@GAMUSSA #POSTGRES @CONFLUENTINC

STREAMING ETL IN PRACTICE: BUILD DATA PIPELINES WITHOUT A SINGLE LINE OF CODE!

@GAMUSSA

Special Thanks!

@RMOFF

confluent

RAFFLE, YEAH 🚀

- must follow @gamussa @confluentinc
- 📸🖼️🐘
- Tag @gamussa
- With #peoplepostgresdata

APACHE kafka

confluent

@gamussa #Postgres @confluentinc

confluent

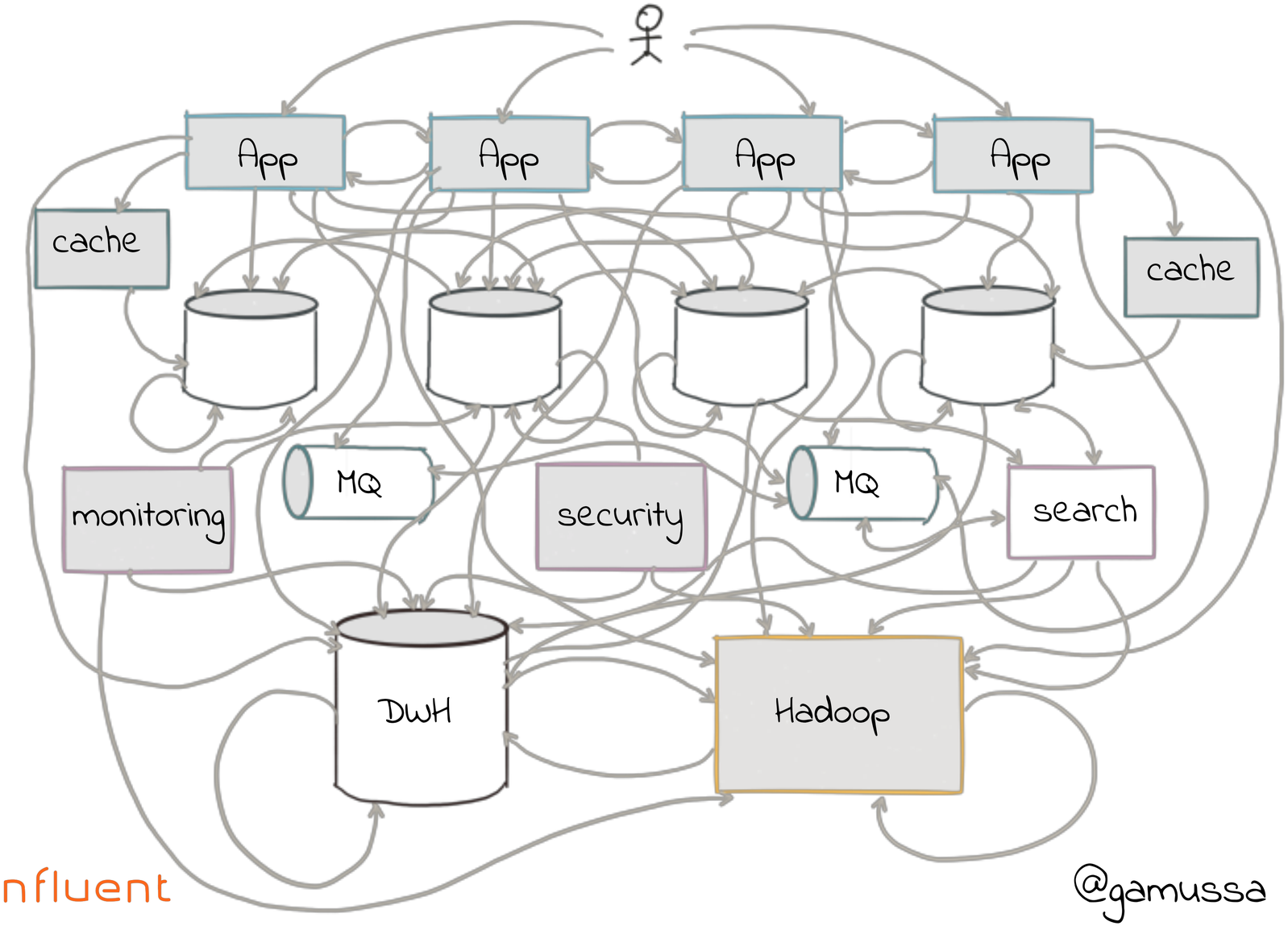
APACHE KAFKA IS

AN EVENT STREAMING PLATFORM

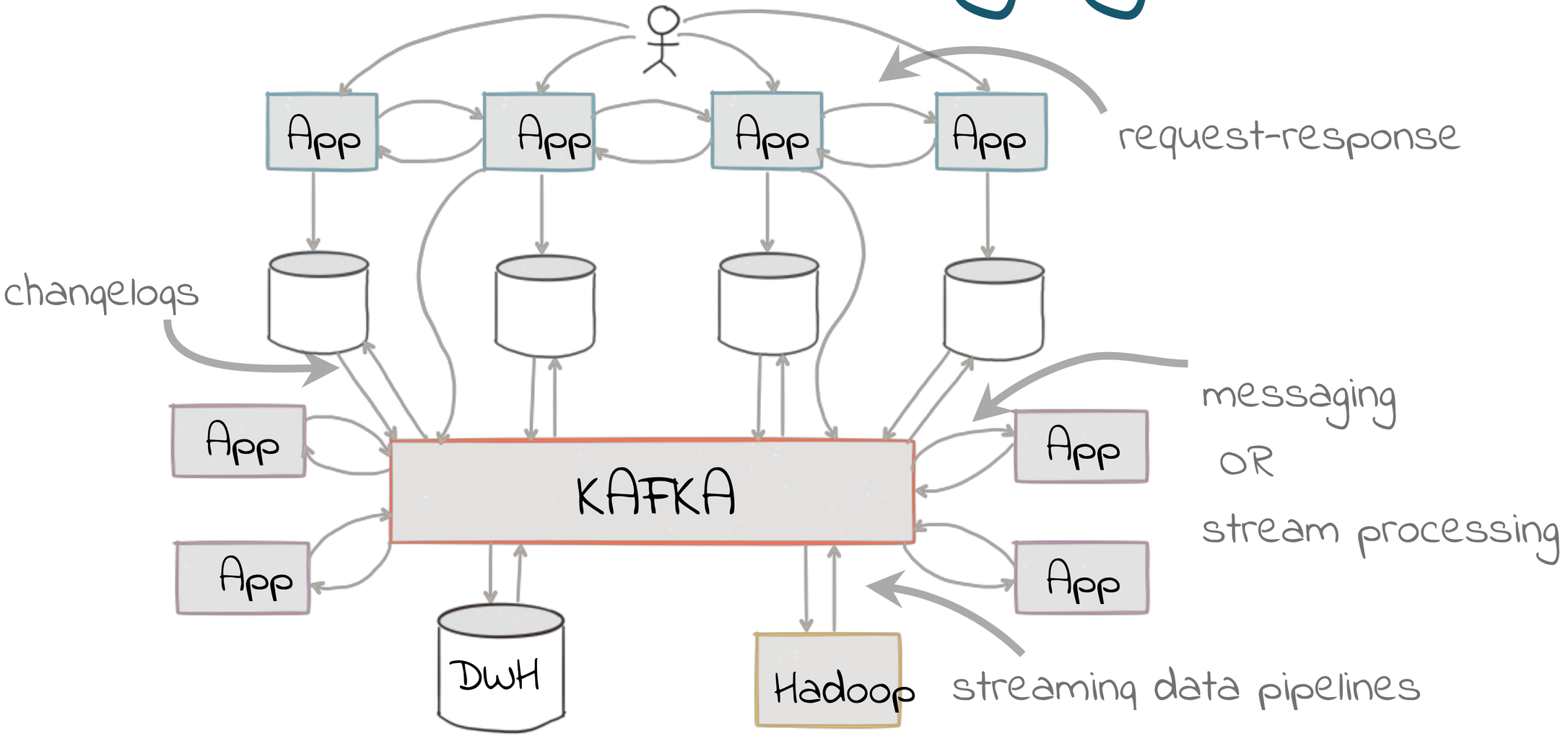
BUT WHAT IS

AN EVENT STREAMING PLATFORM?

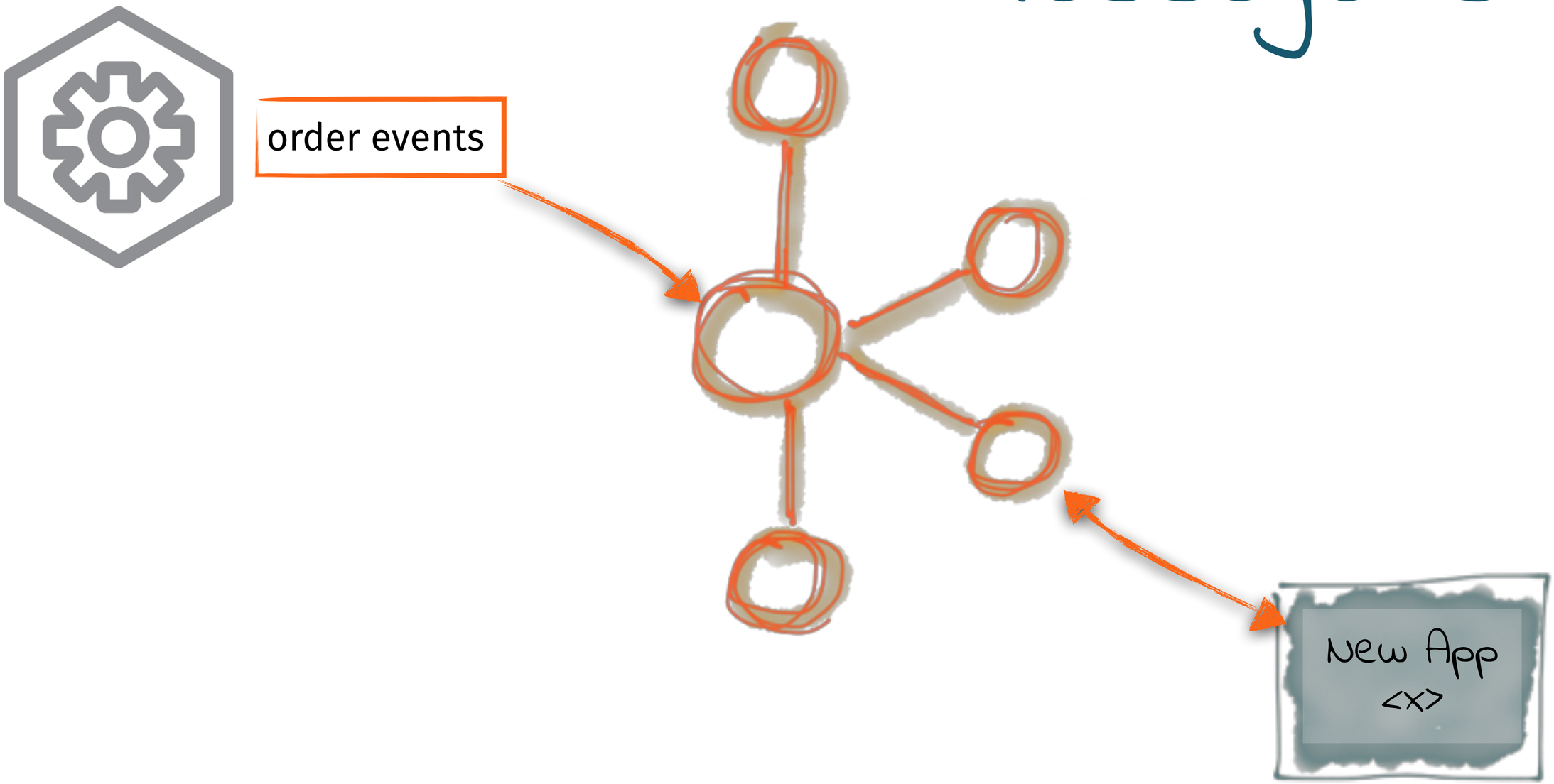
A bit of a mess...



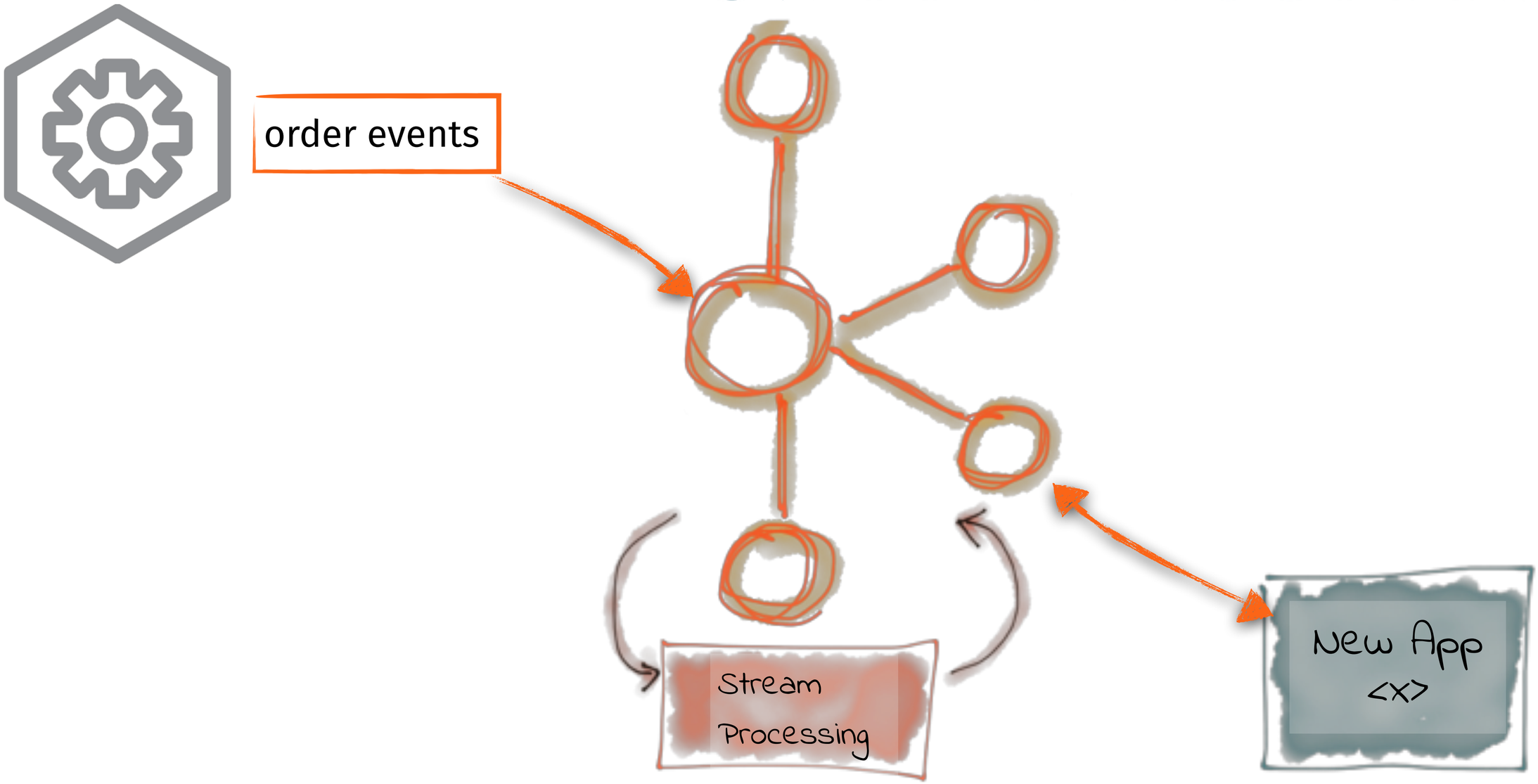
Streams are changing all of this



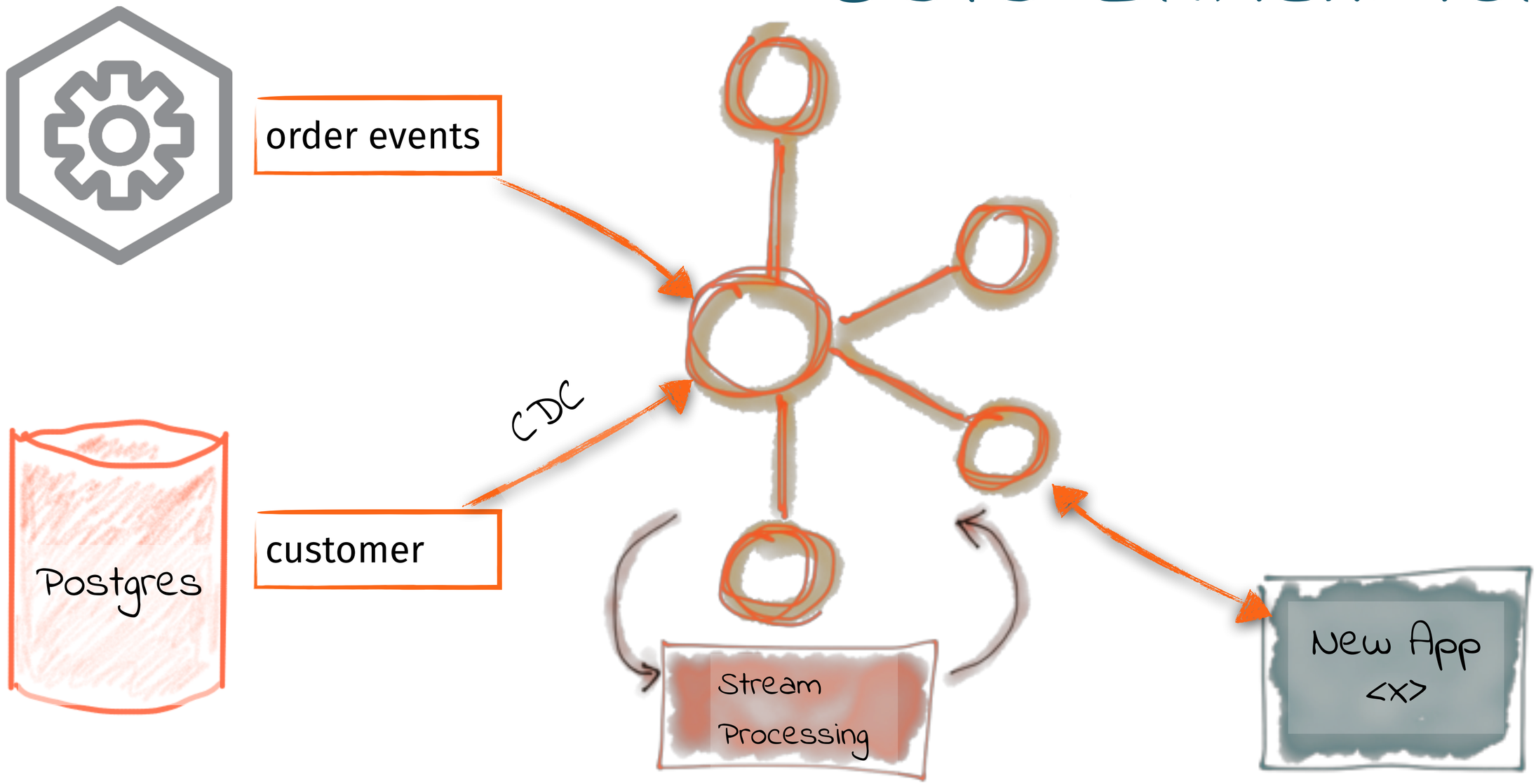
Message Bus



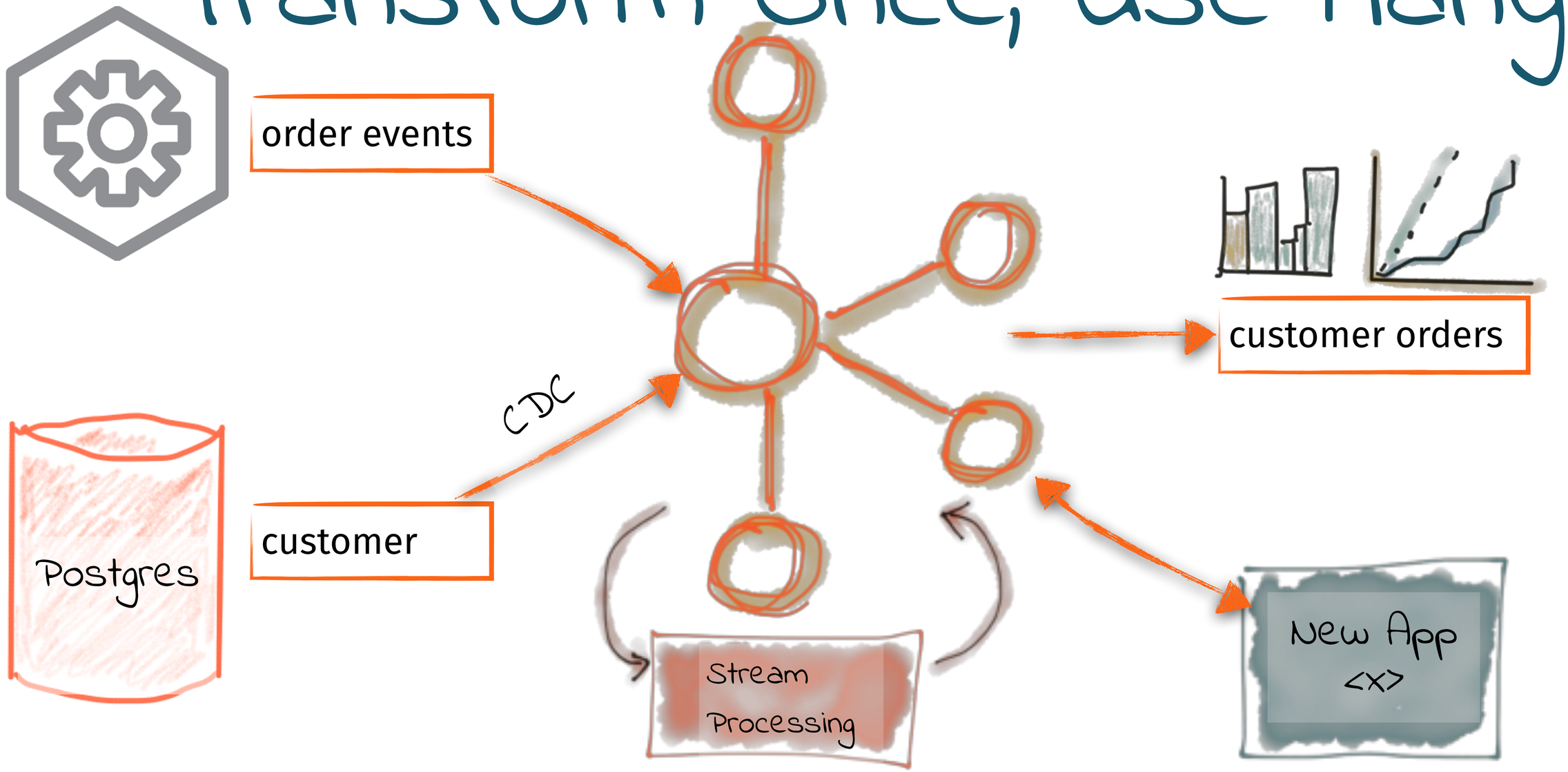
Stream Processing



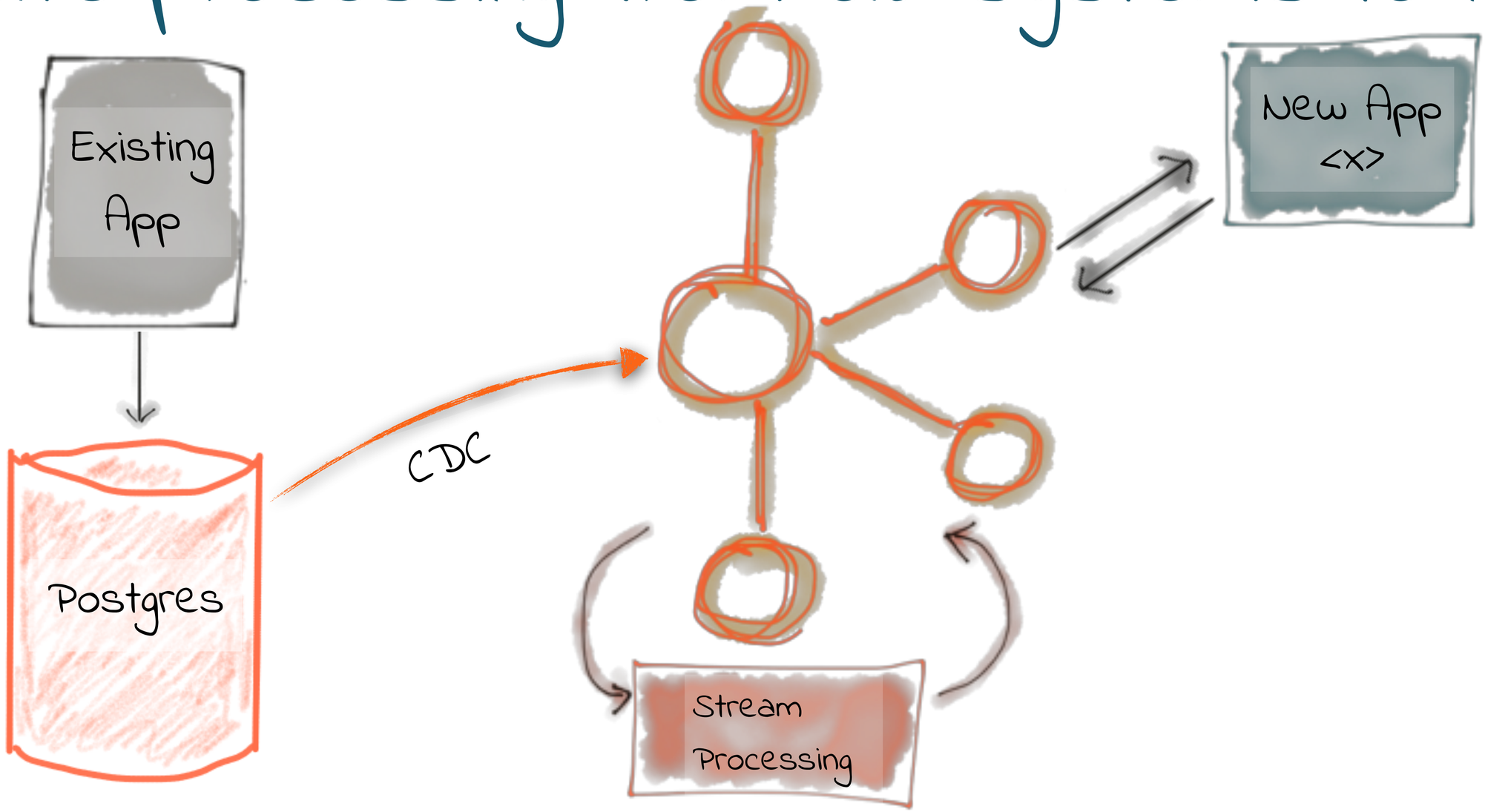
Data Enrichment



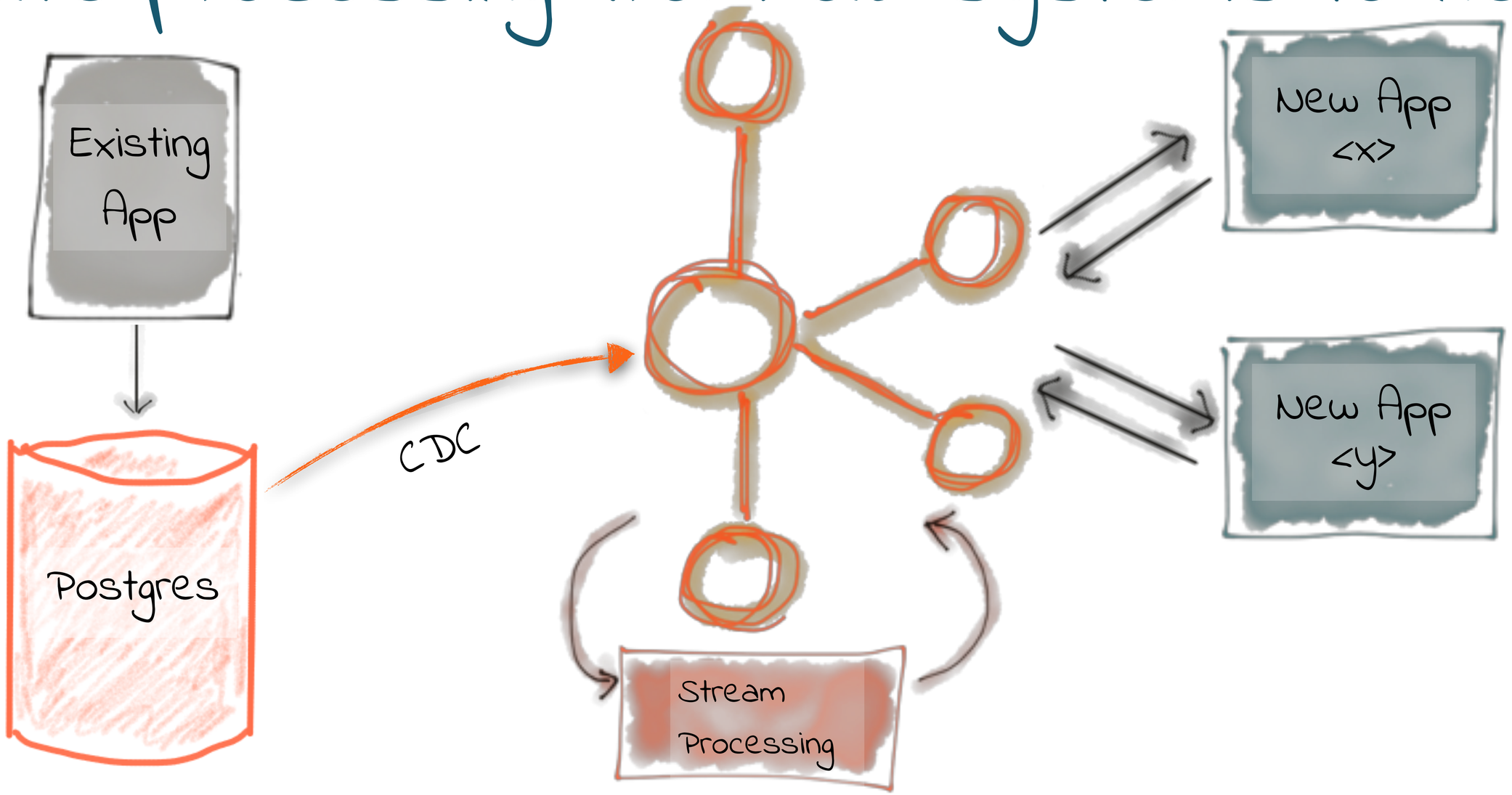
Transform once, use Many



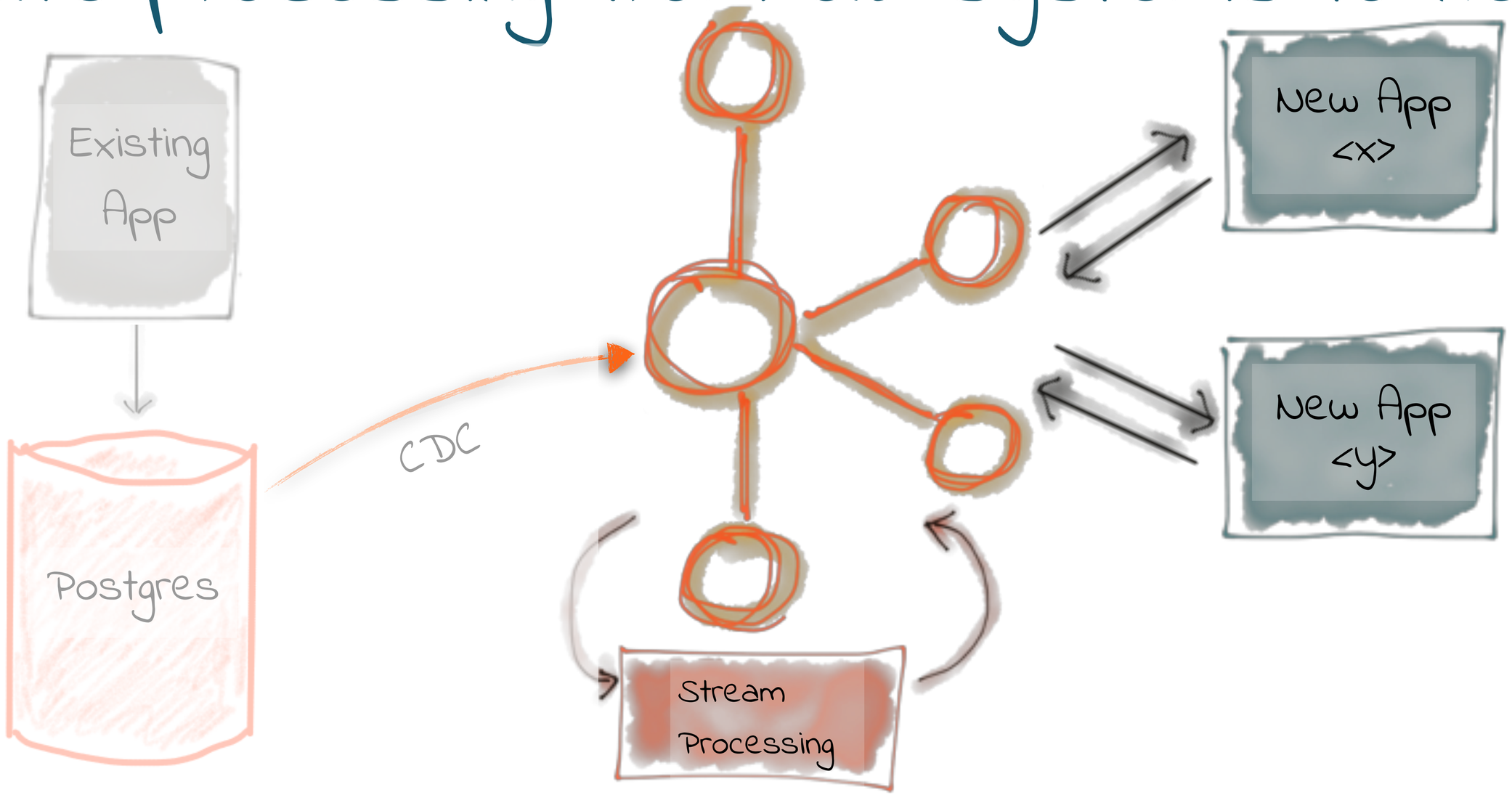
Evolve processing from old systems to new

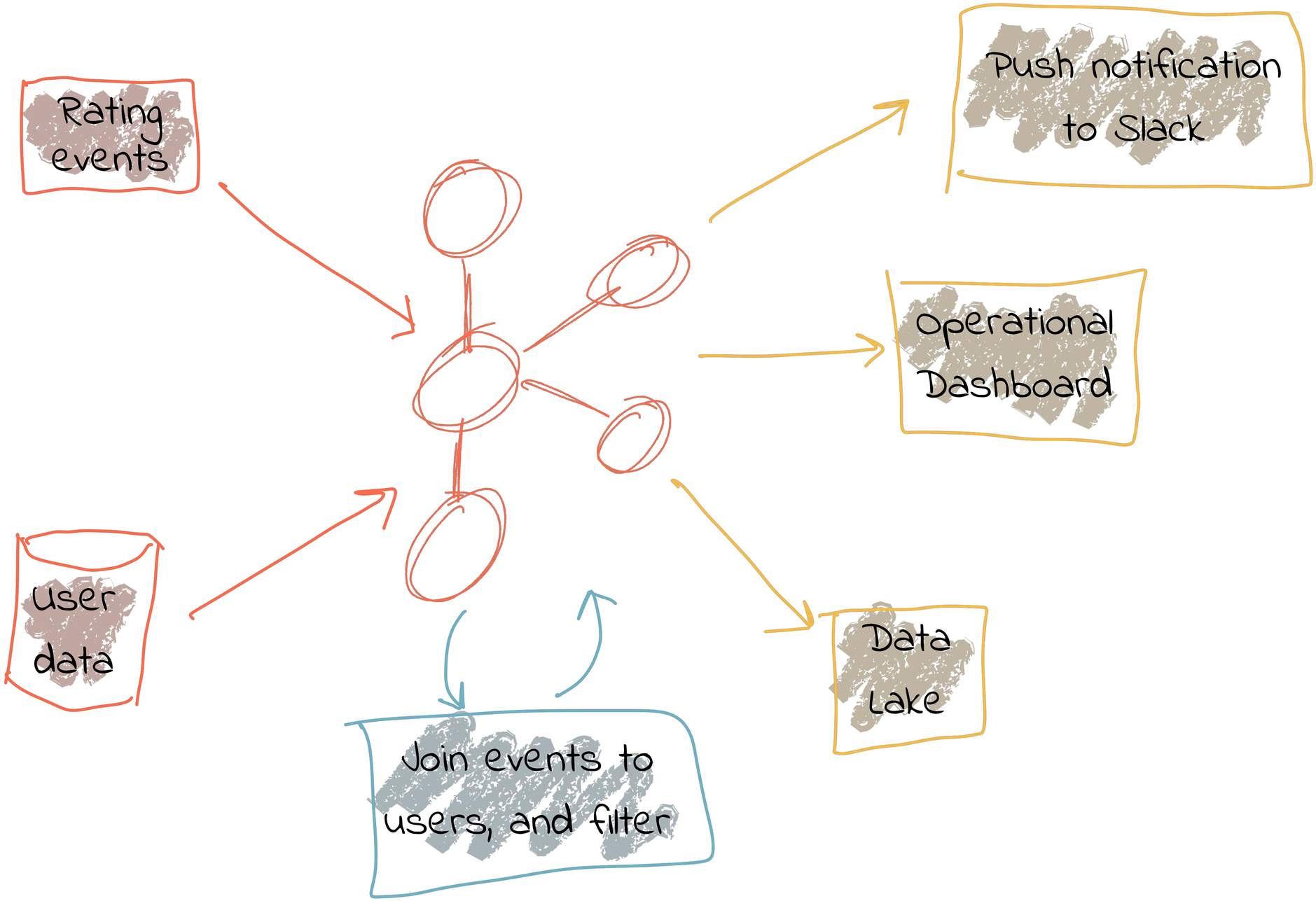


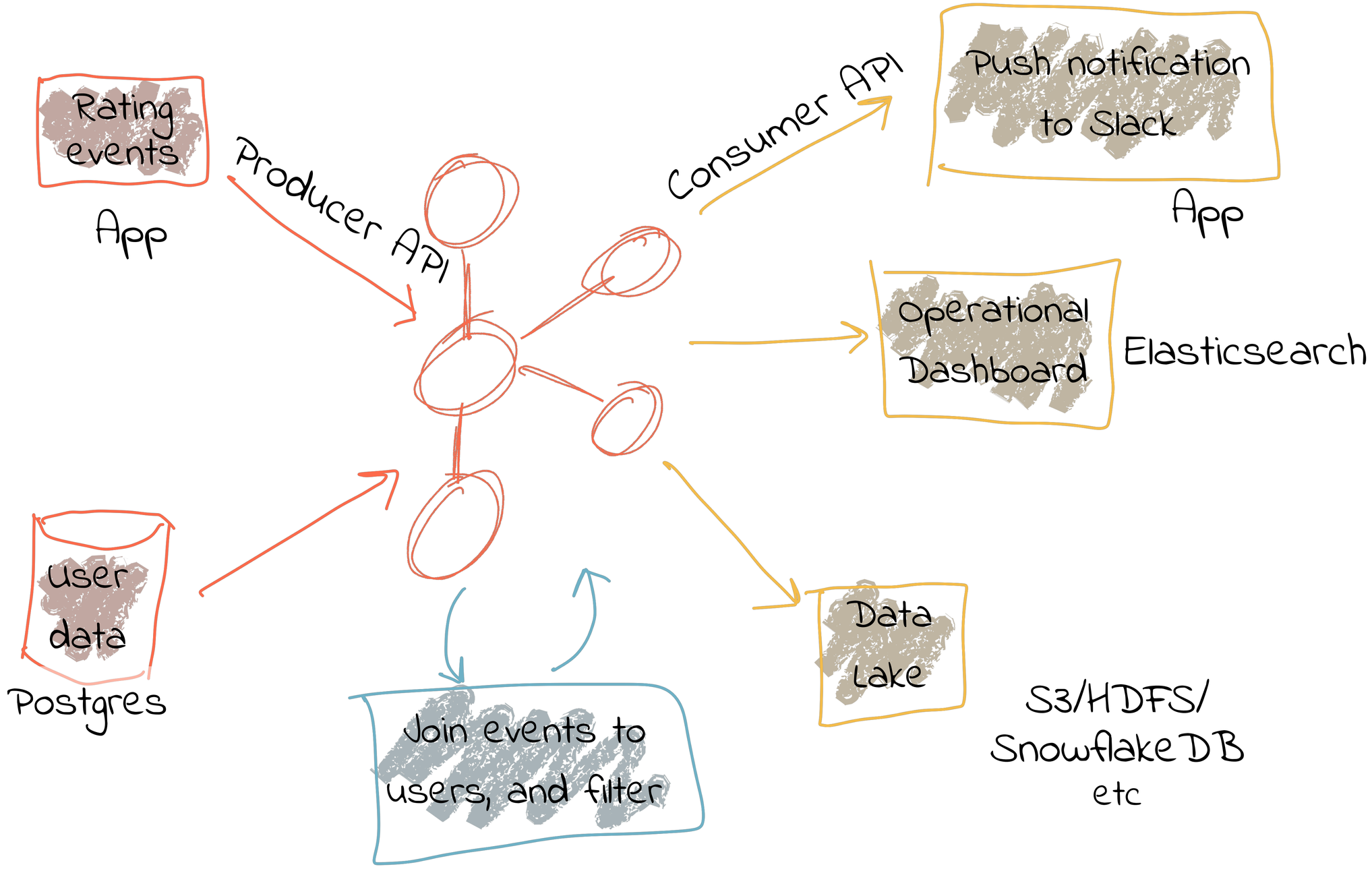
Evolve processing from old systems to new

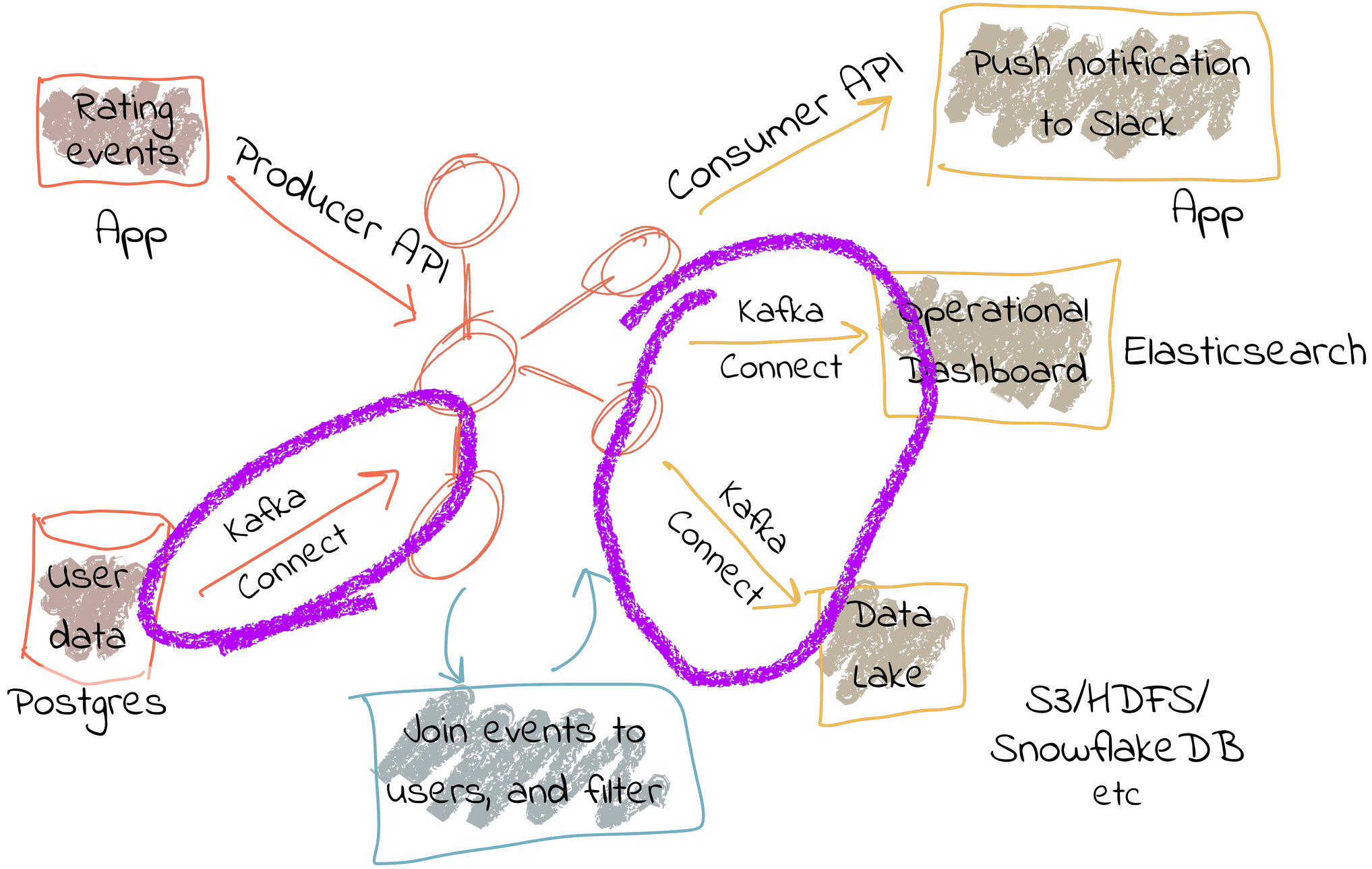


Evolve processing from old systems to new



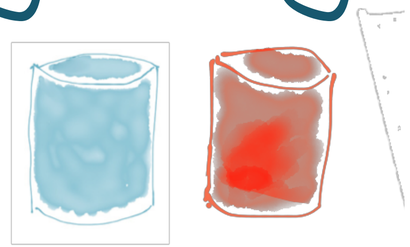

Rating events
Push notification to Slack
Operational Dashboard
User data
Data Lake
Join events to users, and filter


Rating events
App
Producer API
Consumer API
Push notification to Slack
App
Operational Dashboard
Elasticsearch
user data
Postgres
Join events to users, and filter
Data Lake
S3/HDFS/
Snowflake DB
etc

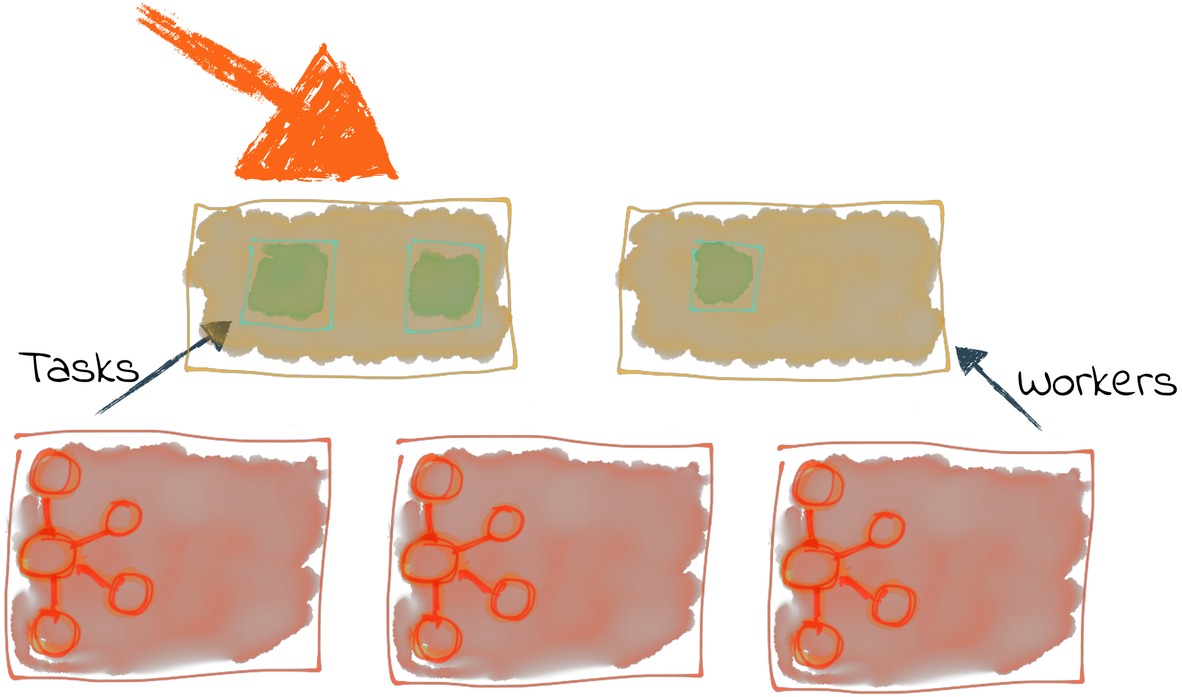

Rating events
App
Producer API
Consumer API
Push notification to Slack
App
Kafka Connect
Operational Dashboard
Elasticsearch
Kafka Connect
user data
Postgres
Kafka Connect
Data Lake
S3/HDFS/ Snowflake DB etc
Join events to users, and filter

Streaming Integration with Kafka Connect

ATTUNITY
ORACLE
Microsoft SQL Server
MySQL
syslog
flat file
CSV
JSON
MQTT



Sources



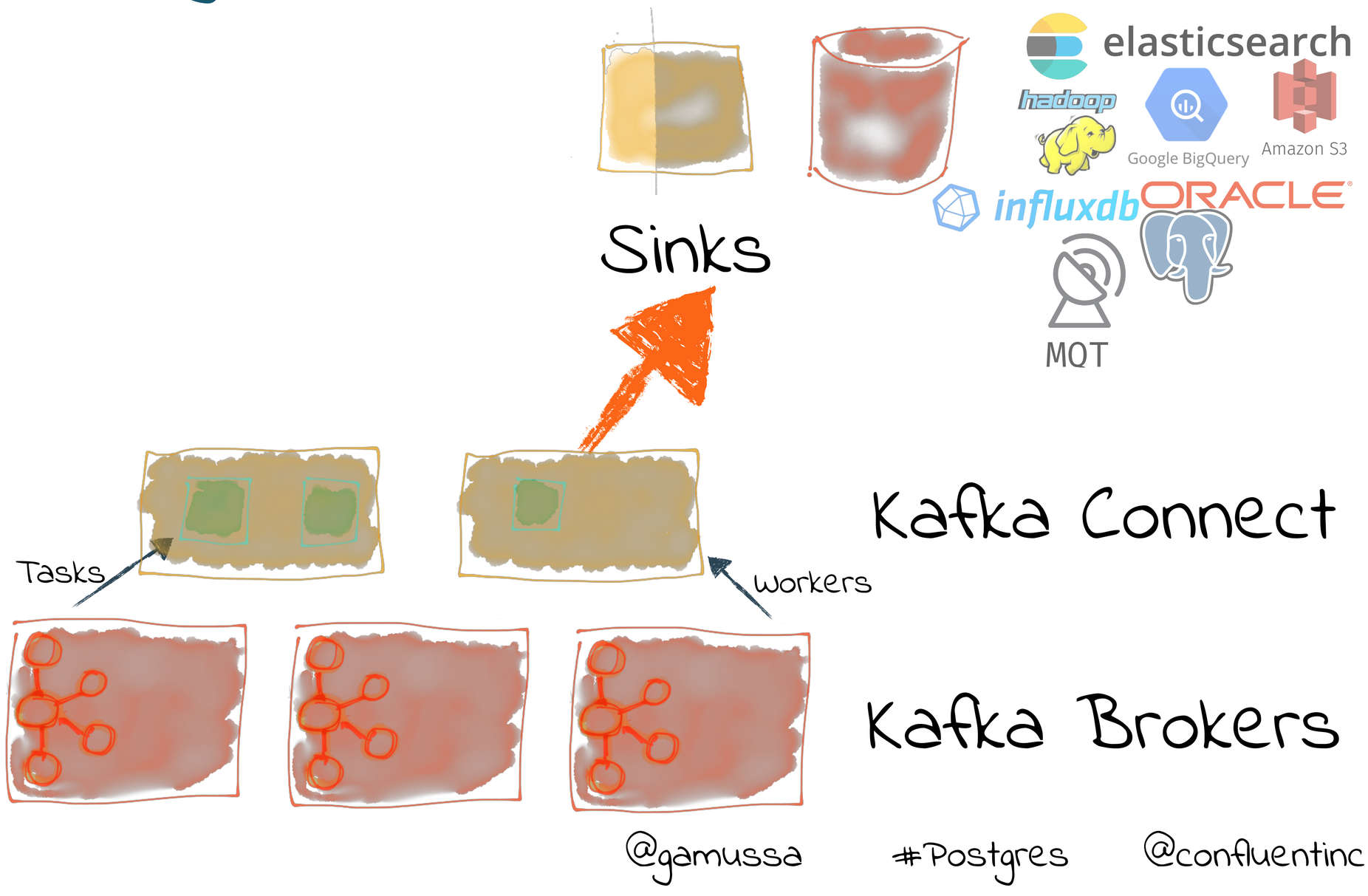
Kafka Connect

Kafka Brokers

confluent

@gamussa #Postgres @confluentinc

Streaming Integration with Kafka Connect



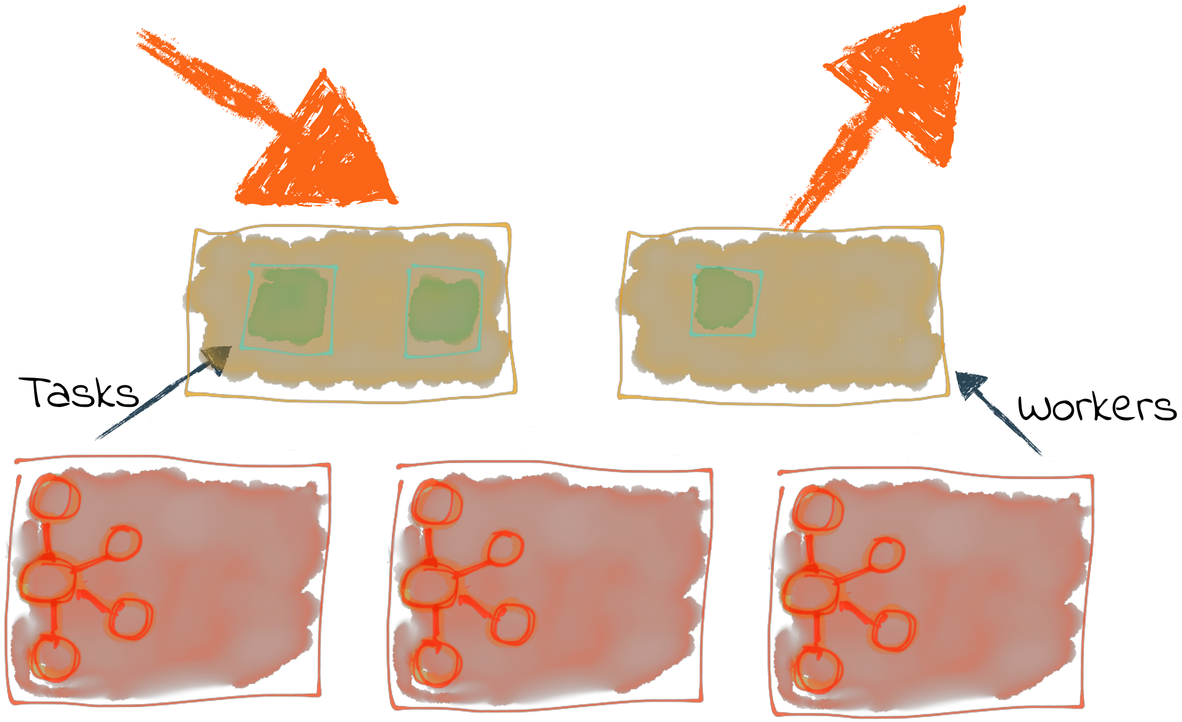
Streaming Integration with Kafka Connect

ATTUNITY
ORACLE
Microsoft SQL Server
MySQL
syslog
flat file
CSV
JSON
MQTT

elasticsearch
hadoop
Google BigQuery
Amazon S3
influxdb
ORACLE
MQT

Sources

Sinks



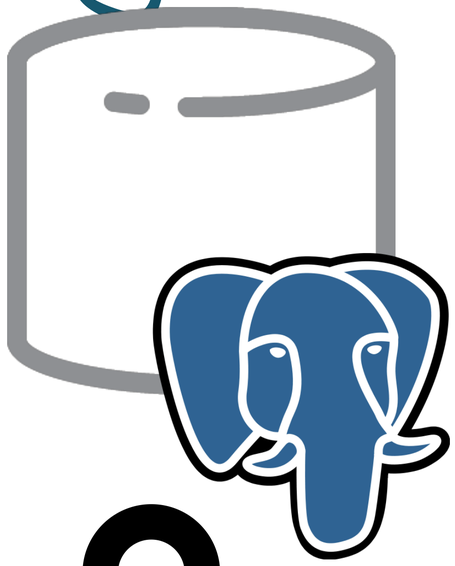
Kafka Connect

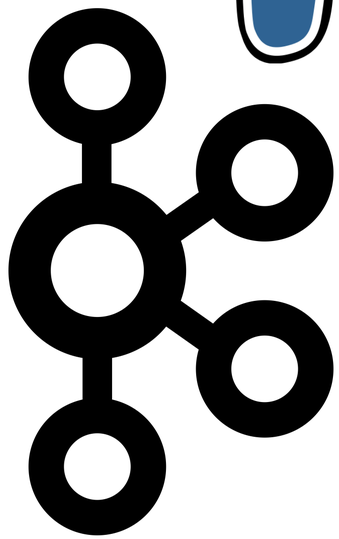
Kafka Brokers

confluent

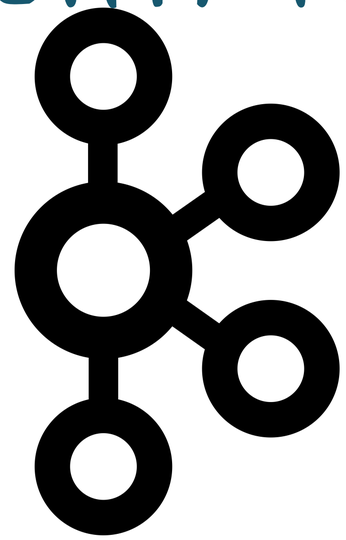
@gamussa #Postgres @confluentinc

Integrating Postgres with Kafka

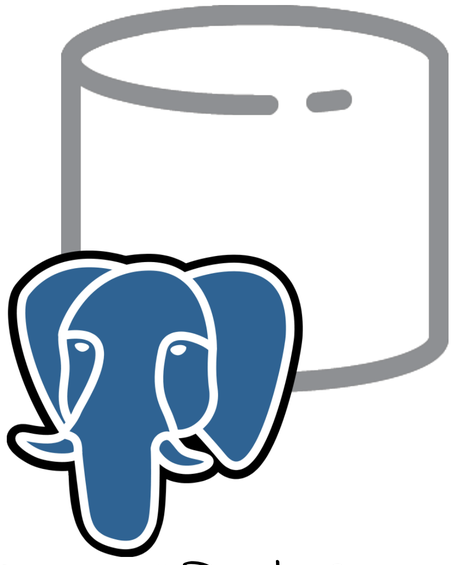




Kafka Connect &
Debezium

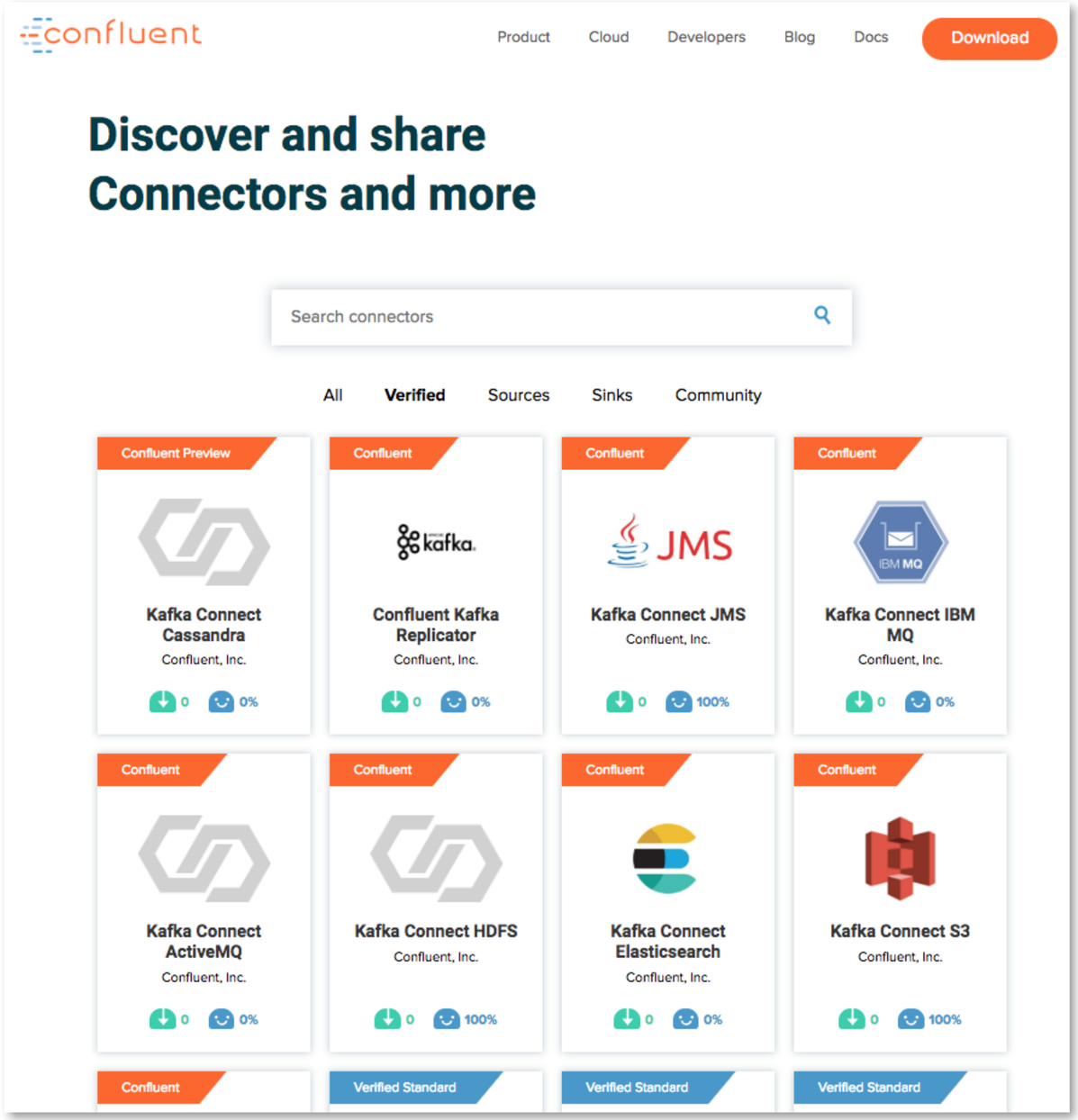


Kafka Connect &
JDBC Sink



confluent

@gamussa #Postgres @confluentinc



Confluent Hub

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

hub.confluent.io

confluent

@gamussa #Postgres @confluentinc

Stop! Demo time!

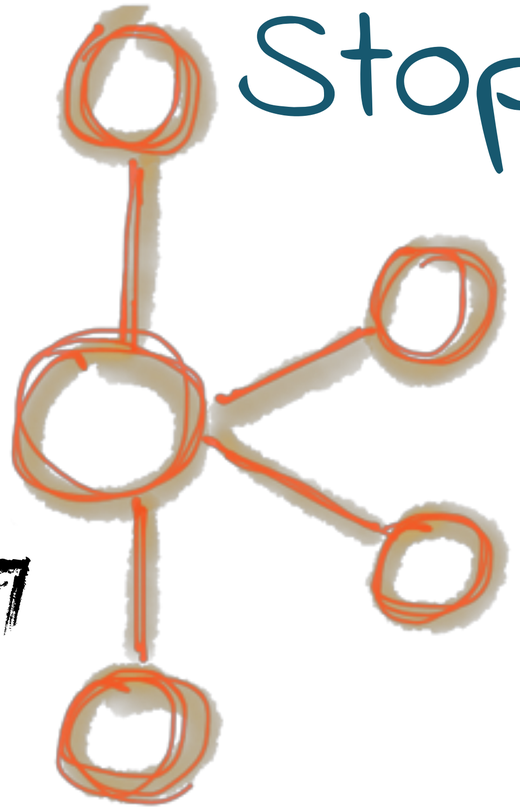

```
{
  "rating_id": 2133,
  "user_id": 9,
  "stars": 1,
  "route_id": 7219,
  "rating_time": 1519402815063,
  "channel": "web",
  "message": "worst. flight. ever. #neveragain"
}
```


Producer API

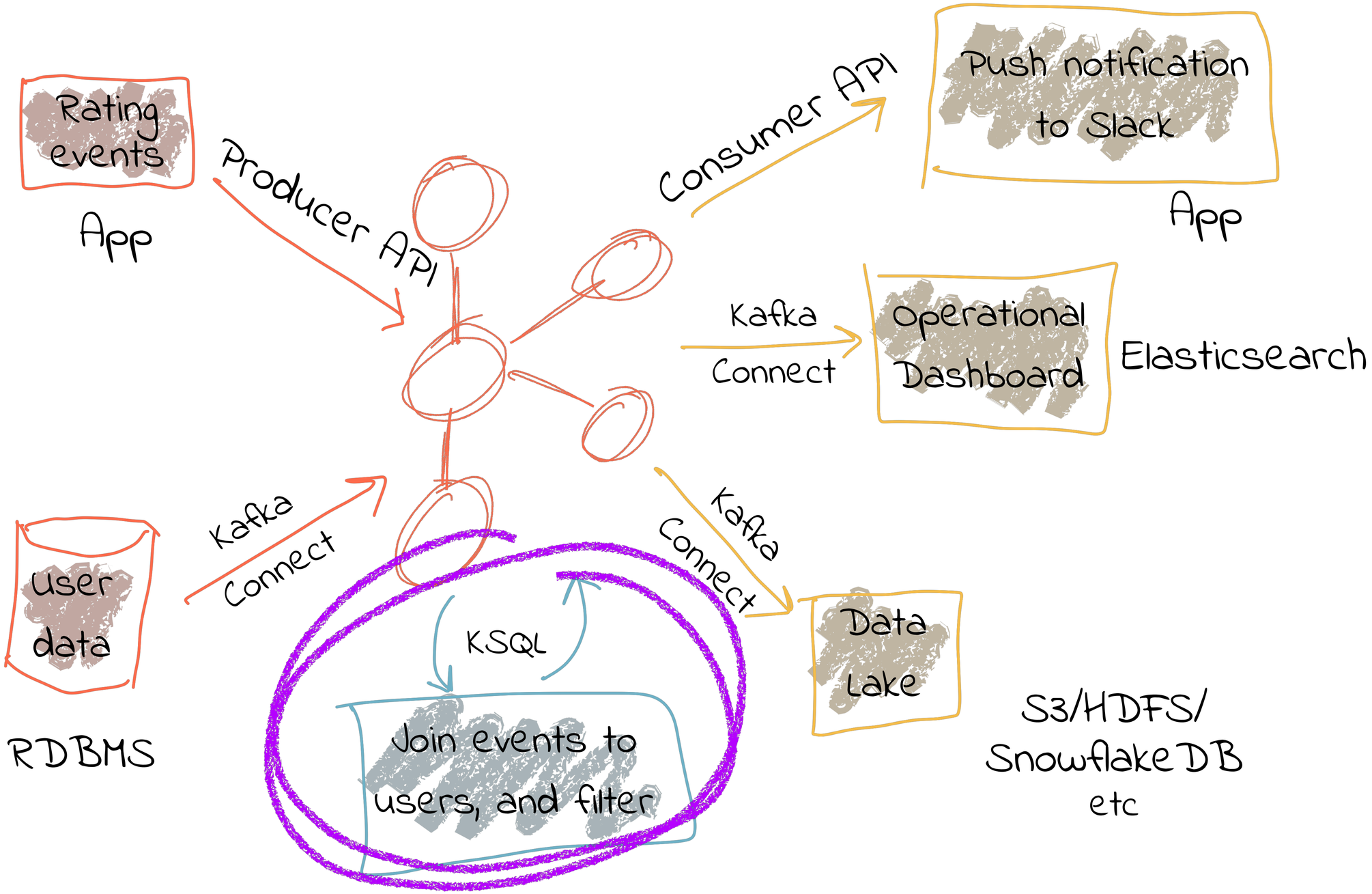
Postgres

Kafka Connect

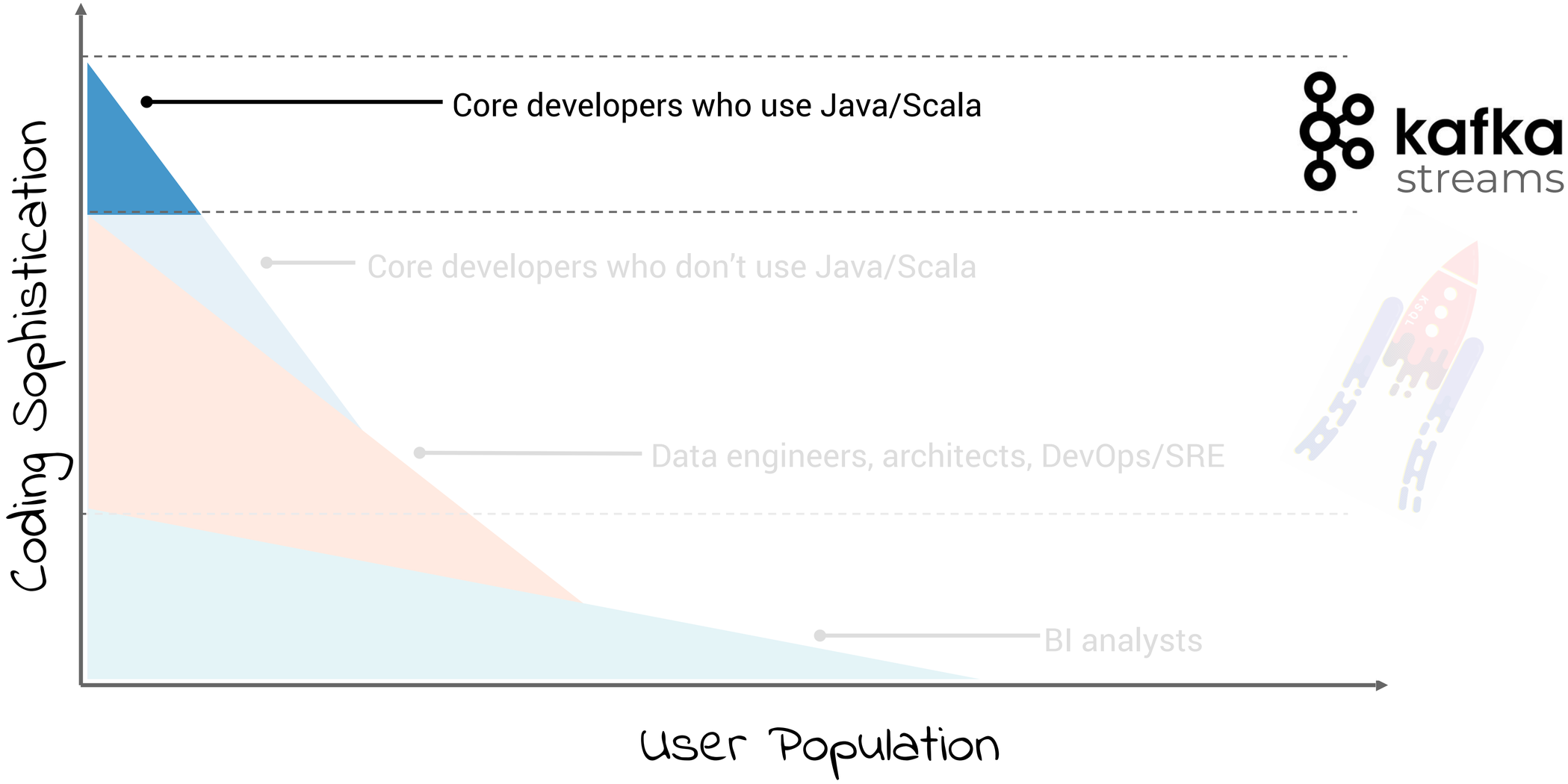
Debezium



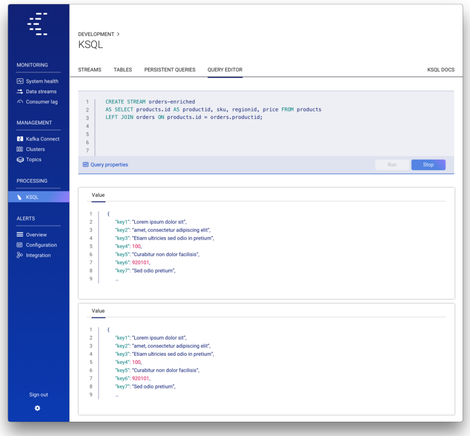

```
{
  "uid": 9,
  "name": "Neha",
  "locale": "en_US",
  "address_city": "Palo Alto",
  "elite": "P"
}
```


Rating events
App
Producer API
Consumer API
Push notification to Slack
App
Kafka Connect
Operational Dashboard
Elasticsearch
user data
Kafka Connect
Kafka Connect
KSQL
Join events to users, and filter
Data Lake
S3/HDFS/ Snowflake DB etc
RDBMS

Lower the bar to enter the world of streaming



KSQL #FTW



1 UI

2 CLI

3 REST

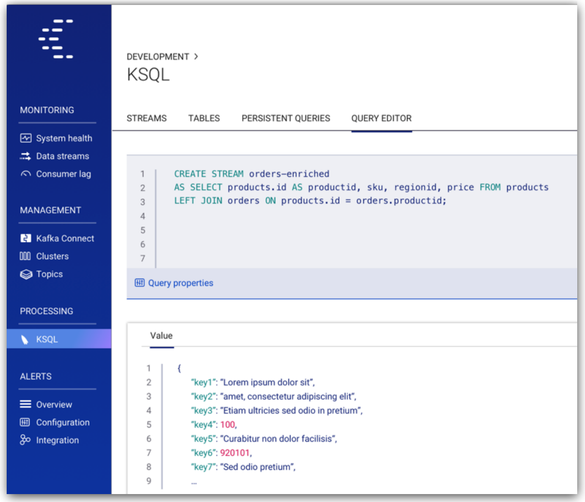
4 Headless

confluent

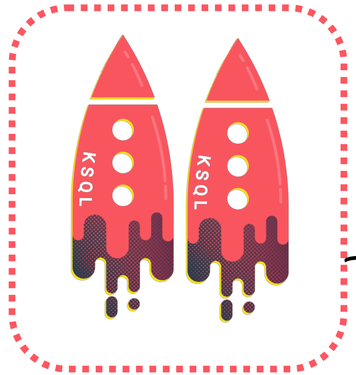
@gamussa #Postgres @confluentinc

Interactive KSQL
for development and testing

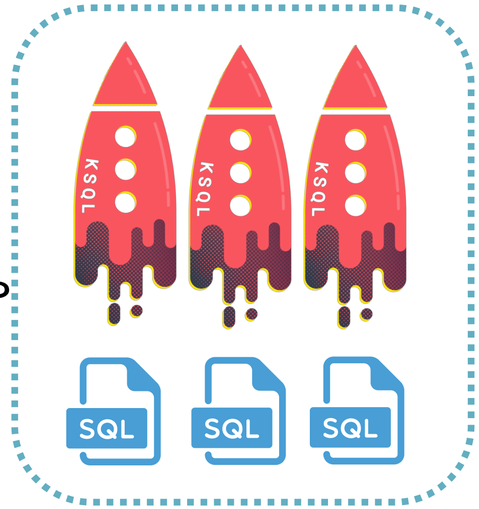
Headless KSQL
for Production



REST



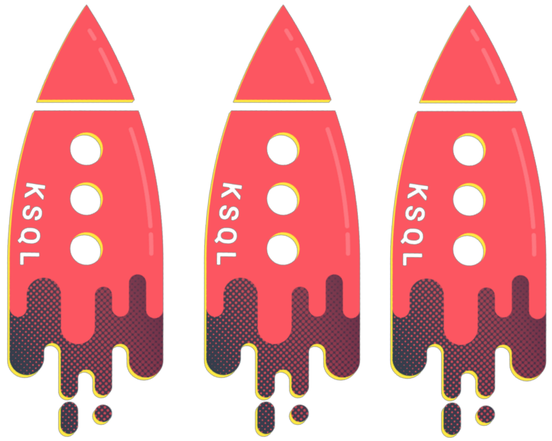
Desired KSQL queries have been identified



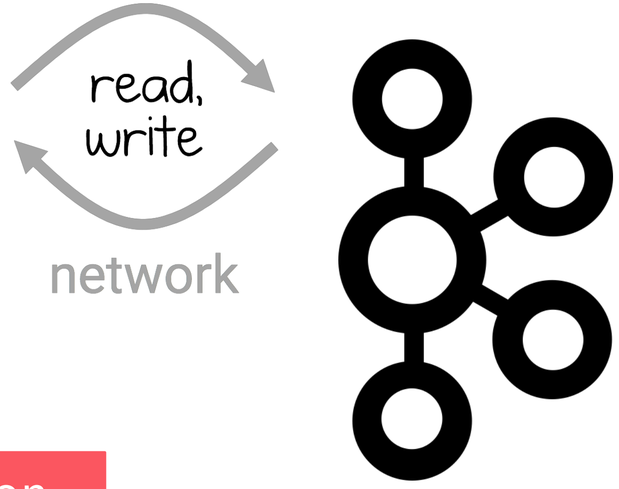
"Hmm, let me try out this idea..."

Interaction with Kafka

KSQL
(processing)



Kafka
(data)



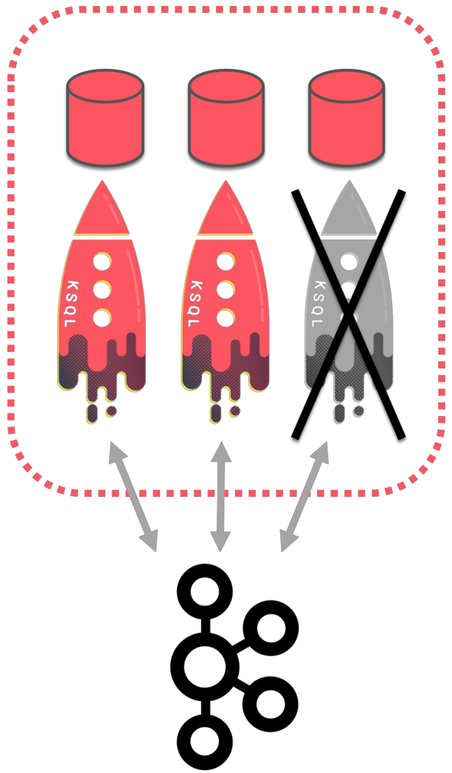
JVM application
with Kafka Streams (processing)

Does not run on
Kafka brokers

Does not run on
Kafka brokers

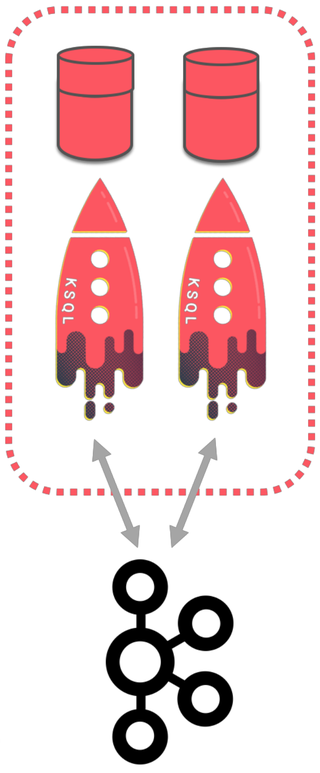
Fault-Tolerance, powered by Kafka

#3 died so #1 and #2 take over

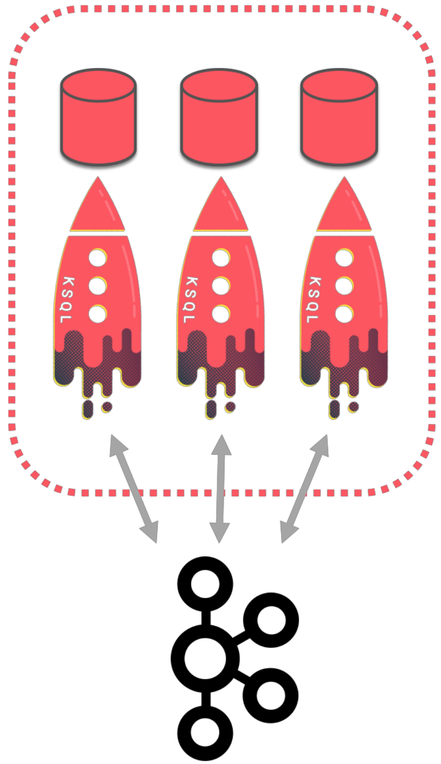


1. Kafka consumer group rebalance is triggered
2. Processing and state of #3 is migrated via Kafka to remaining servers #1 + #2

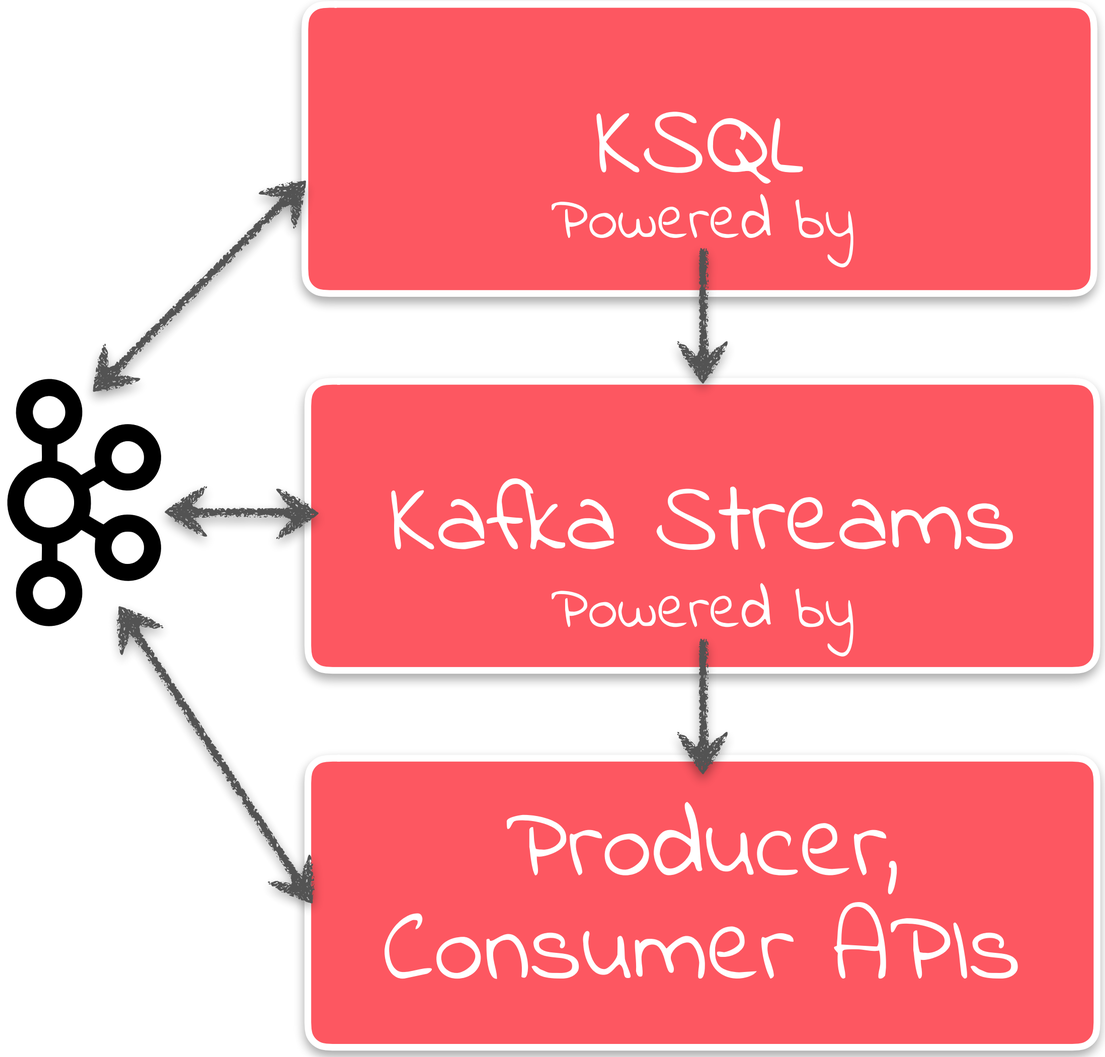
#3 is back so the work is split again



1. Kafka consumer group rebalance is triggered
2. Part of processing incl. state is migrated via Kafka from #1 + #2 to server #3



Standing on the shoulders of Streaming Giants



Ease of use

Flexibility

```
CREATE STREAM,
CREATE TABLE, SELECT, JOIN,
GROUP BY, SUM, …
```

KSQL UDFs

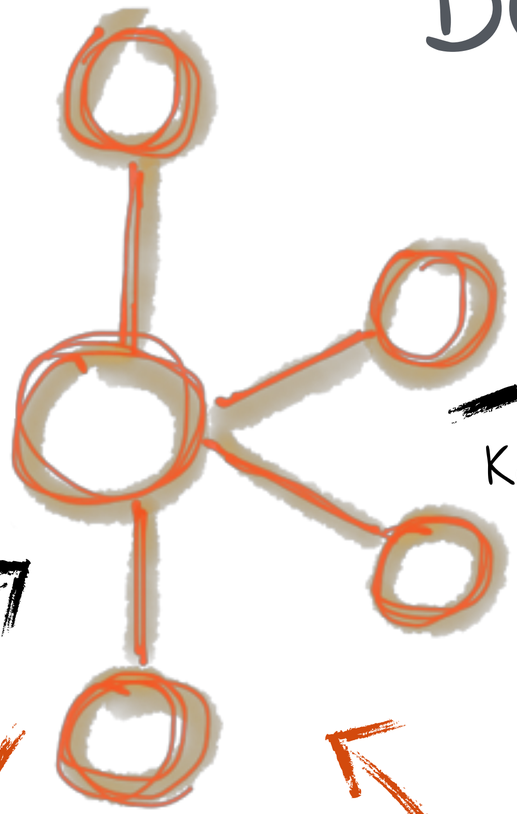
```
KStream<>, KTable<>, filter(), map(),
flatMap(), join(), aggregate(),
transform(), …
```

```
subscribe(), poll(), send(),
flush(), beginTransaction(), …
```

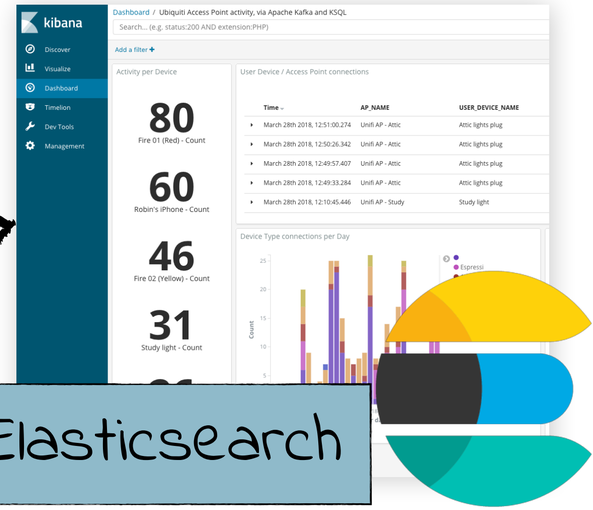
Demo Time!


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


Producer API



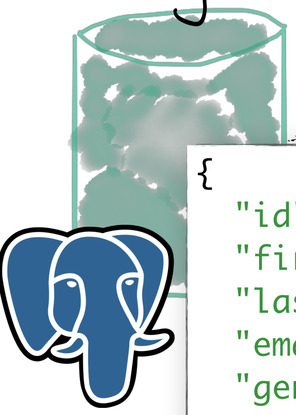
Kafka Connect



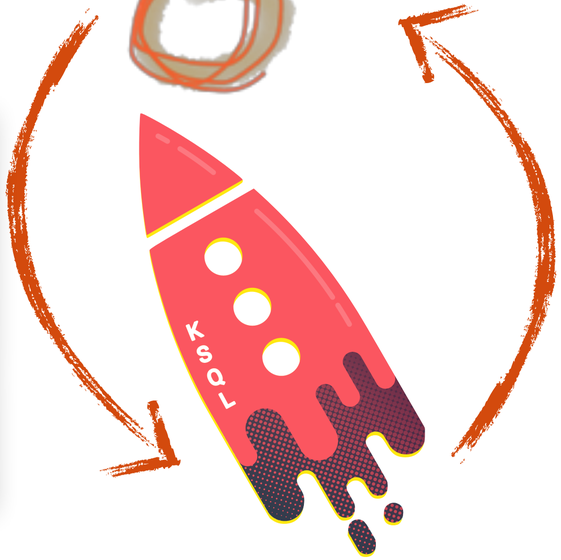
Elasticsearch

Postgres

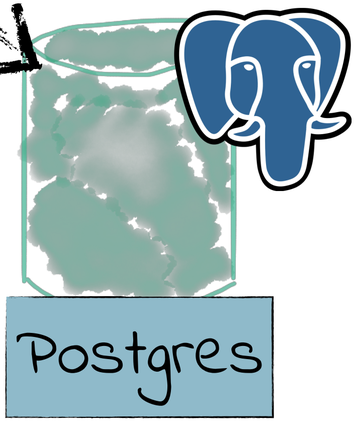
Kafka Connect




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

Kafka Connect



Postgres

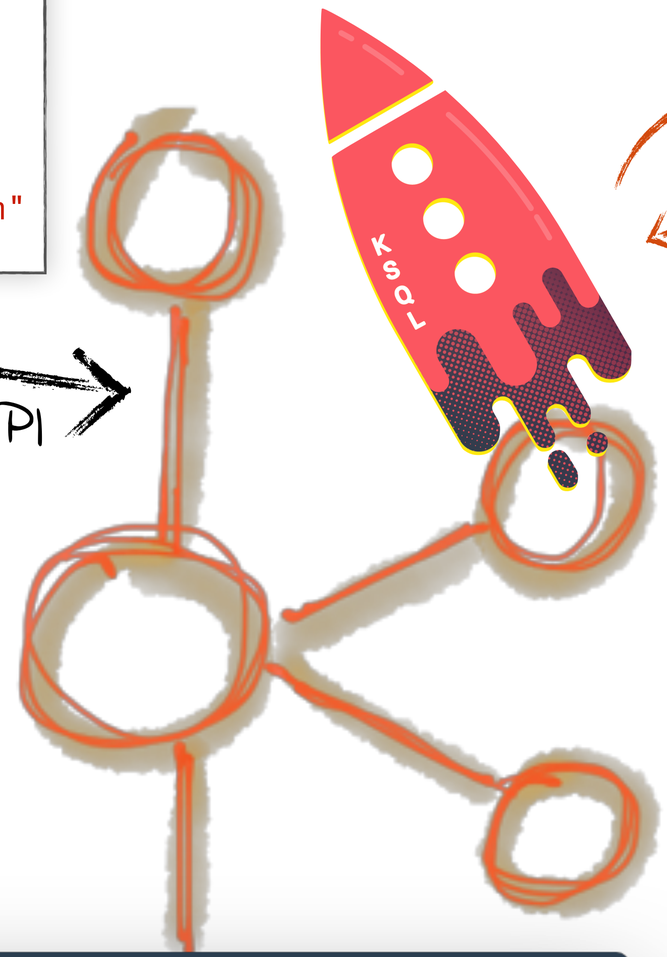
confluent

@gamussa #Postgres @confluentinc

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

Filter all ratings where STARS<3

POOR_RATINGS

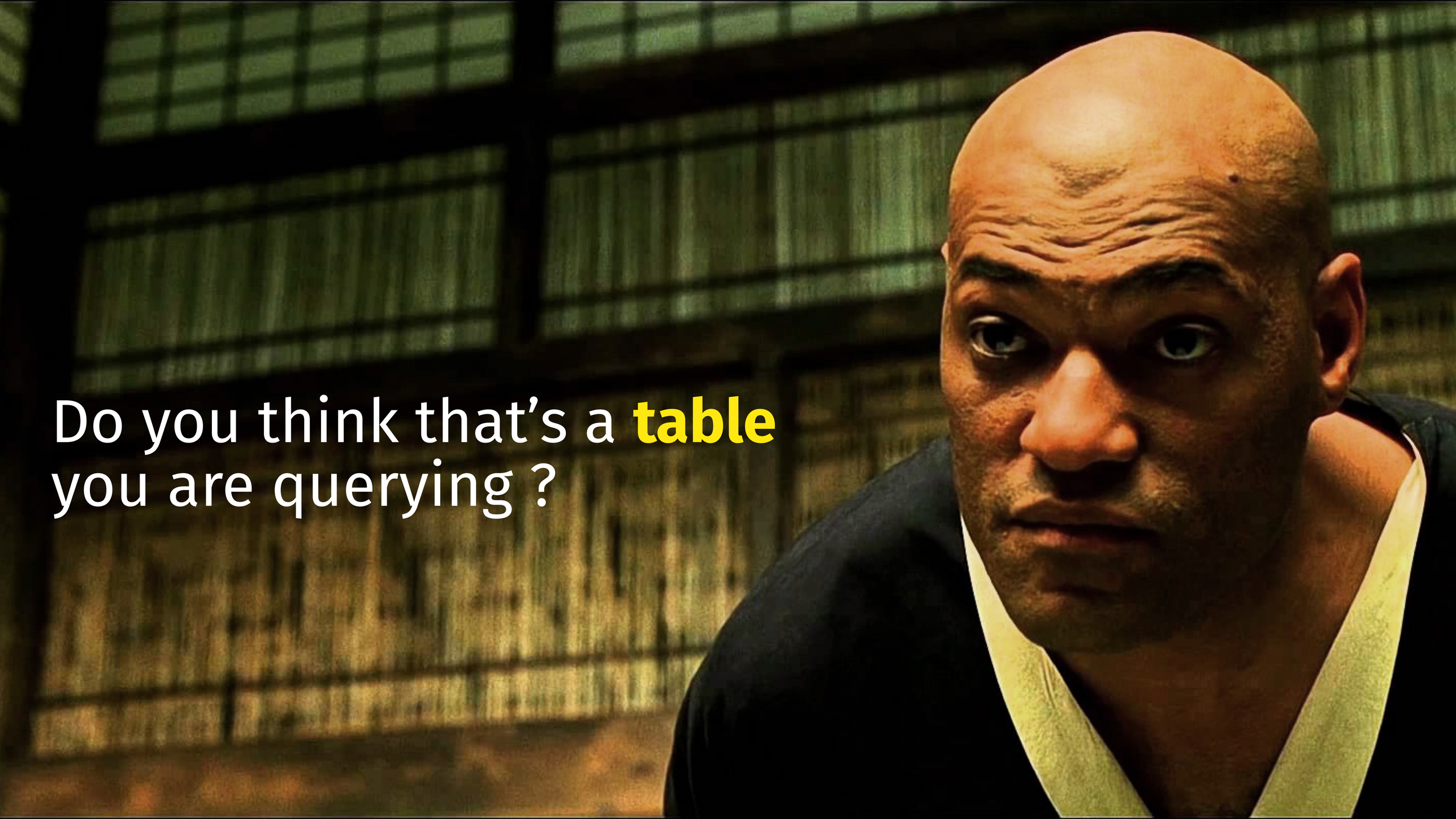


Producer API

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

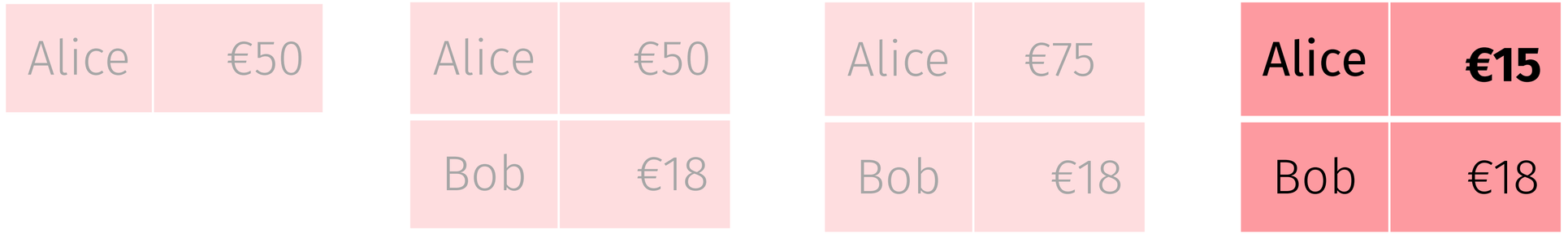
confluent

@gamussa #Postgres @confluentinc

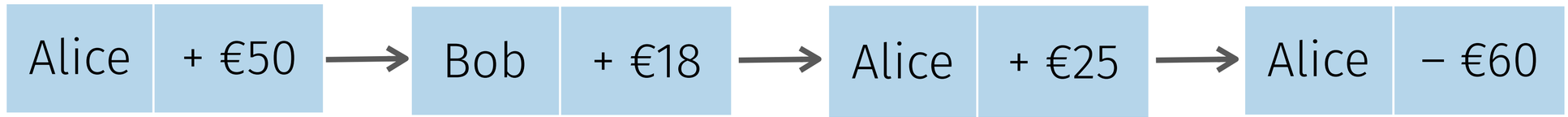

Do you think that’s a **table**
you are querying ?

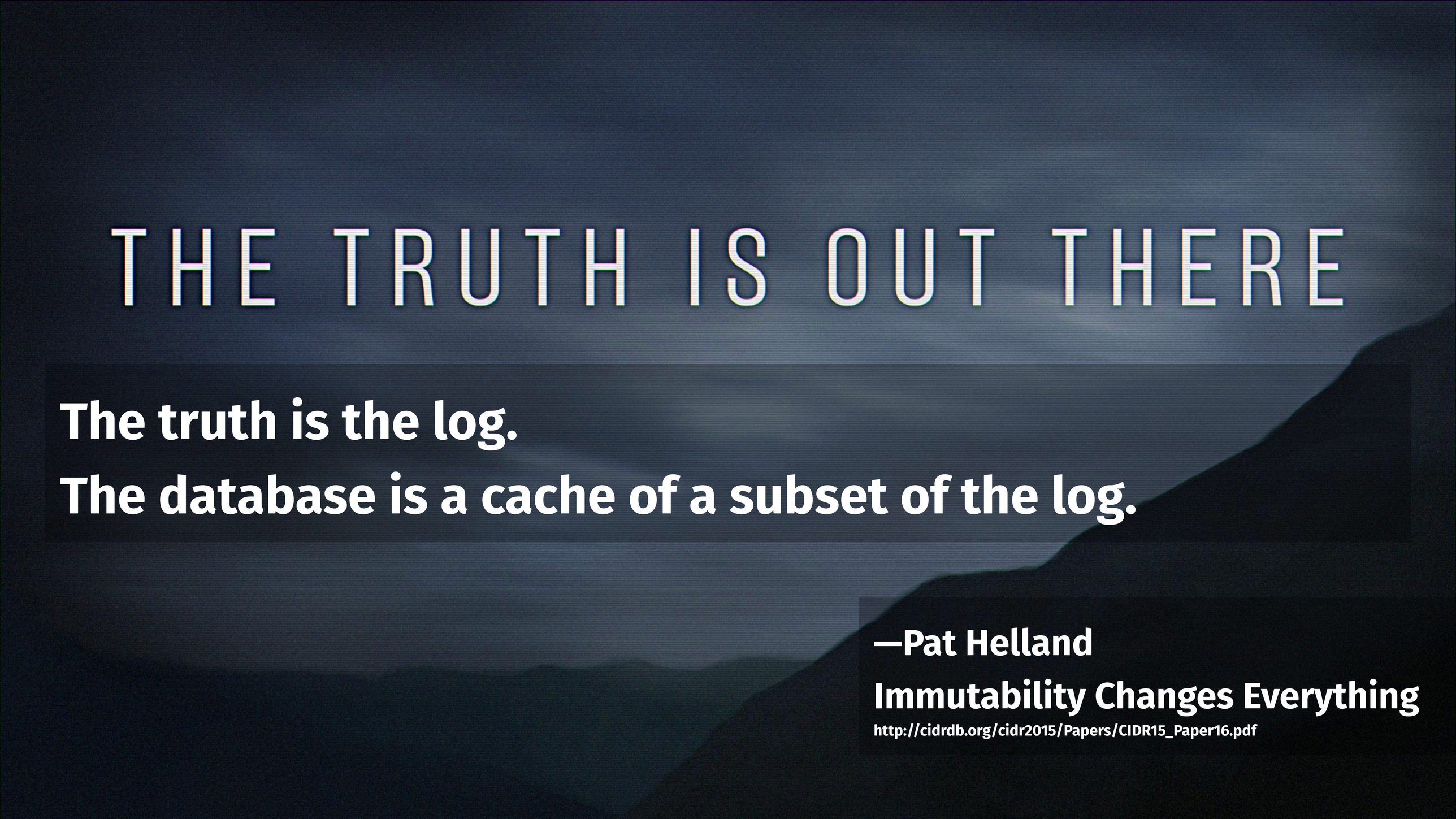
The Stream-Table Duality

Table
(balance)



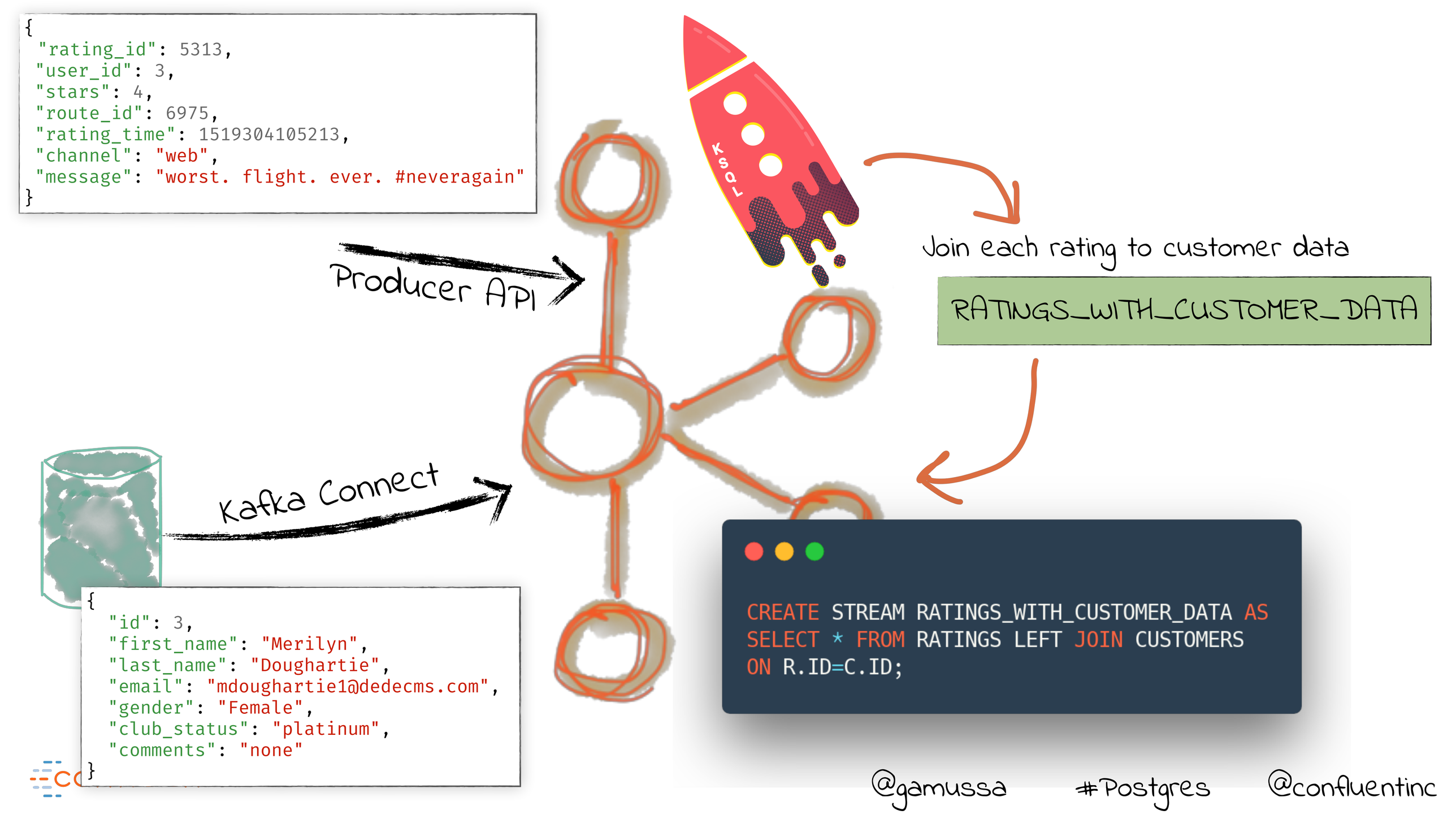
Stream
(payments)

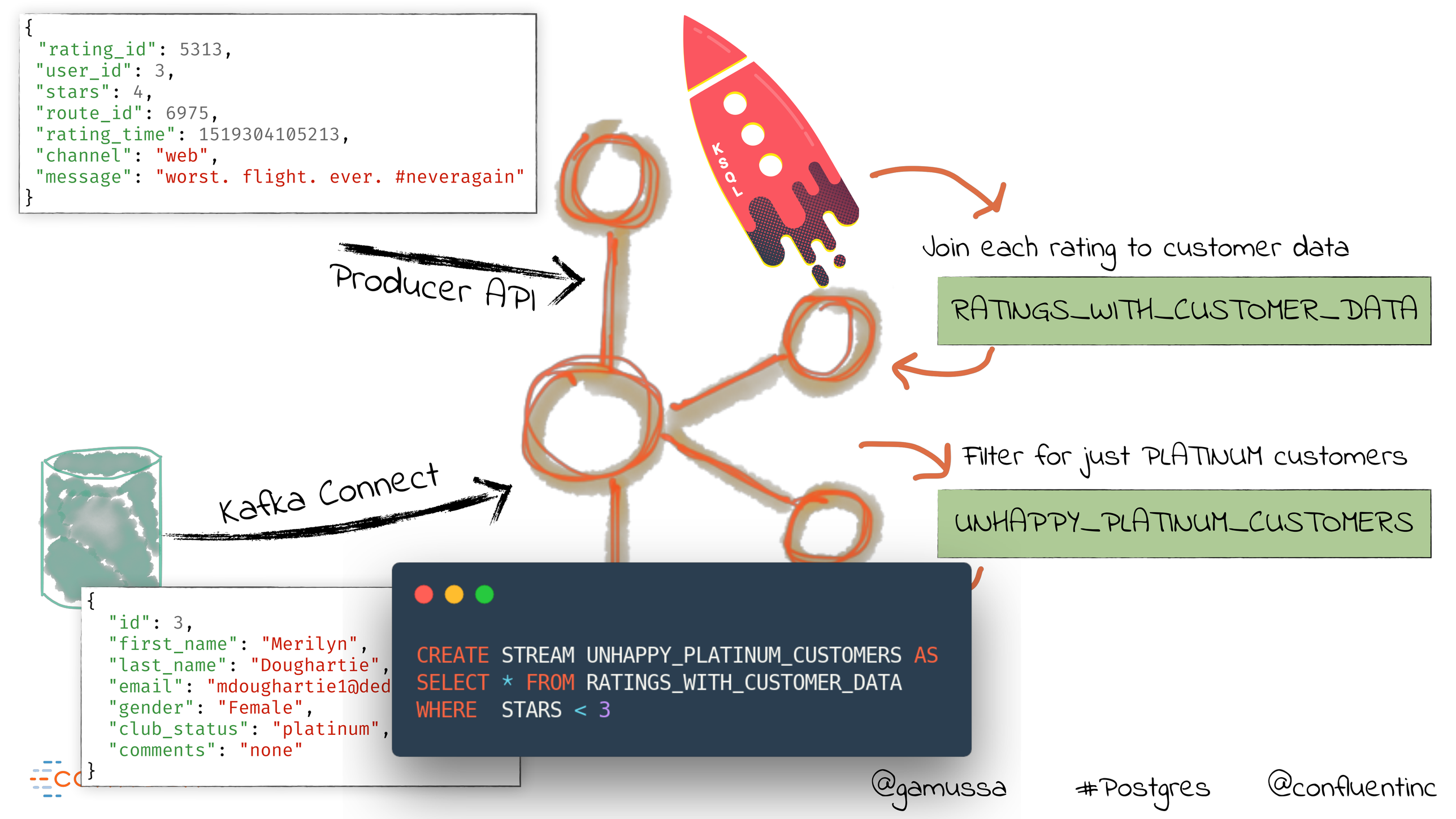




The truth is the log.
The database is a cache of a subset of the log.

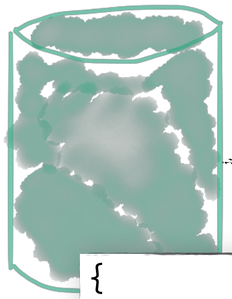
—Pat Helland
Immutability Changes Everything
http://cidrdb.org/cidr2015/Papers/CIDR15_Paper16.pdf


{
"rating_id": 5313,
"user_id": 3,
"stars": 4,
"route_id": 6975,
"rating_time": 1519304105213,
"channel": "web",
"message": "worst. flight. ever. #neveragain"
}
KSQL
Producer API
Join each rating to customer data
RATINGS_WITH_CUSTOMER_DATA
Kafka Connect
CREATE STREAM RATINGS_WITH_CUSTOMER_DATA AS
SELECT * FROM RATINGS LEFT JOIN CUSTOMERS
ON R.ID=C.ID;
{
"id": 3,
"first_name": "Merilyn",
"last_name": "Doughartie",
"email": "mdoughartie1@dedecms.com",
"gender": "Female",
"club_status": "platinum",
"comments": "none"
}
@gamussa
#Postgres
@confluentinc

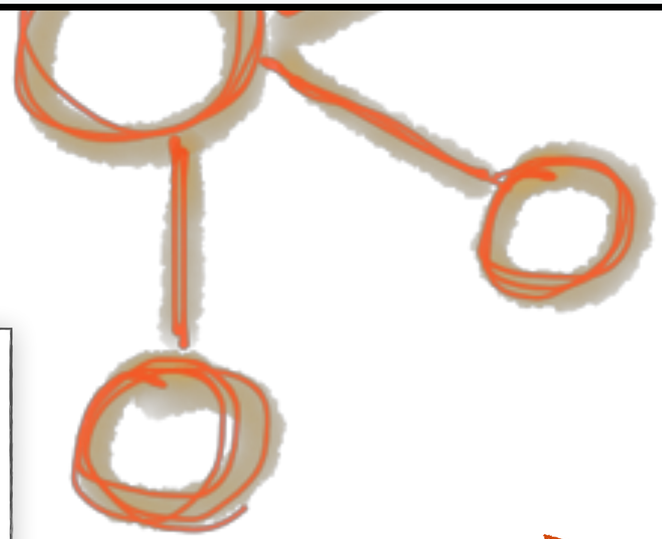

{
"rating_id": 5313,
"user_id": 3,
"stars": 4,
"route_id": 6975,
"rating_time": 1519304105213,
"channel": "web",
"message": "worst. flight. ever. #neveragain"
}
Producer API
KSQL
Join each rating to customer data
RATINGS_WITH_CUSTOMER_DATA
Kafka Connect
Filter for just PLATINUM customers
UNHAPPY_PLATINUM_CUSTOMERS
{
"id": 3,
"first_name": "Merilyn",
"last_name": "Doughartie",
"email": "mdoughartie1@ded
"gender": "Female",
"club_status": "platinum",
"comments": "none"
}
CREATE STREAM UNHAPPY_PLATINUM_CUSTOMERS AS
SELECT * FROM RATINGS_WITH_CUSTOMER_DATA
WHERE STARS < 3
@gamussa
#Postgres
@confluentinc

```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105
 "channel": "web",
 "message": "worst. flight.
}
```


```
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;
```



Kafka Connect




```
{
  "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

@gamussa #Postgres @confluentinc

Resources and Next Steps

https://github.com/confluentinc/examples

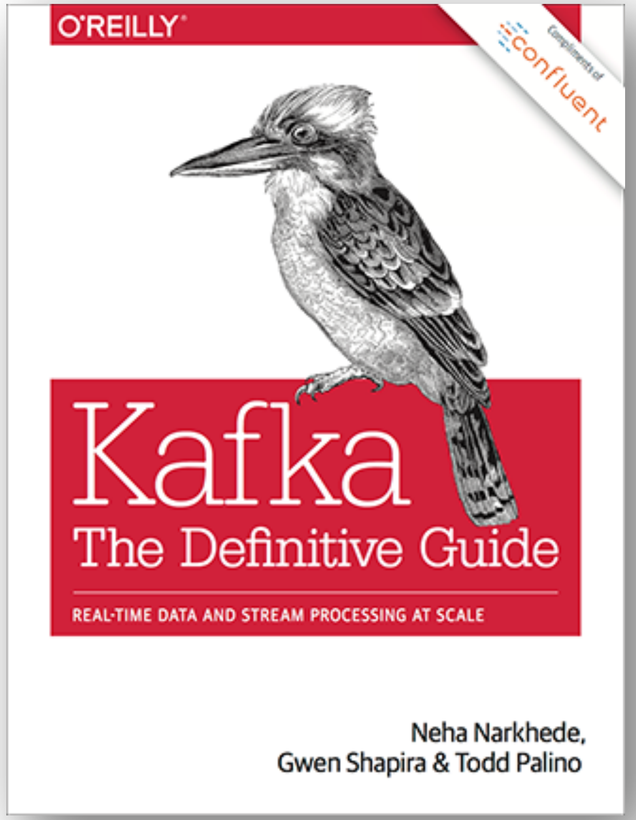
http://confluent.io

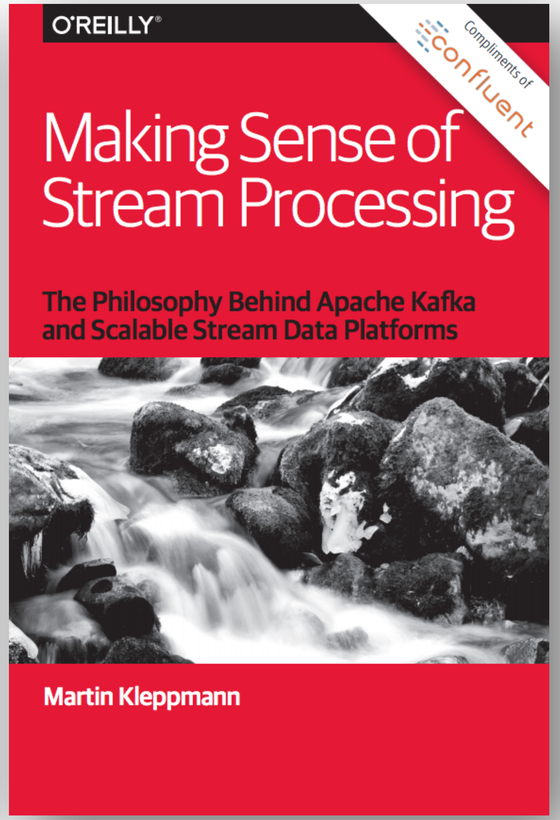
https://slackpass.io/confluentcommunity

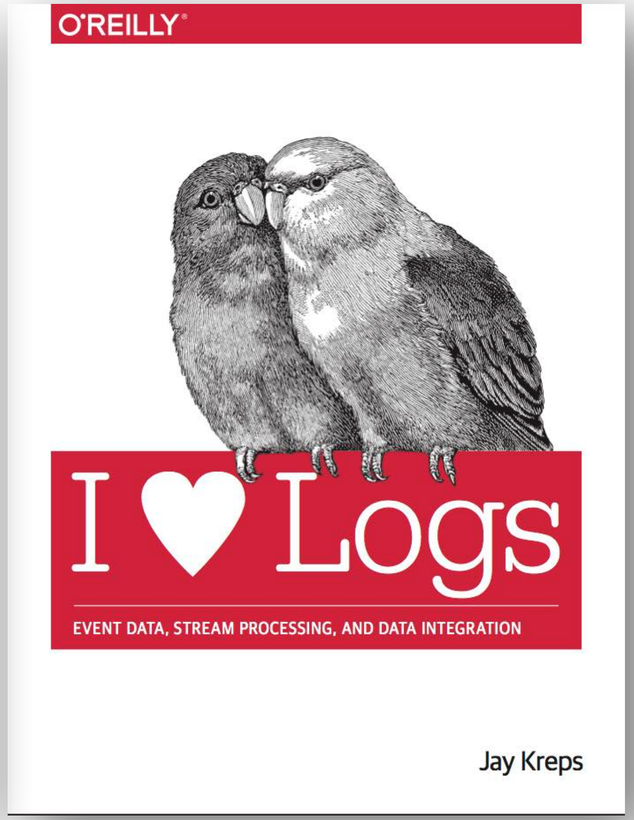
#ksql #connect

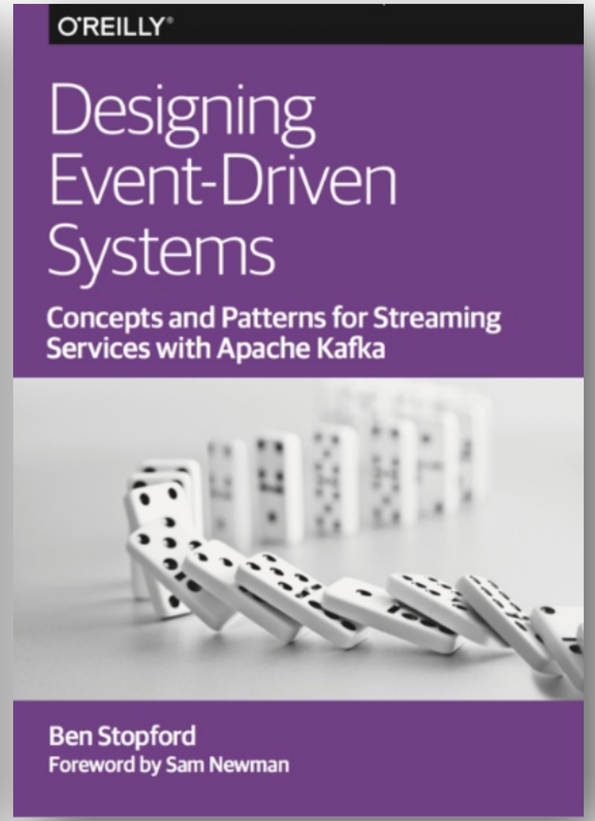
Free Books!

https://www.confluent.io/apache-kafka-stream-processing-book-bundle









confluent

@gamussa #Postgres @confluentinc

confluent

ONE LAST THING...

@gamussa #Postgres @confluentinc


kafka summit
https://kafka-summit.org
Gamov30
APRIL 2, 2019
NEW YORK CITY
REGISTER
LEARN MORE

THANKS!

@gamussa
viktor@confluent.io