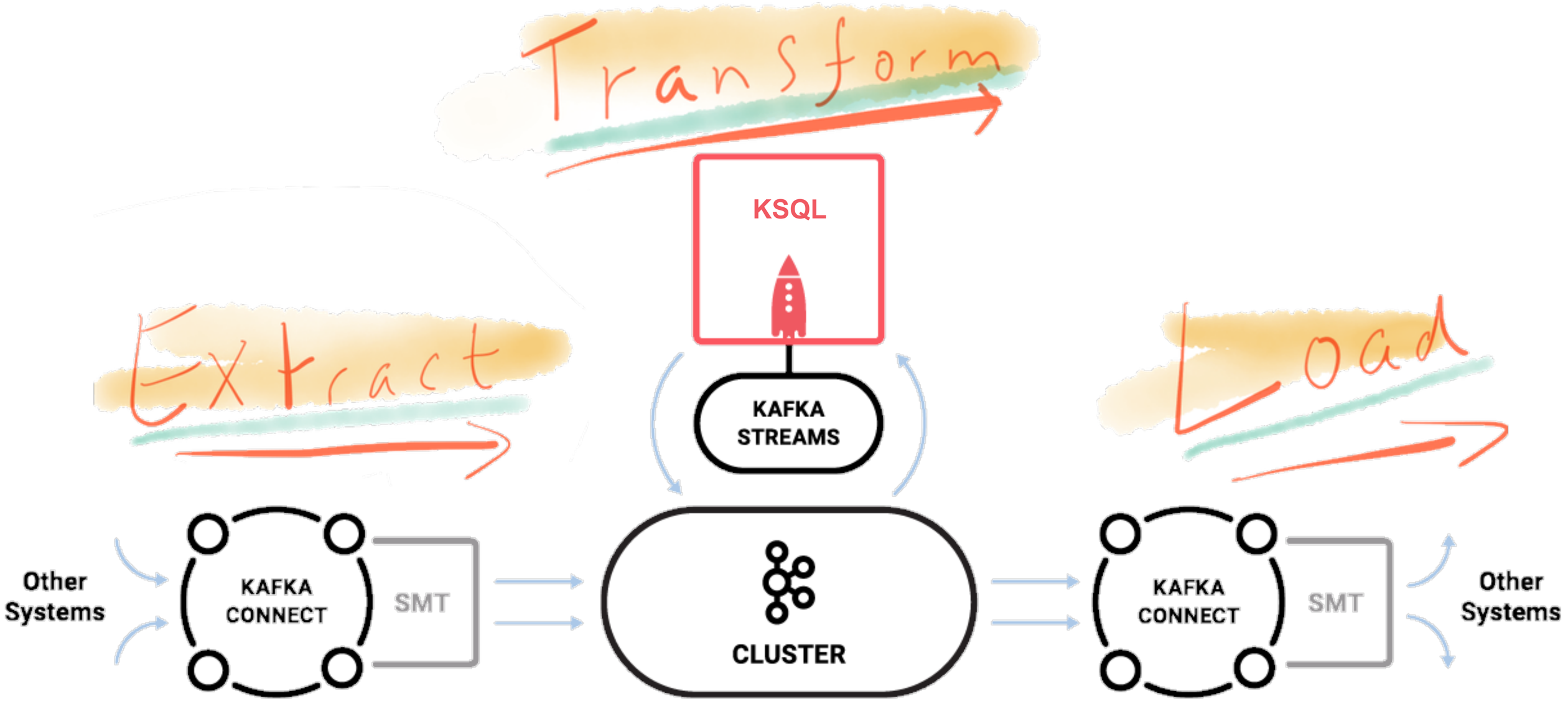
Evolve processing from old systems to new



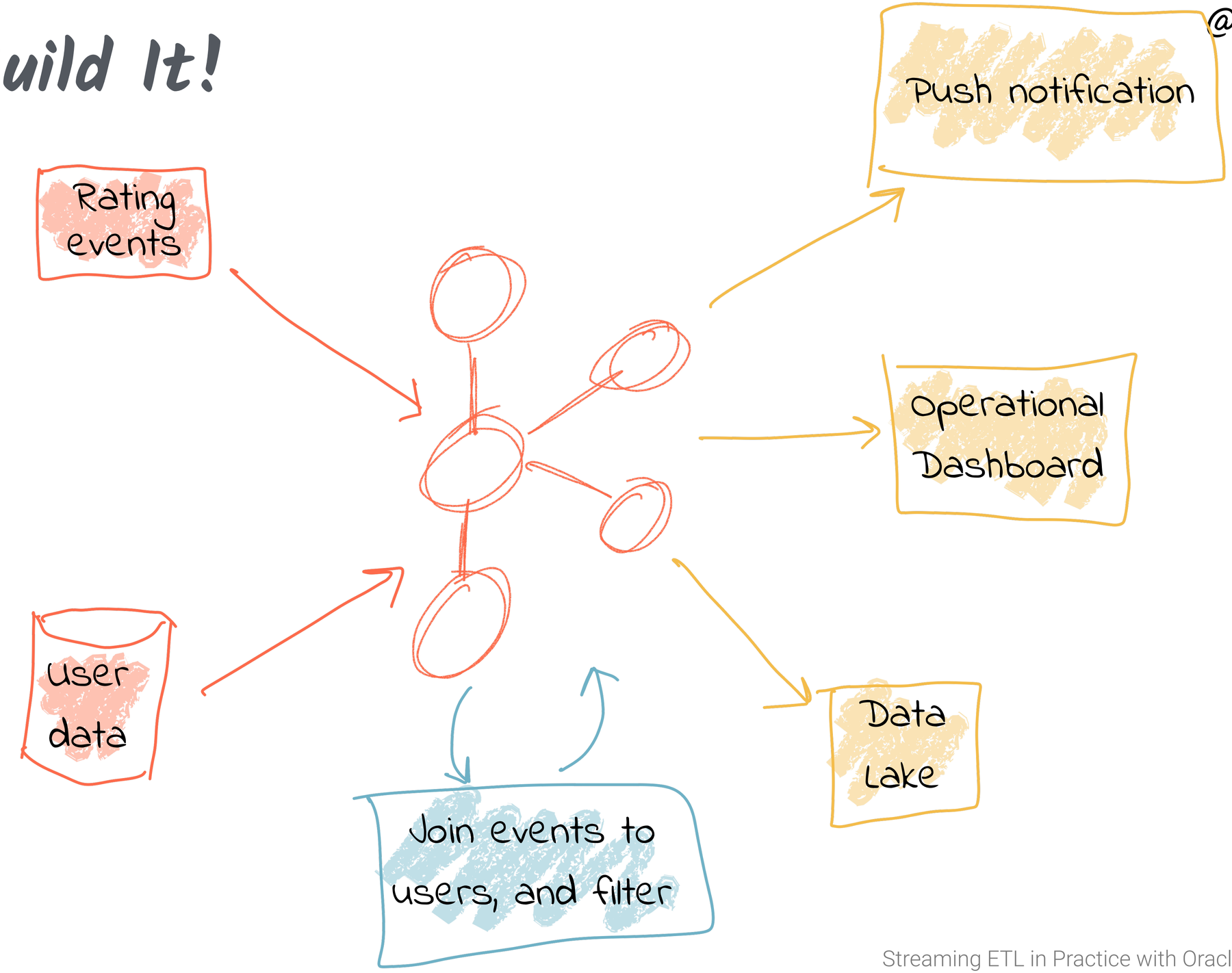
Evolve processing from old systems to new



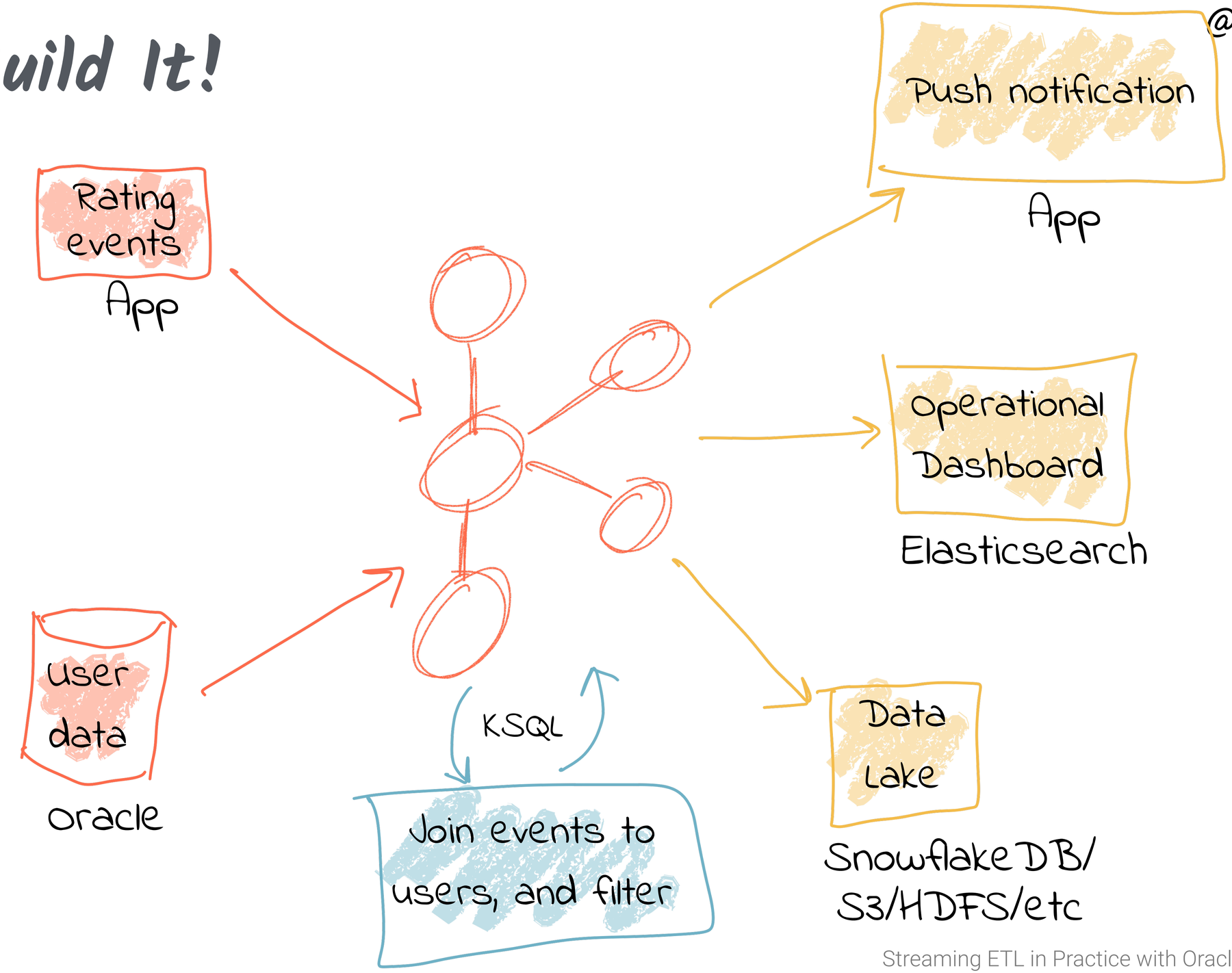
Streaming ETL, powered by Apache Kafka and Confluent Platform



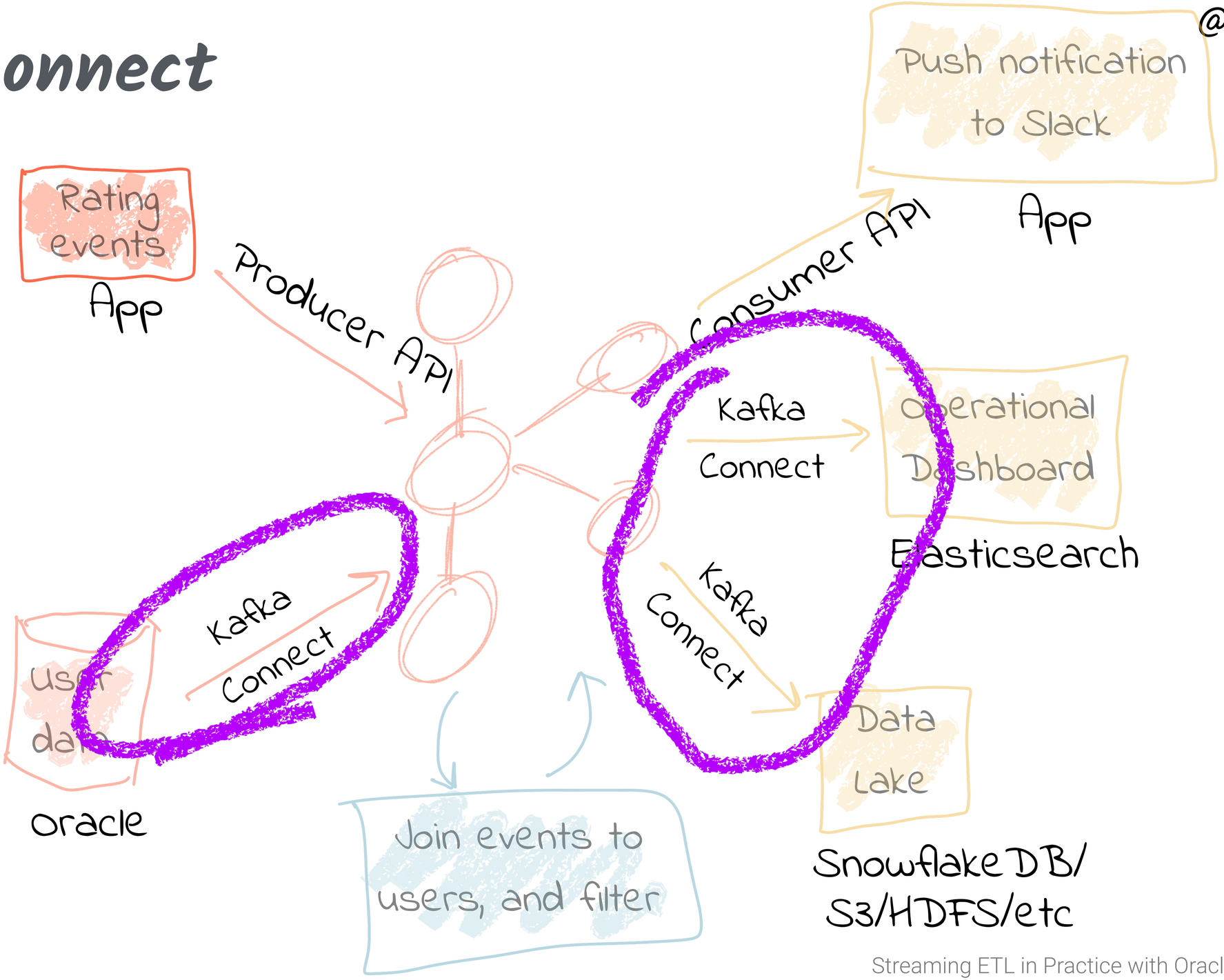
Let's Build It!



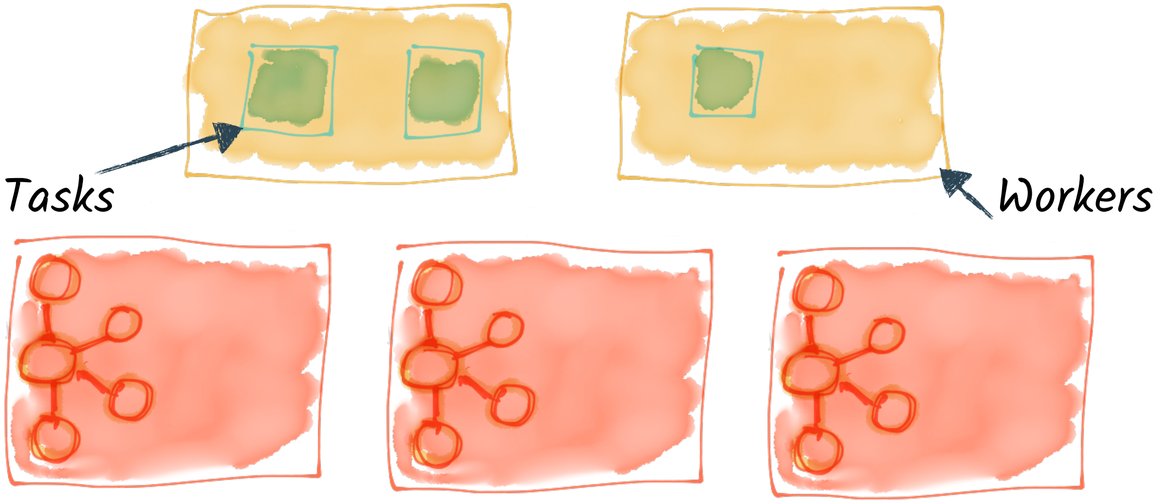
Let's Build It!



Kafka Connect



Streaming Integration with Kafka Connect



Kafka Connect

Kafka Brokers

Kafka Connect

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

```
{
"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/

Serialisation & Schemas

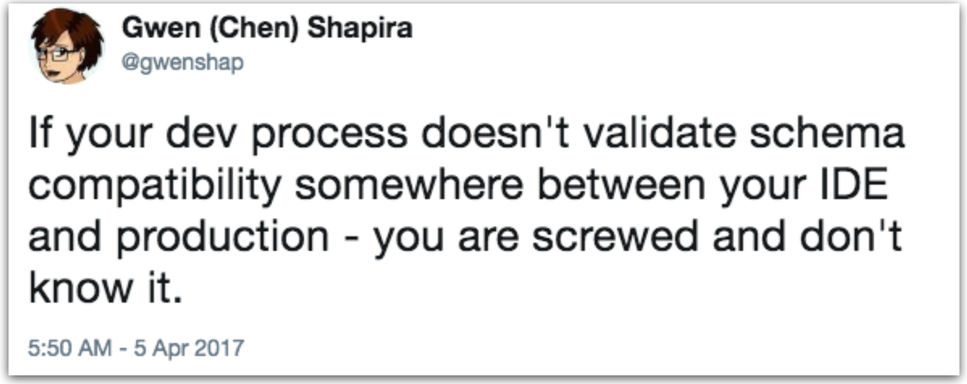
Avro

-> Confluent Schema Registry

Protobuf

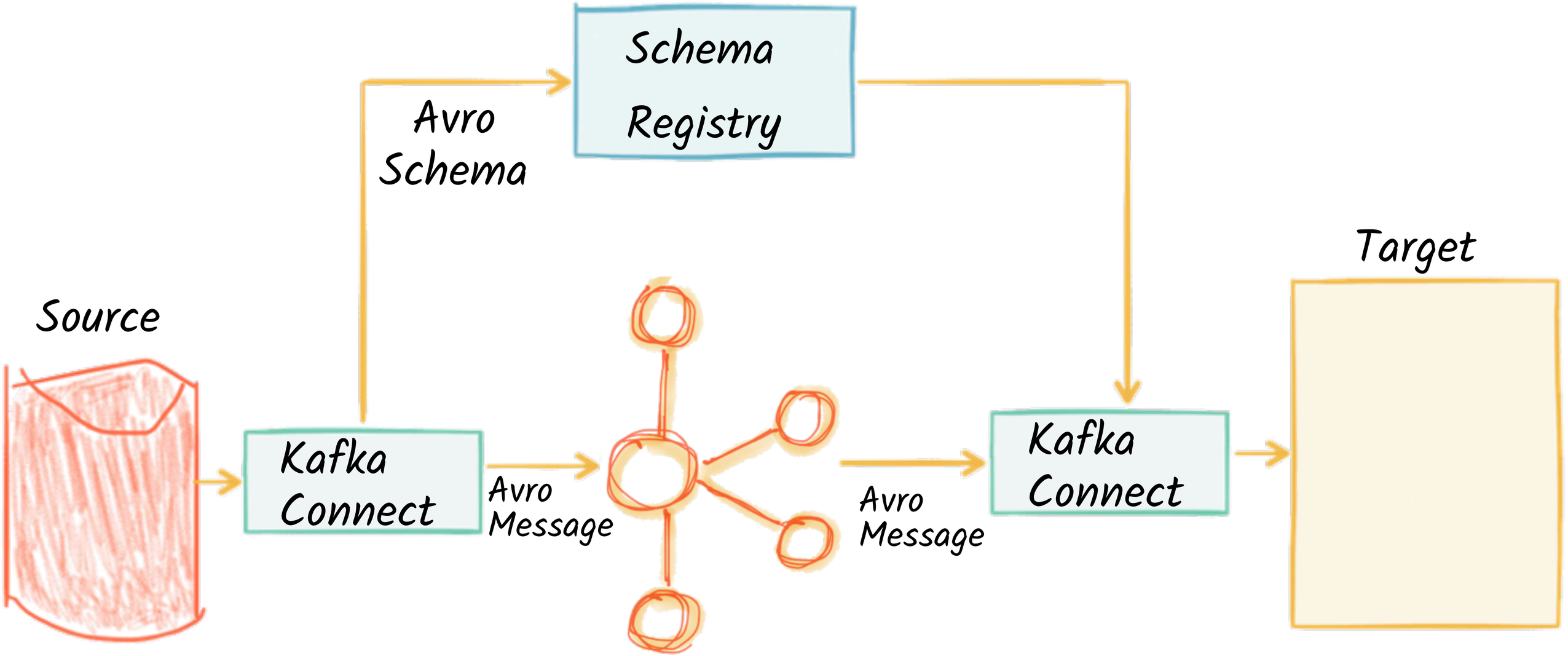
JSON

CSV



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

The Confluent Schema Registry



Confluent Hub

CONFLUENT HUB

Discover and share Connectors and more

Search Connectors

All Verified Sources Sinks Community

Confluent Supported

Debezium MongoDB CDC Connector
Debezium Community
Read More

Confluent Supported

Debezium MySQL CDC Connector
Debezium Community
Read More

Confluent Supported

Debezium PostgreSQL CDC Connector
Debezium Community
Read More

Confluent Supported

Debezium SQL Server CDC Connector
Debezium Community
Read More

hub.confluent.io

Change-Data-Capture (CDC)

- **CDC** is a generic term referring to *capturing changing data* typically from a RDBMS.
- Two general approaches:
  - **Query-based CDC**
  - **Log-based CDC**

There are other options including hacks with Triggers, Flashback etc but these are system and/or technology-specific.

Query-based CDC

- Use a database query to try and identify new & changed rows

```
SELECT * FROM my_table
WHERE col > <value of col last time we polled>
```

- Implemented with the open source **Kafka Connect JDBC connector**
  - Can import based on table names, schema, or bespoke SQL query
  - Incremental ingest driven through incrementing ID column and/or timestamp column

Log-based CDC

- Use the database's transaction log to identify every single change event
- Various CDC tools available that integrate with Apache Kafka (more of this later…)

```
Logdump 12 >pos 6636
Reading forward from RBA 6636
Logdump 13 >n
-------------------------------------------------------------------
Hdr-Ind    :     E  (x45)     Partition  :     .  (x0c)
UndoFlag   :     .  (x00)     BeforeAfter:     A  (x41)
RecLength  :   256  (x0100)   IO Time    : 2016/09/06 11:59:23.000.589
IOType     :     5  (x05)     OrigNode   :   255  (xff)
TransInd   :     .  (x00)     FormatType :     R  (x52)
SyskeyLen  :     0  (x00)     Incomplete :     .  (x00)
AuditRBA   :        393       AuditPos   : 30266384
Continued  :     N  (x00)     RecCount   :     1  (x01)

2016/09/06 11:59:23.000.589 Insert               Len   256 RBA 6636
Name: ORCL.SOE.CUSTOMERS  (TDR Index: 3)
After  Image:                                          Partition 12   G  b
0000 000a 0000 0000 0000 0001 86a1 0001 000a 0000 | ....................
0006 616e 7477 616e 0002 000b 0000 0007 7361 6d70 | ..antwan........samp
736f 6e00 0300 0600 0000 0275 7300 0400 0b00 0000 | son........us.......
0741 4d45 5249 4341 0005 000a 0000 0000 0000 0000 | .AMERICA............
8980 0006 001d 0000 0019 616e 7477 616e 2e73 616d | ..........antwan.sam
7073 6f6e 406f 7261 636c 652e 636f 6d00 0700 0a00 | pson@oracle.com.....
0000 0000 0000 0000 9500 0800 1500 0032 3031 362d | ................2016-
Column     0 (x0000), Len    10 (x000a)
0000 0000 0000 0001 86a1                          | ..........
Column     1 (x0001), Len    10 (x000a)
0000 0006 616e 7477 616e                          | ....antwan
Column     2 (x0002), Len    11 (x000b)
0000 0007 7361 6d70 736f 6e                       | ....sampson
Column     3 (x0003), Len     6 (x0006)
0000 0002 7573                                    | ....us
Column     4 (x0004), Len    11 (x000b)
0000 0007 414d 4552 4943 41                       | ....AMERICA
Column     5 (x0005), Len    10 (x000a)
0000 0000 0000 0000 8980                          | ..........
Column     6 (x0006), Len    29 (x001d)
0000 0019 616e 7477 616e 2e73 616d 7073 6f6e 406f | ....antwan.sampson@o
7261 636c 652e 636f 6d                            | racle.com
Column     7 (x0007), Len    10 (x000a)
0000 0000 0000 0000 0095                          | ..........
Column     8 (x0008), Len    21 (x0015)
```

Query-based vs Log-based CDC

- **Query-based**
  - +Usually easier to setup, and requires fewer permissions
  - - Needs specific columns in source schema
  - - Impact of polling the DB (or higher latencies tradeoff)
  - - Can't track deletes, or multiple events between polling interval

Photo by Matese Fields on Unsplash

Read more: http://cnfl.io/kafka-cdc

Query-based vs Log-based CDC

Photo by Sebastian Pociecha on Unsplash

- **Log-based**
  - +Greater data fidelity
  - +Lower latency
  - +Lower impact on source
  - - More setup steps
  - - Higher system privileges required
  - - For propriatory databases, usually $$$

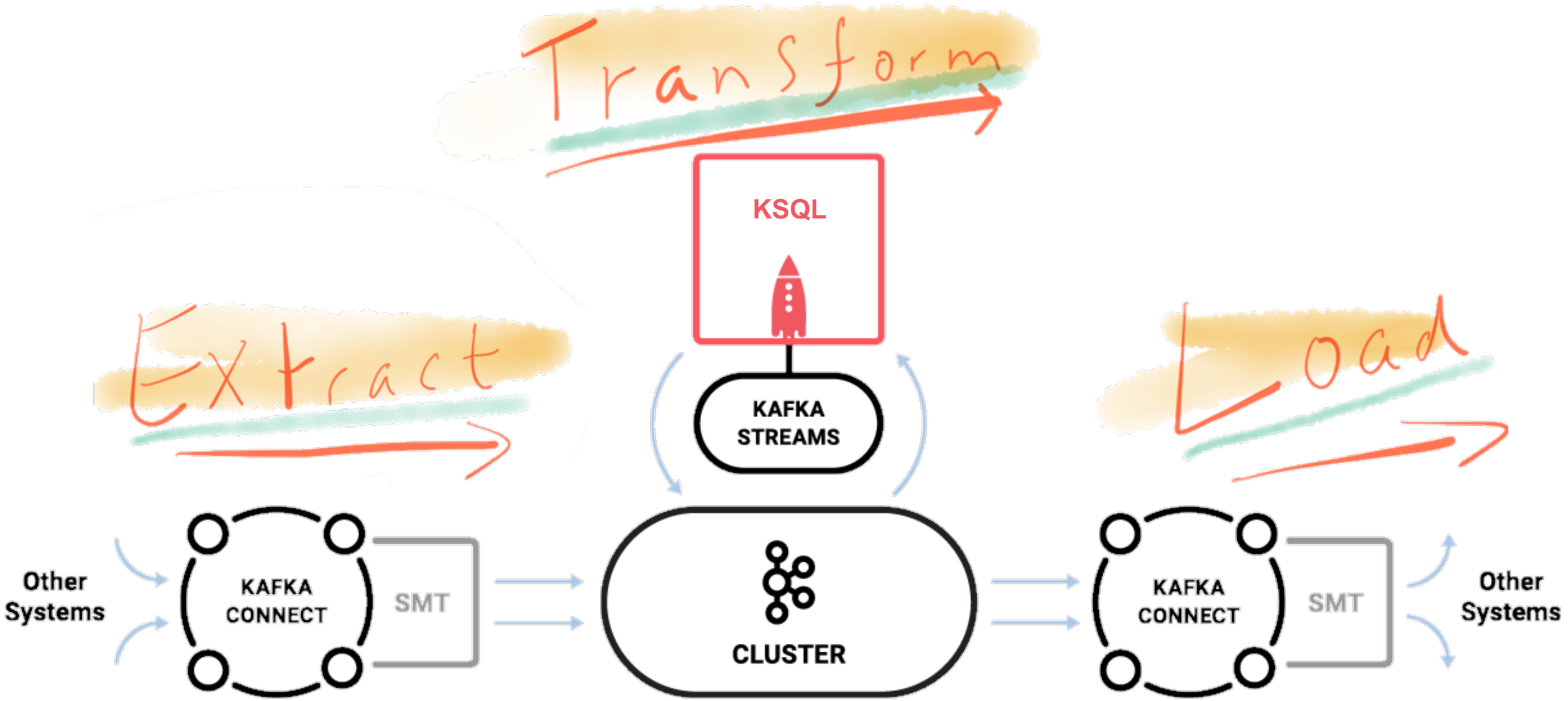
Read more: http://cnfl.io/kafka-cdc

Oracle and Kafka integration

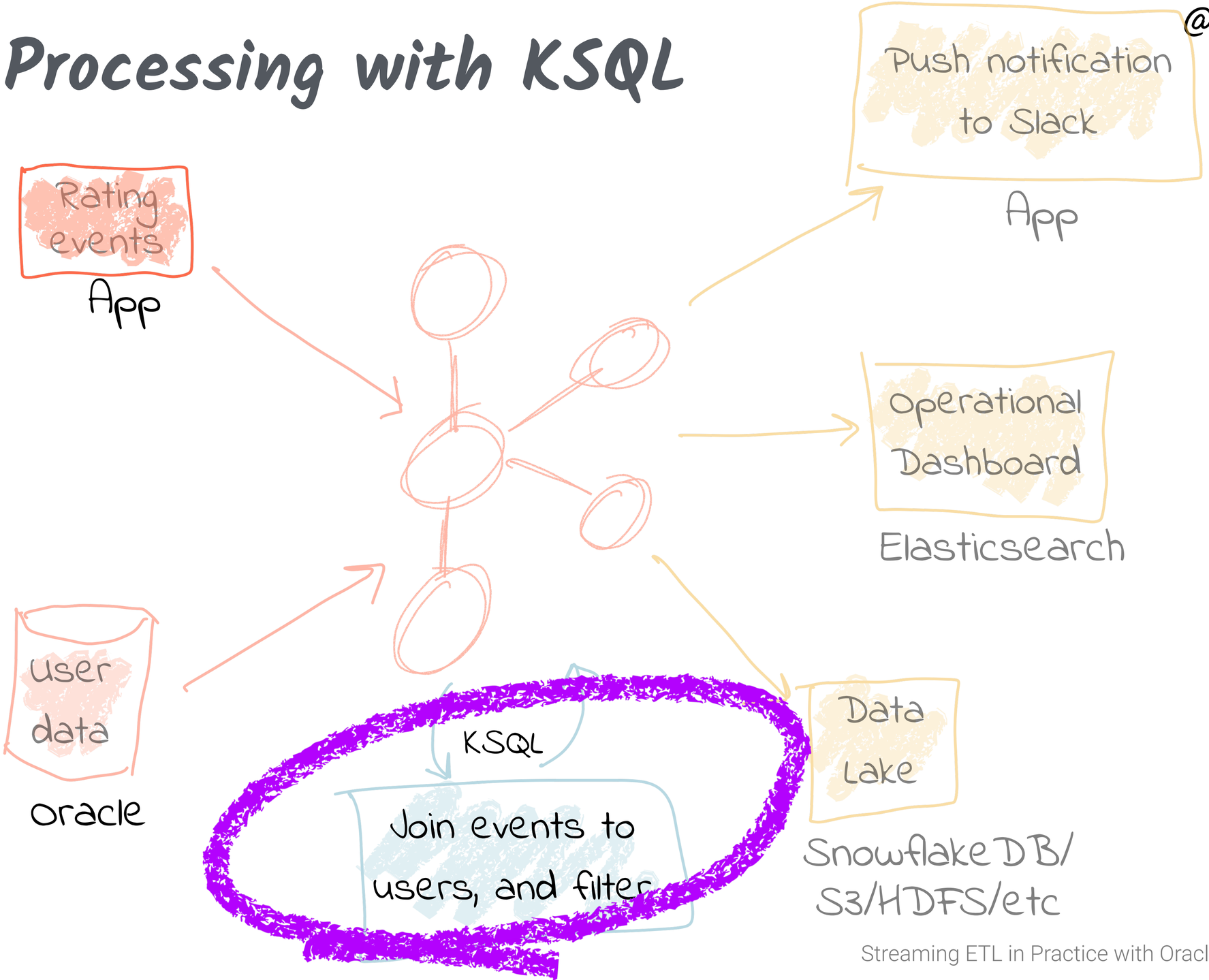
- **Oracle GoldenGate for Big Data**—Requires the OGGBD licence, not just OGG
- **Debezium**—Open source, Oracle support in Beta
  - currently uses XStream—which requires OGG licence
- **Attunity, IBM IIDR, HVR, SQData, tcVision, StreamSets**—all offer commerical CDC integration into Kafka with support for Schema Registry
  - **DBVisit Replicate**—no longer under development
- **JDBC Connector**—Open source, but not "true" CDC

https://rmoff.net/2018/12/12/streaming-data-from-oracle-into-kafka-december-2018/

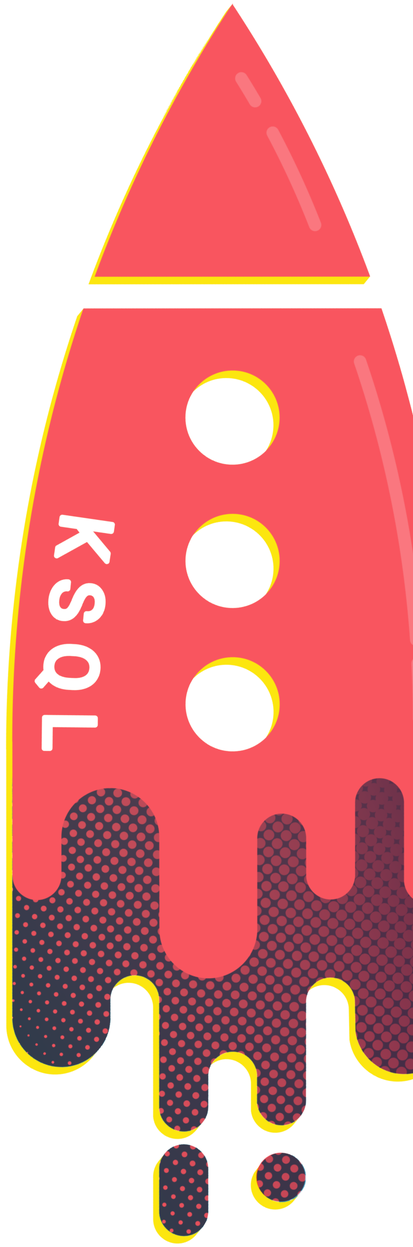
Streaming ETL, powered by Apache Kafka and Confluent Platform



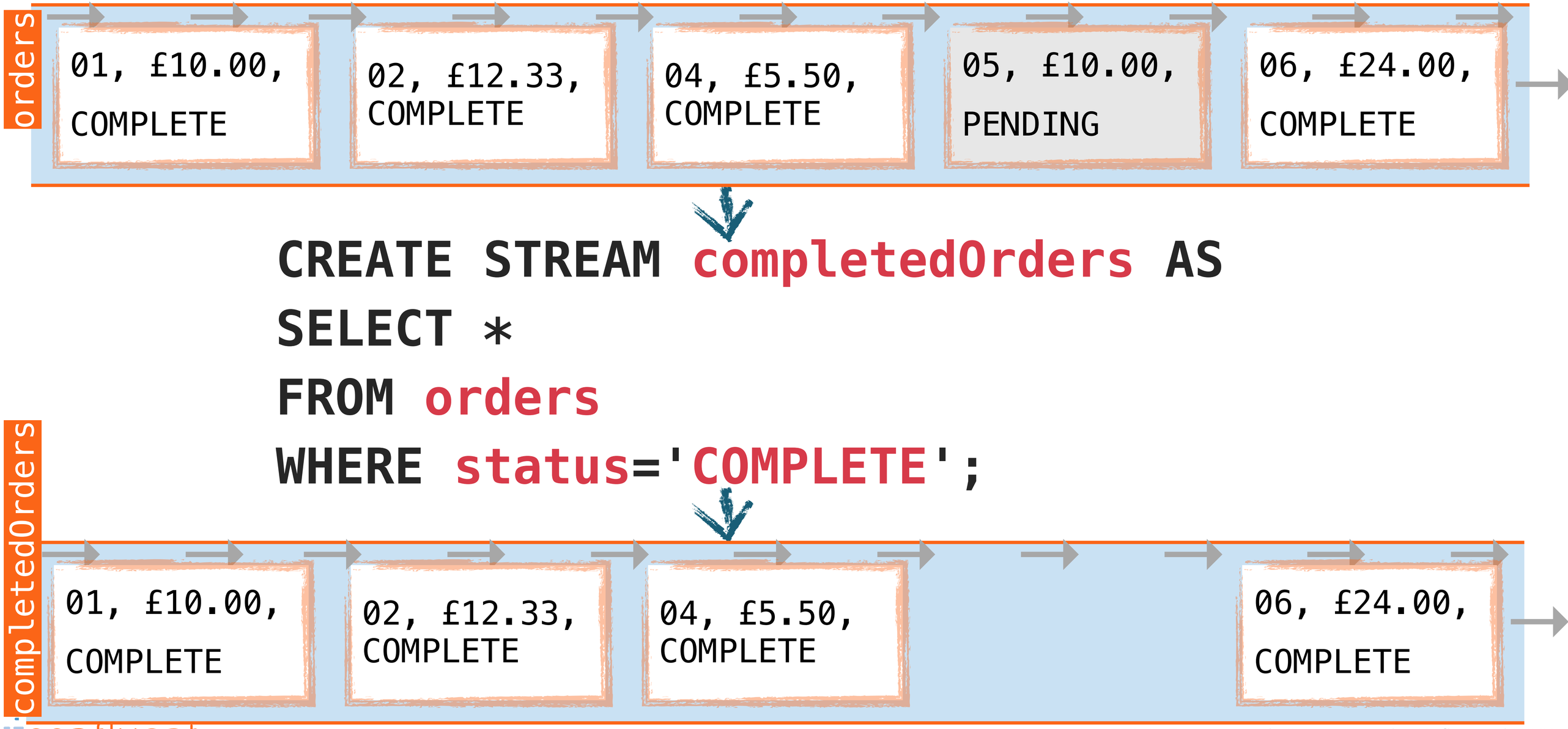
Stream Processing with KSQL



KSQL
is the
Streaming
SQL Engine
for
Apache Kafka



Filter messages with KSQL




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

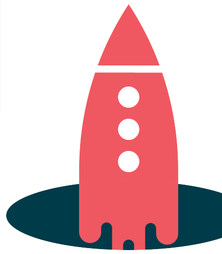
```
{"country":"UK",
"customerCount":2}
```

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

KSQL for Anomaly Detection

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

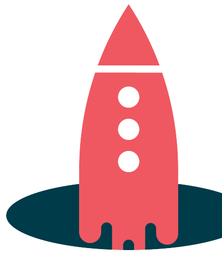
```
CREATE TABLE possible_fraud AS
  SELECT card_number, count(*)
    FROM authorization_attempts
    WINDOW TUMBLING (SIZE 5 SECONDS)
    GROUP BY card_number
    HAVING count(*) > 3;
```



KSQL for Data Transformation

Make simple derivations of existing topics from the command line

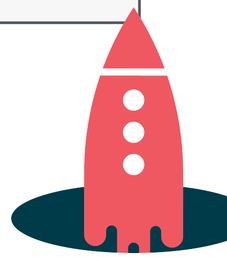
```
CREATE STREAM pageviews
  WITH (PARTITIONS=4,
        VALUE_FORMAT='AVRO') AS
  SELECT * FROM pageviews_json;
```



KSQL for Streaming ETL

Joining, filtering, and aggregating streams of event data

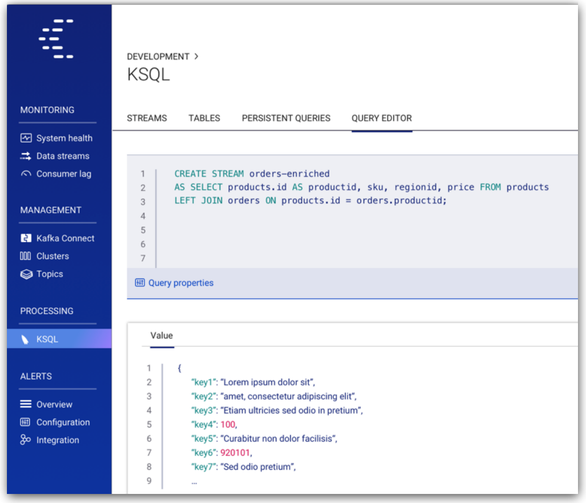
```
CREATE STREAM vip_actions AS
  SELECT userid, page, action
  FROM clickstream c
  LEFT JOIN users u
    ON c.userid = u.user_id
  WHERE u.level = 'Platinum';
```



KSQL in Development and Production

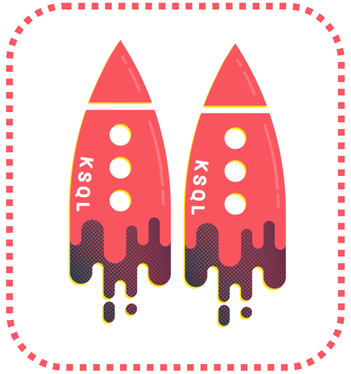
Interactive KSQL
for development and testing

Headless KSQL
for Production

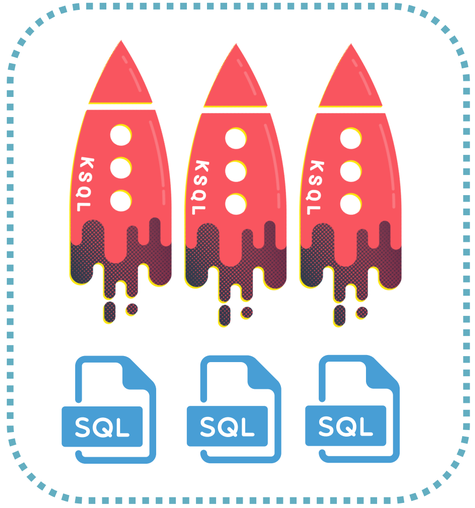


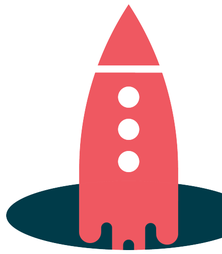
"Hmm, let me try out this idea..."

REST

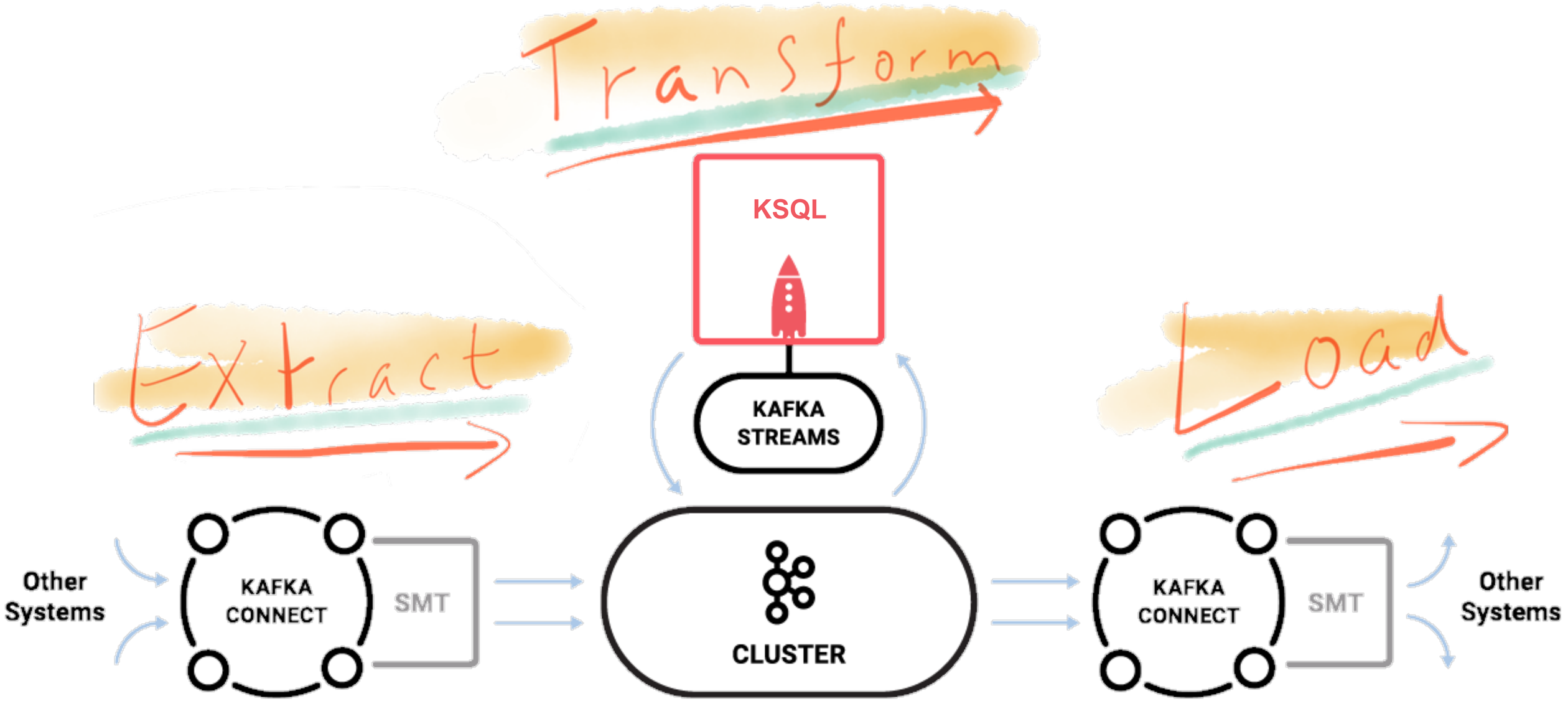


Desired KSQL queries have been identified

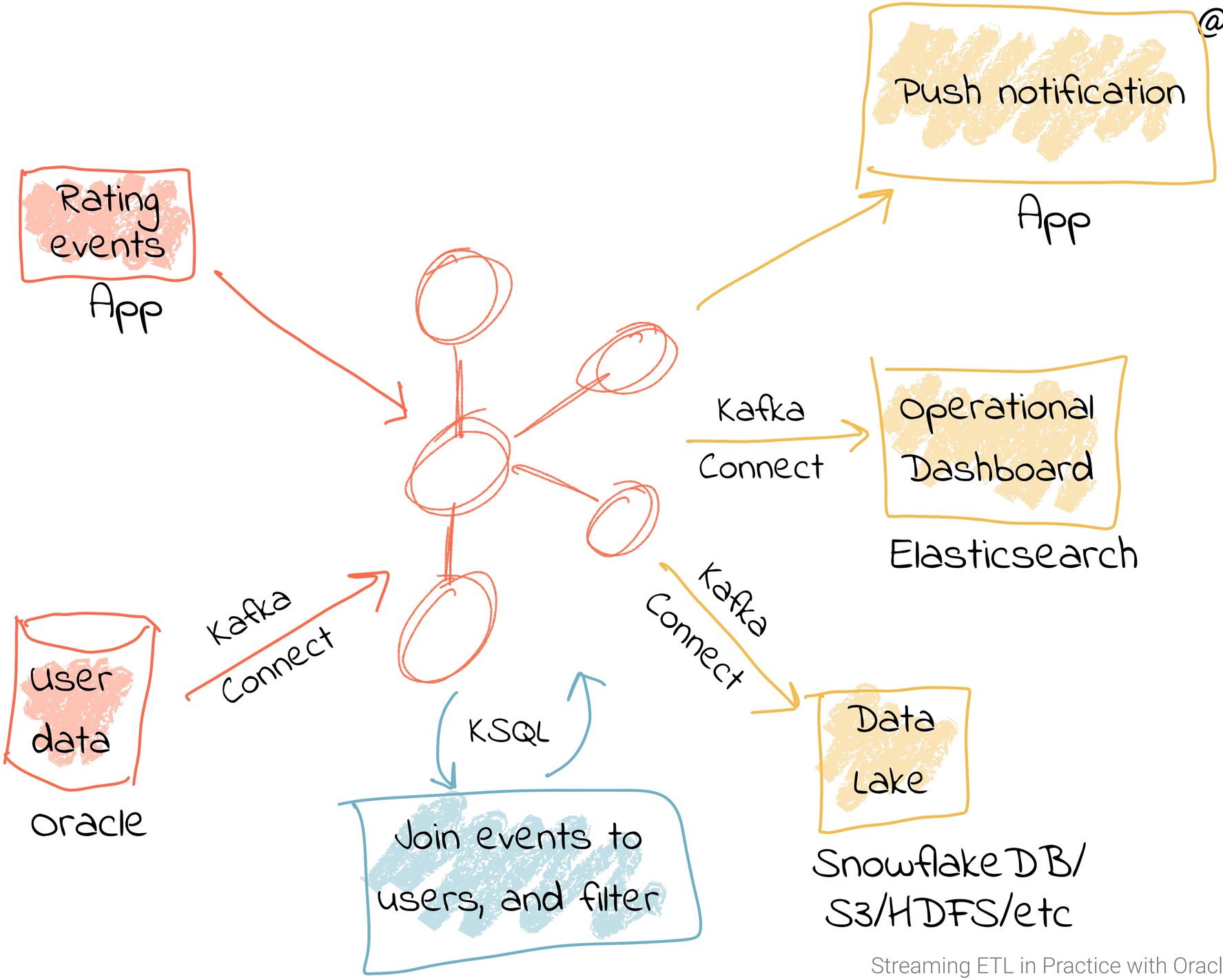




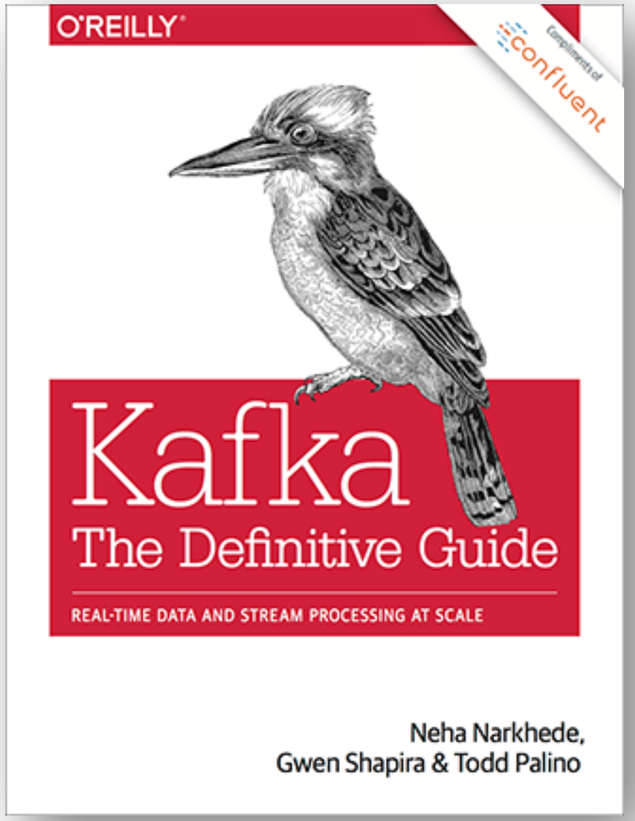
Streaming ETL, powered by Apache Kafka and Confluent Platform

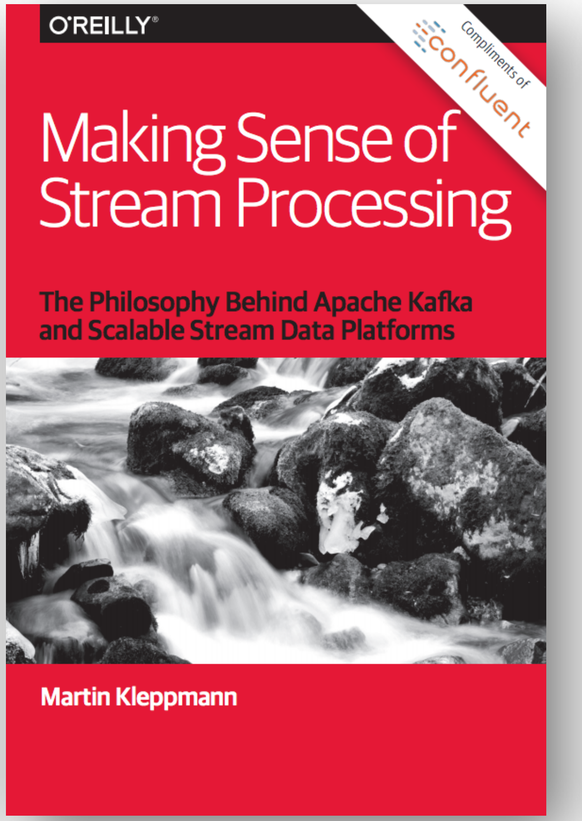


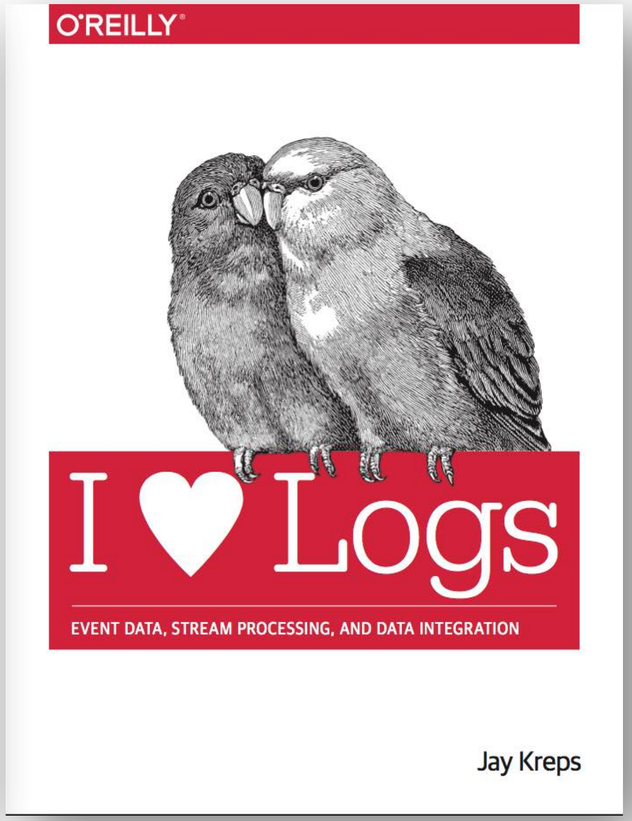
Demo Time!

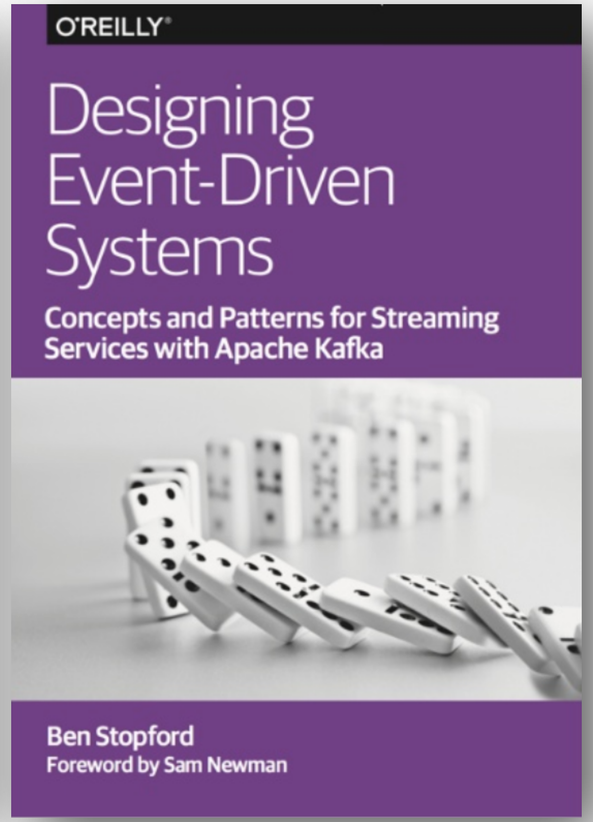

Push notification
App
Rating events
App
Kafka Connect
Operational Dashboard
Elasticsearch
Kafka Connect
Kafka Connect
User data
Oracle
KSQL
Join events to users, and filter
Data Lake
SnowflakeDB/ S3/HDFS/etc

http://cnfl.io/book-bundle









NEXT STOP
SAN FRANCISCO
SEPT 30-OCT 1

CONFLUENT COMMUNITY DISCOUNT CODE

KS19Meetup

25% OFF*

kafka summit

*Standard Priced Conference pass

https://www.confluent.io/download/

http://cnfl.io/kafka-cdc

http://cnfl.io/slack

@rmoff robin@confluent.io

#EOF