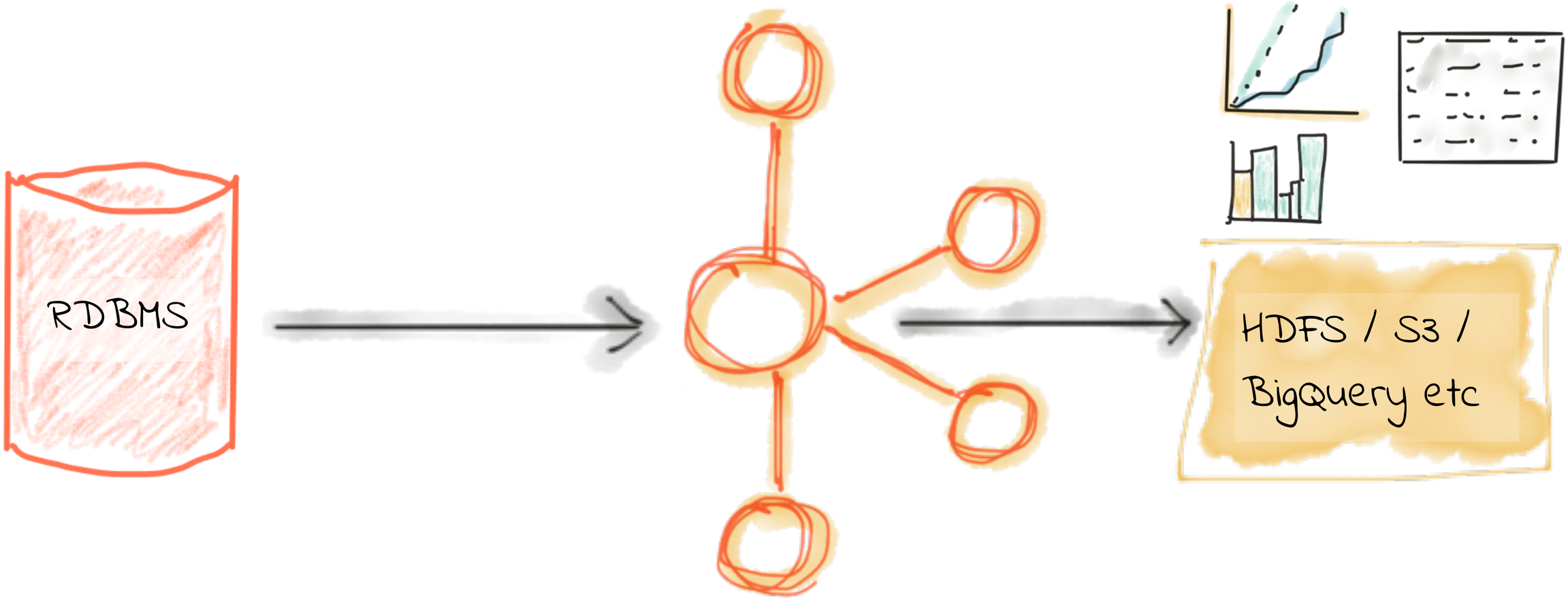

No More Silos:
Integrating Databases and Apache Kafka®
@rmoff
Berlin Buzzwords 2021
#bbuzz

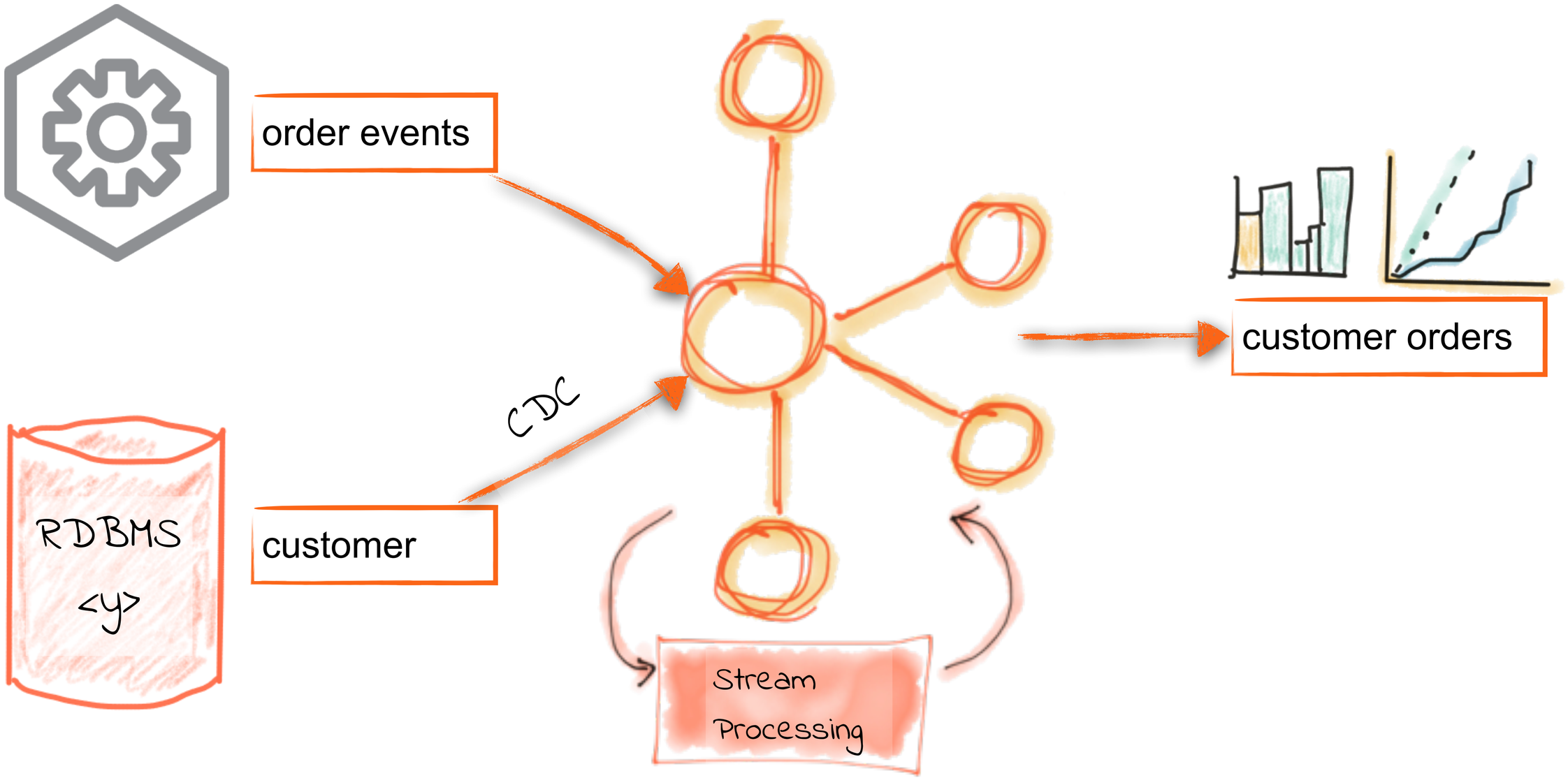


Photo by Emily Morter on Unsplash

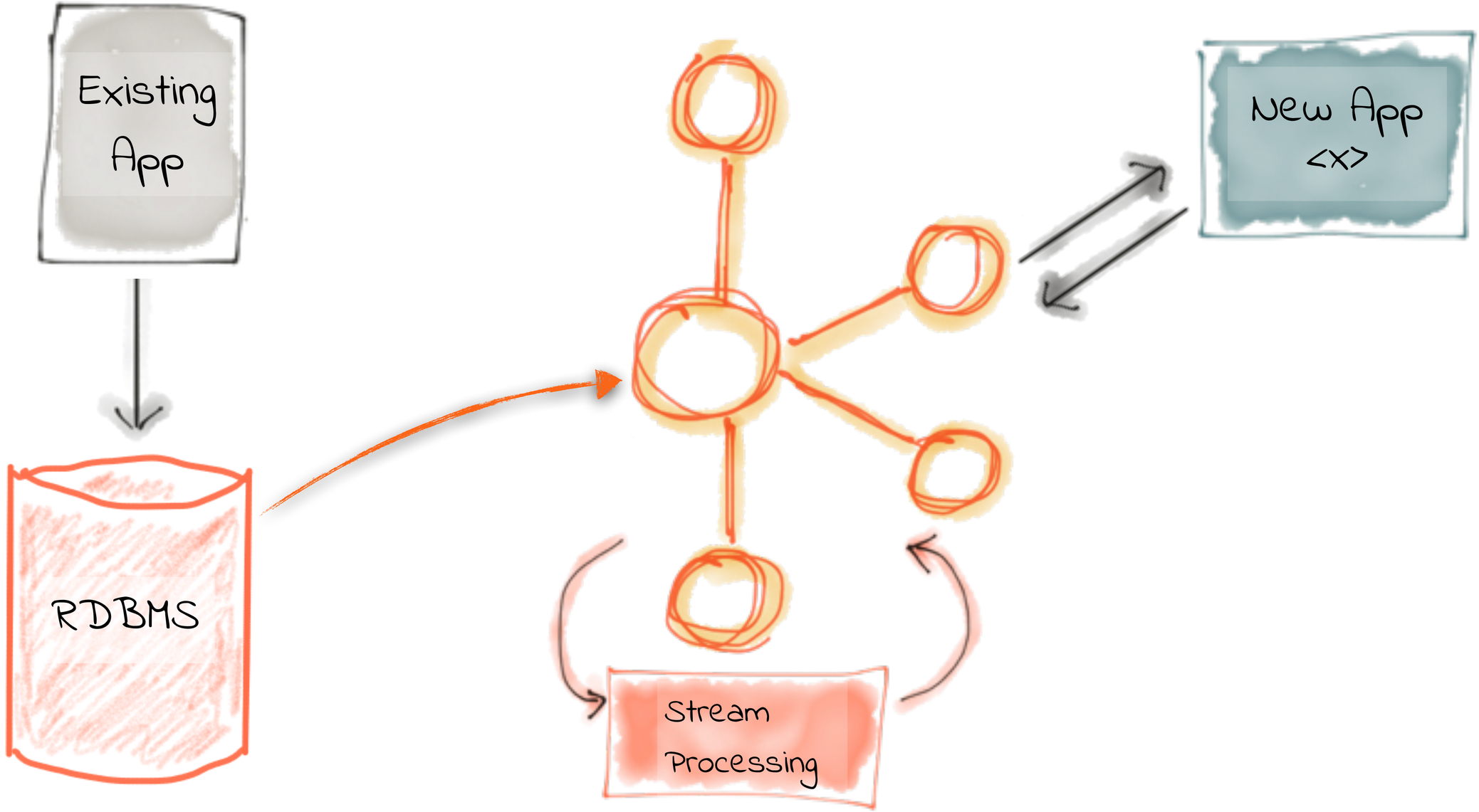
Analytics - Database Offload



Real-time Event Stream Enrichment

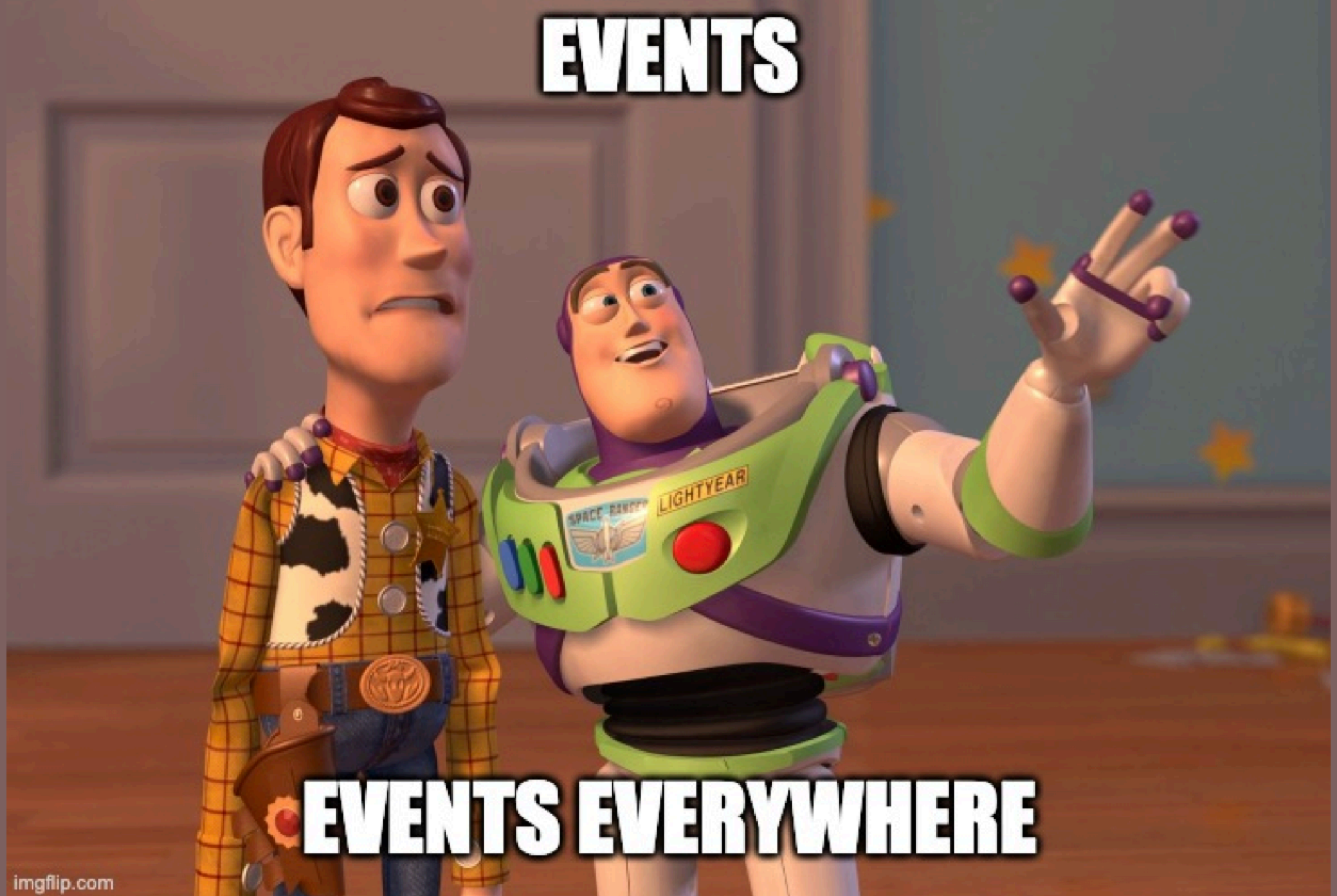


Evolve processing from old systems to new



"

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


EVENTS
LIGHTYEAR
EVENTS EVERYWHERE
imgflip.com

“

Bold claim: all your data is event streams

Human generated events

A Sale



A Stock movement

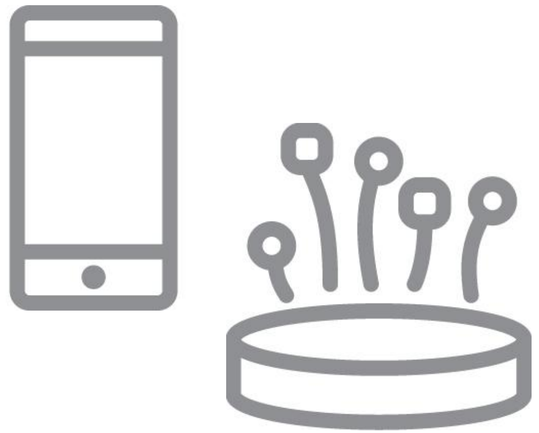


Machine generated events

Networking



IoT

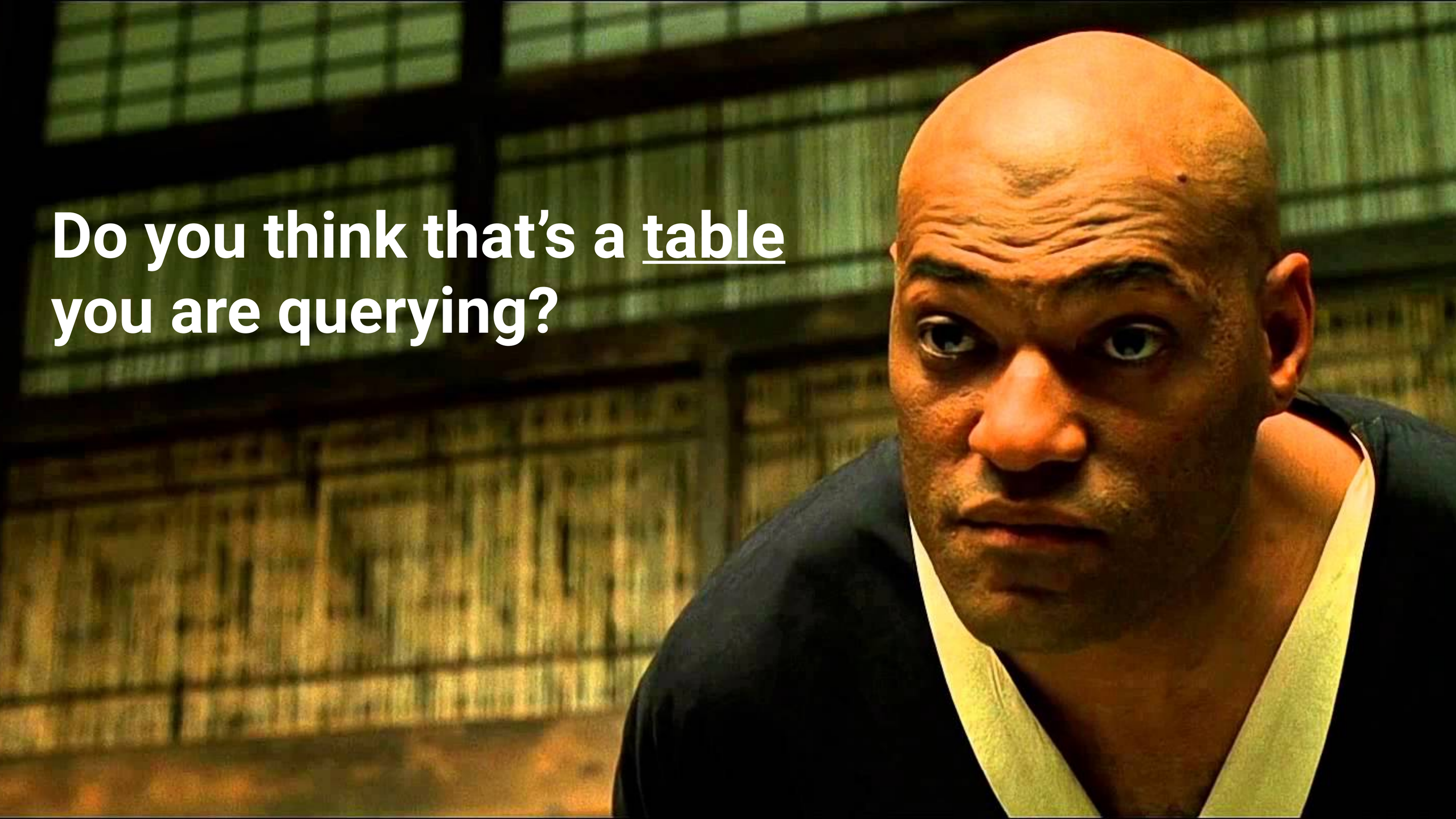


Applications



Databases




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

The Stream Table Duality

Account ID	Balance
12345	€50

The Stream Table Duality

Account ID	Amount
12345	+ €50



Account ID	Balance
12345	€50

The Stream Table Duality

Time



Account ID	Amount
12345	+ €50
12345	+ €25

Account ID	Balance
12345	€75

The Stream Table Duality



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

Account ID	Balance
12345	€15

The Stream Table Duality

Stream

Time



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

Table

Account ID	Balance
12345	€15

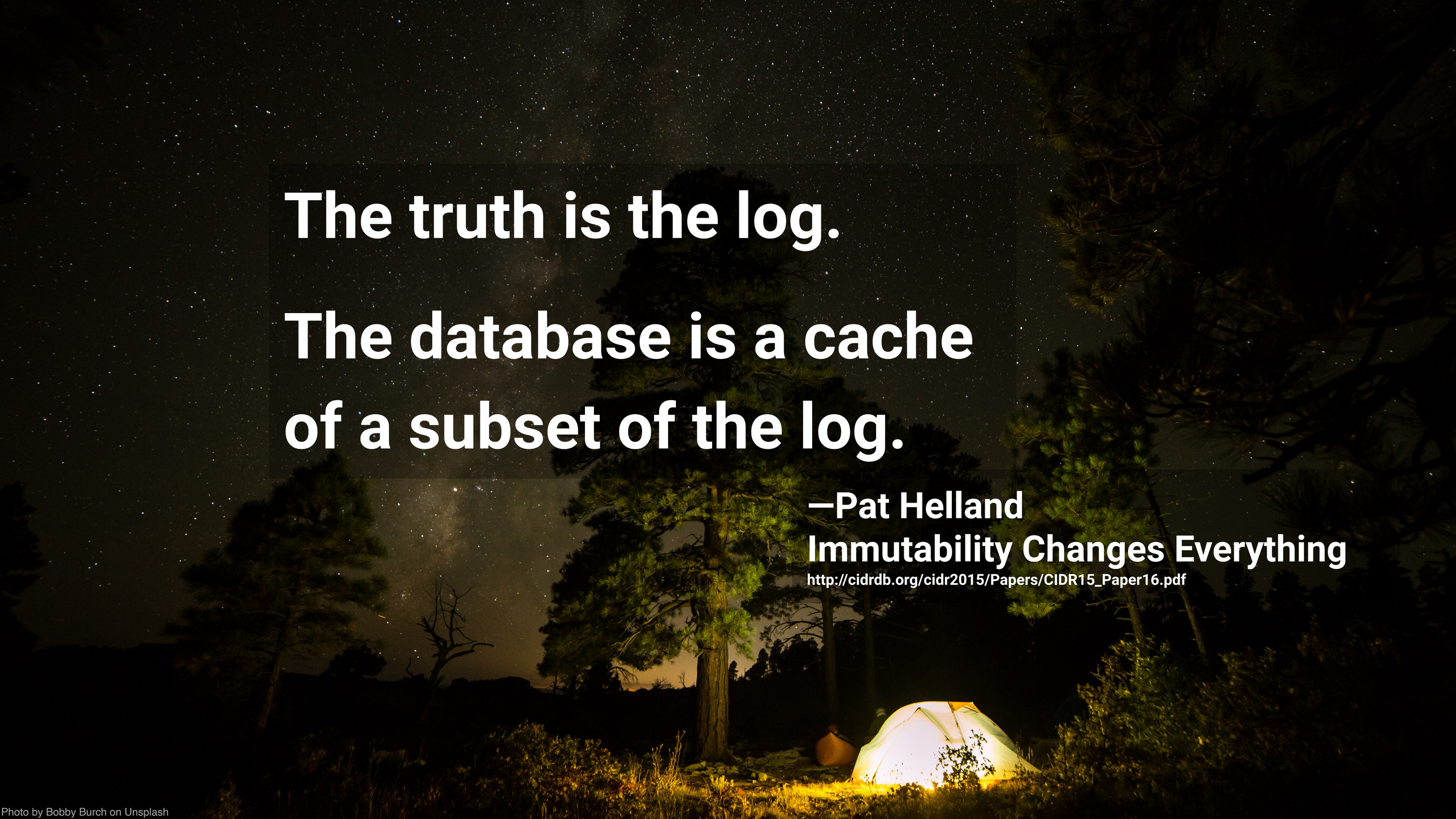
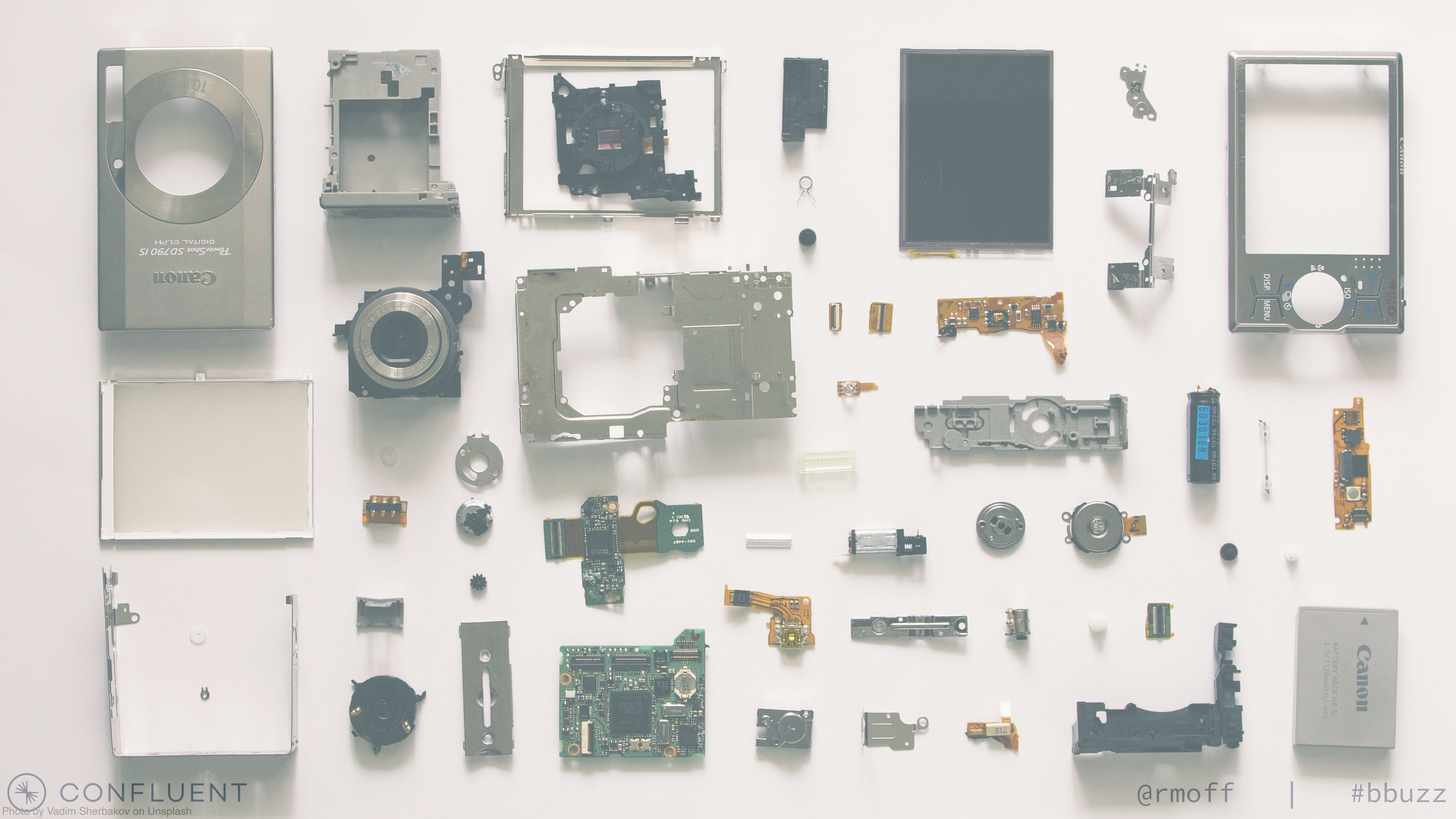


Photo by Bobby Burch on Unsplash

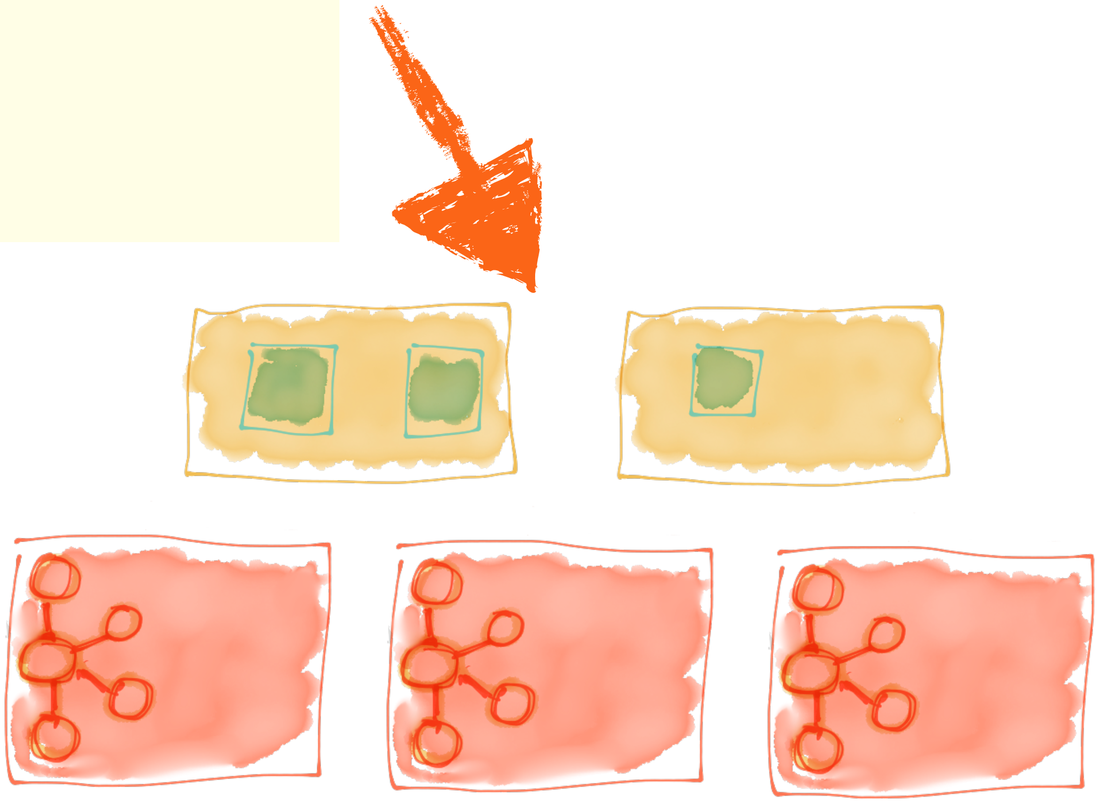


CONFLUENT
Photo by Vadim Sherbakov on Unsplash
@rmoff | #bbuzz

Streaming Integration with Kafka Connect



Sources



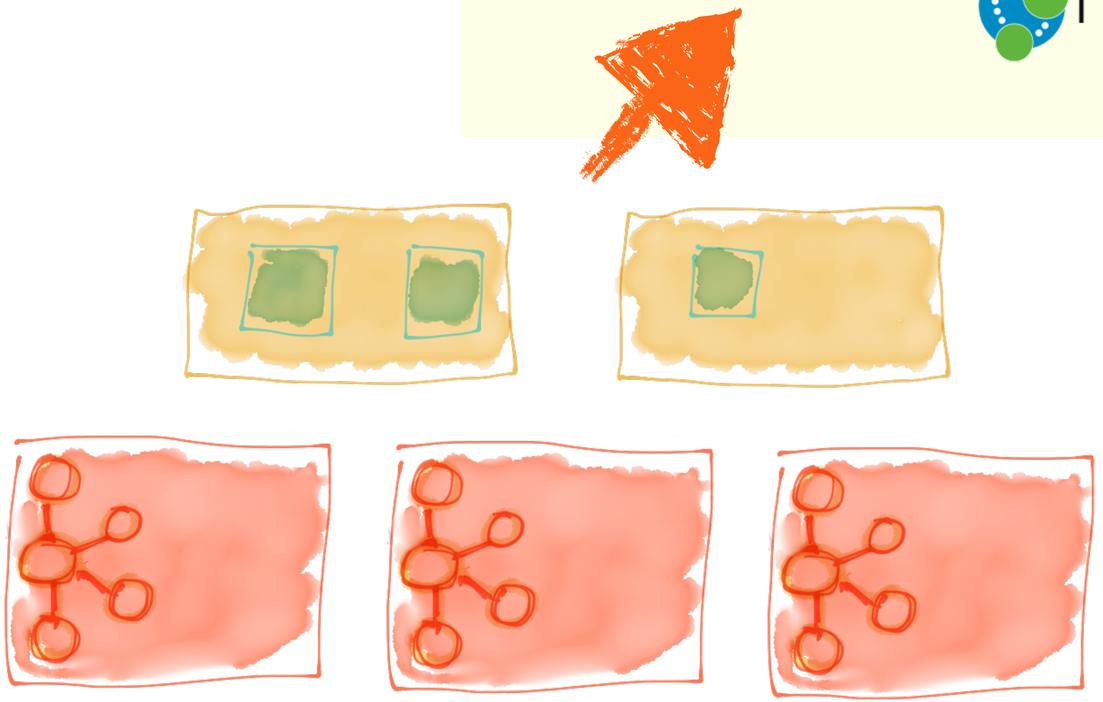
Kafka Connect

Kafka Brokers

Streaming Integration with Kafka Connect

Sinks





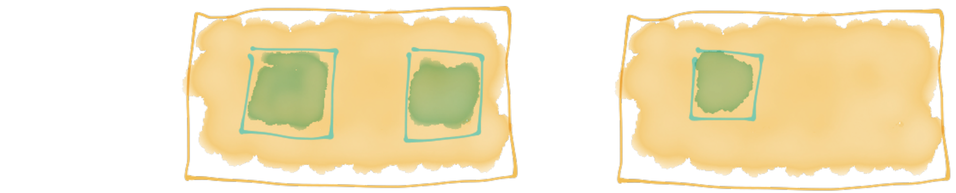
Kafka Connect

Kafka Brokers

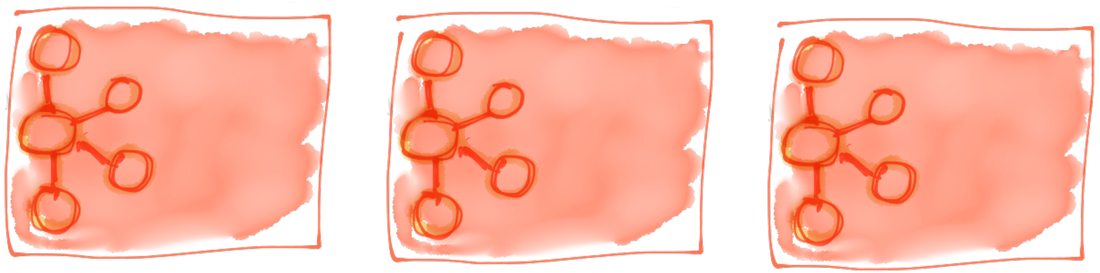
Streaming Integration with Kafka Connect







Kafka Connect



Kafka Brokers

Look Ma, No Code!

```
{
  "connector.class":
    "io.confluent.connect.jdbc.JdbcSourceConnector",
  "connection.url":
    "jdbc:mysql://asgard:3306/demo",
  "table.whitelist":
    "sales,orders,customers"
}
```

Kafka Connect basics

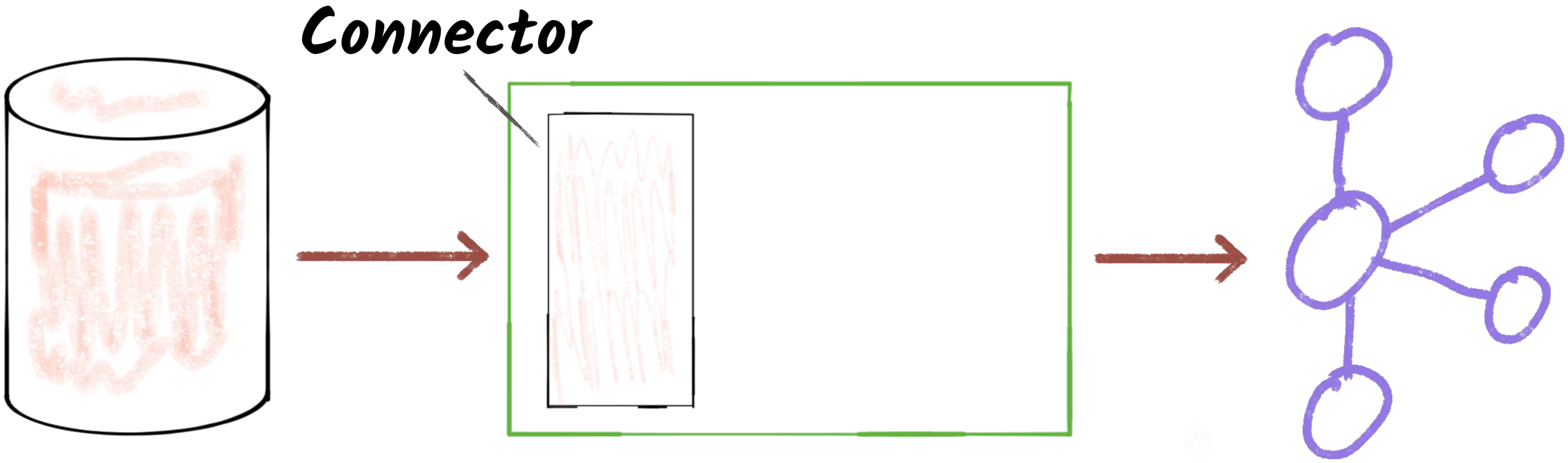


Source

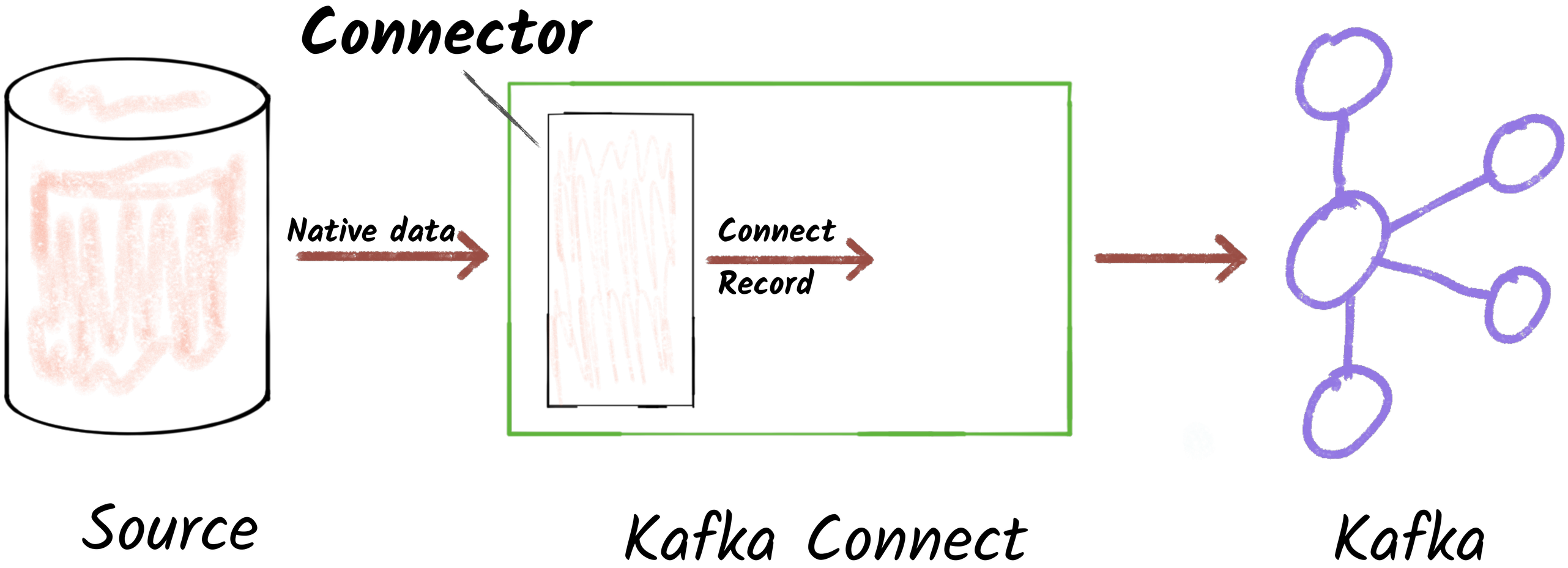
Kafka Connect

Kafka

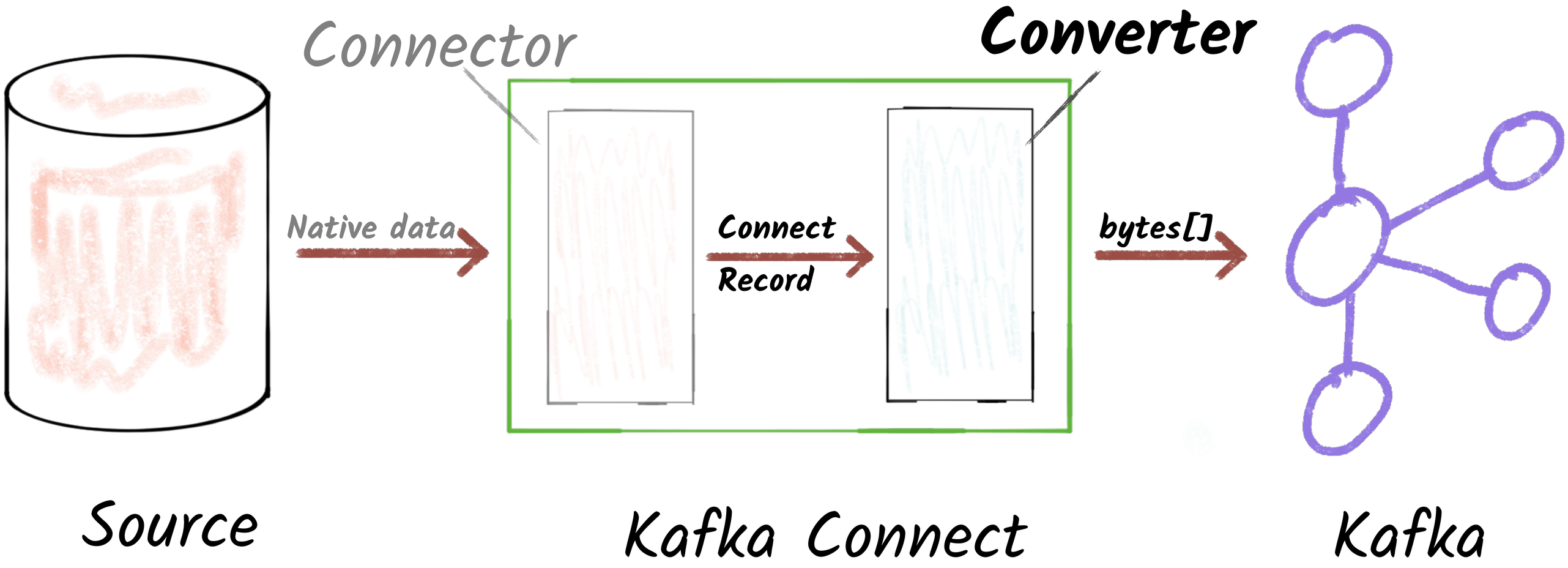
Connectors



Connectors


Connector
Native data
Connect
Record
Source
Kafka Connect
Kafka

Converters


Connector
Converter
Native data
Connect
Record
bytes[]
Source
Kafka Connect
Kafka

Serialisation & Schemas

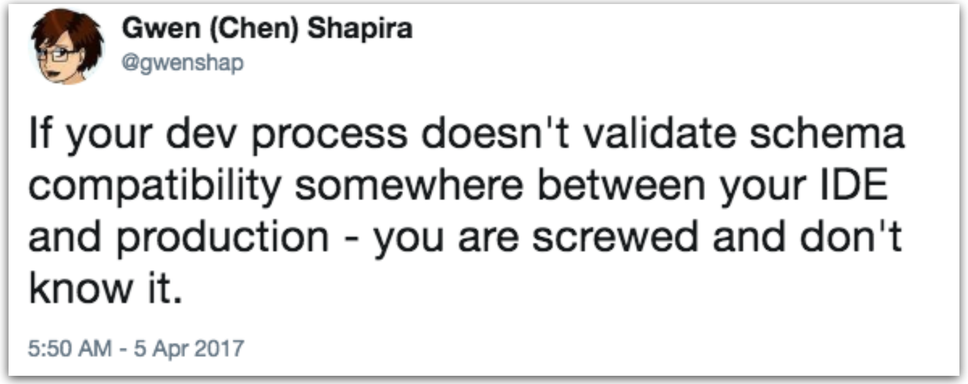
Avro 👍

Protobuf 👍

JSON Schema 👍

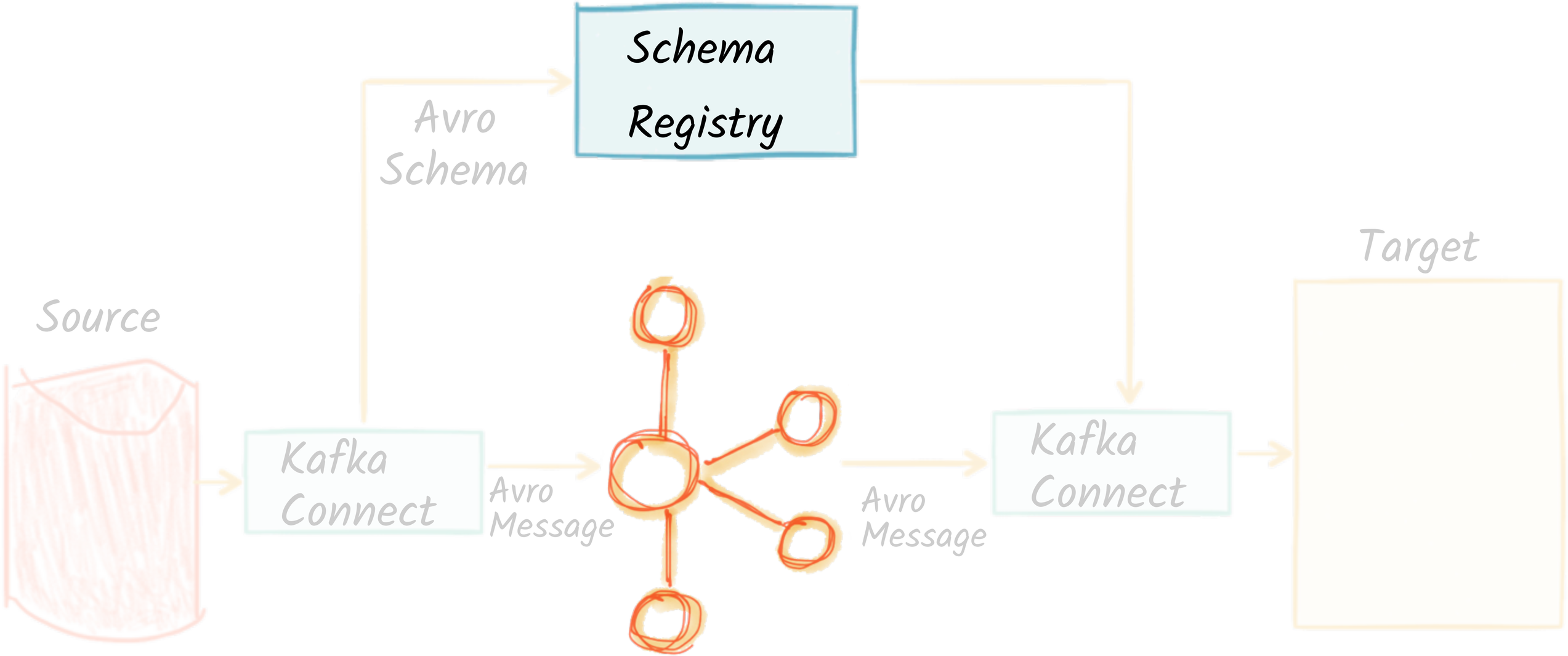
JSON 😬

CSV 

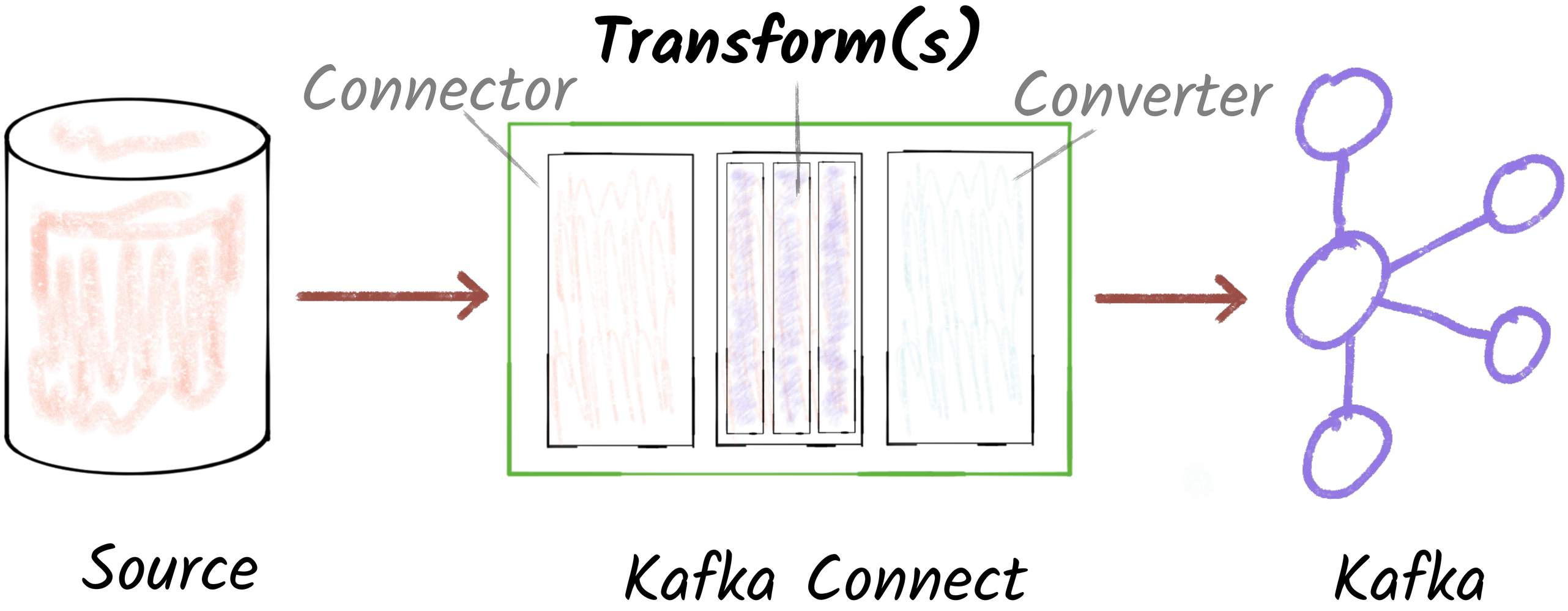


https://rmoff.dev/qcon-schemas

The Confluent Schema Registry



Single Message Transforms

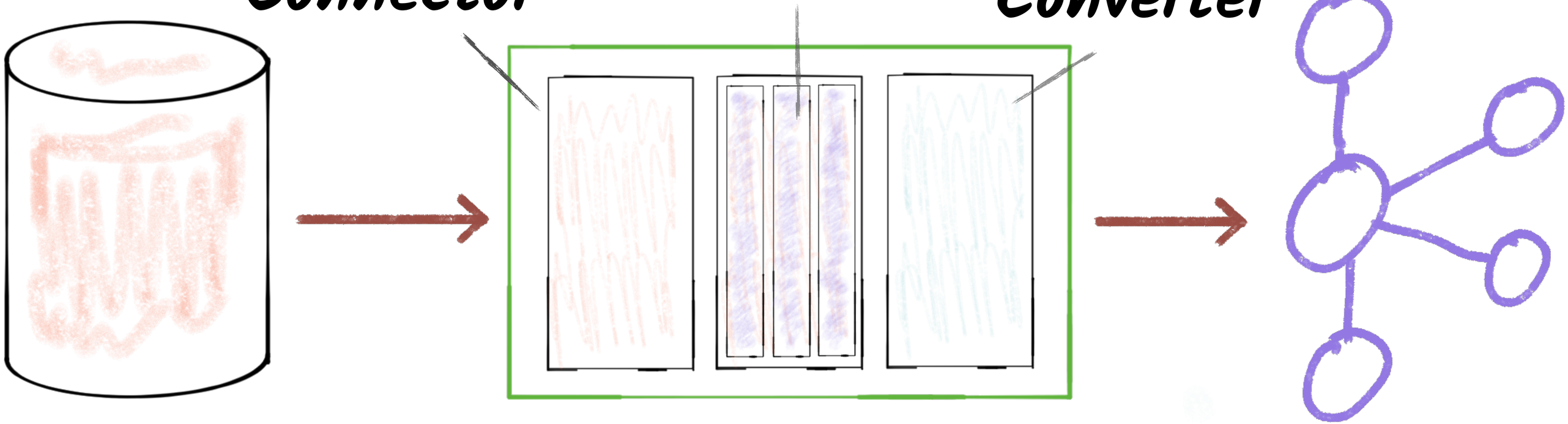


Extensible

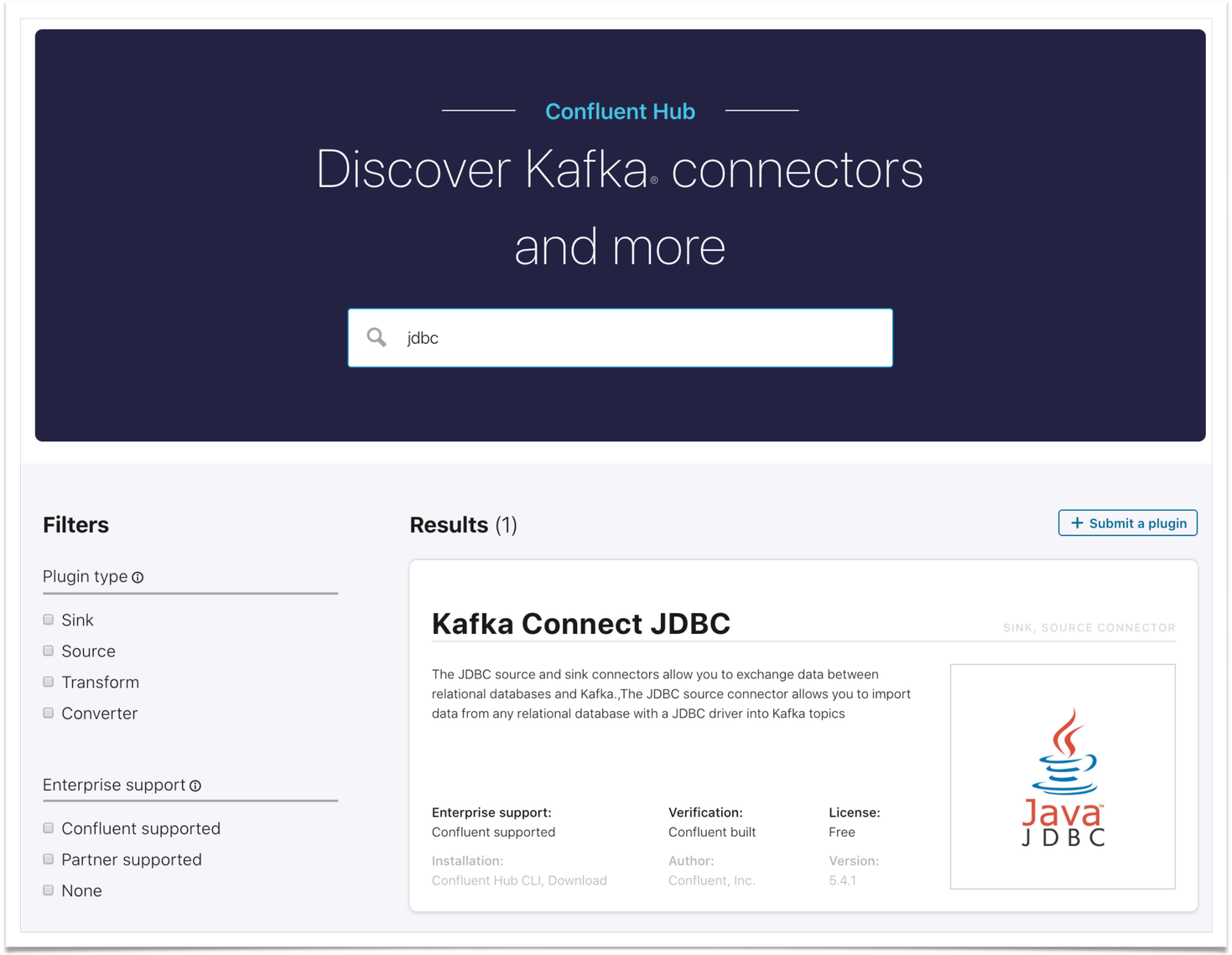
Connector

Transform(s)

Converter



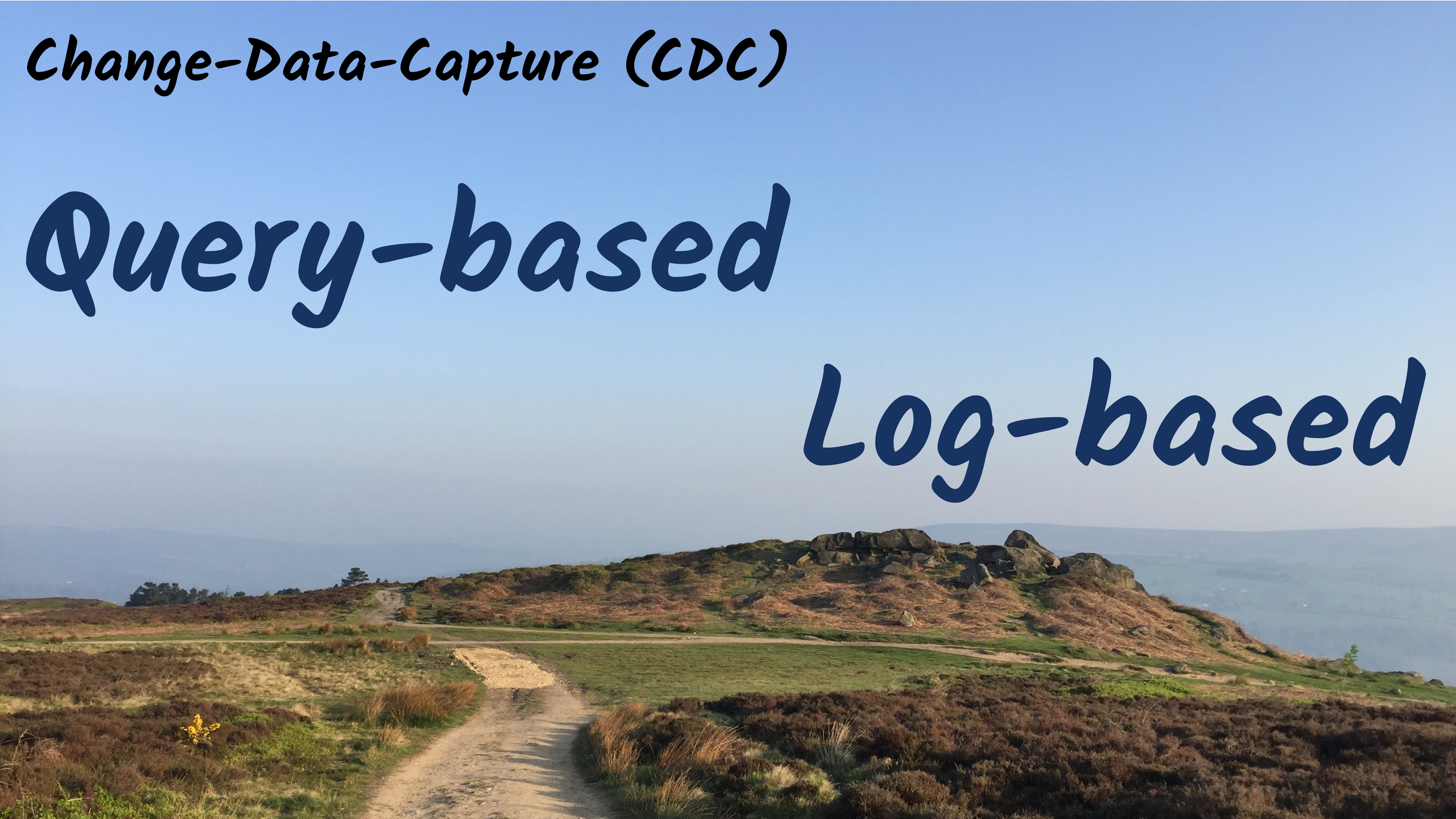
hub.confluent.io



Change-Data-Capture (CDC)

Query-based

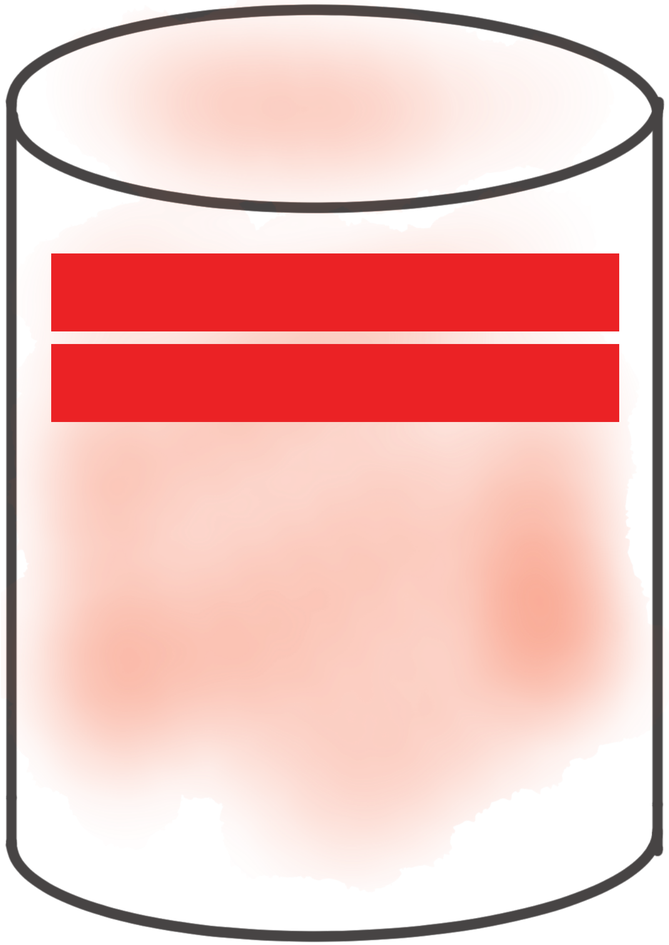
Log-based



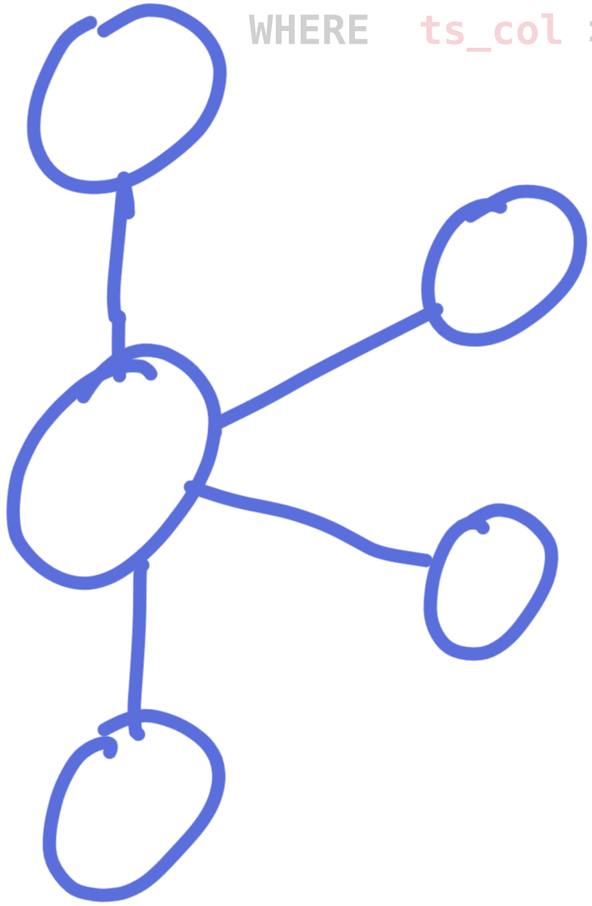
Query-based CDC

```
SELECT *
FROM   my_table
WHERE  ts_col > previous ts
```

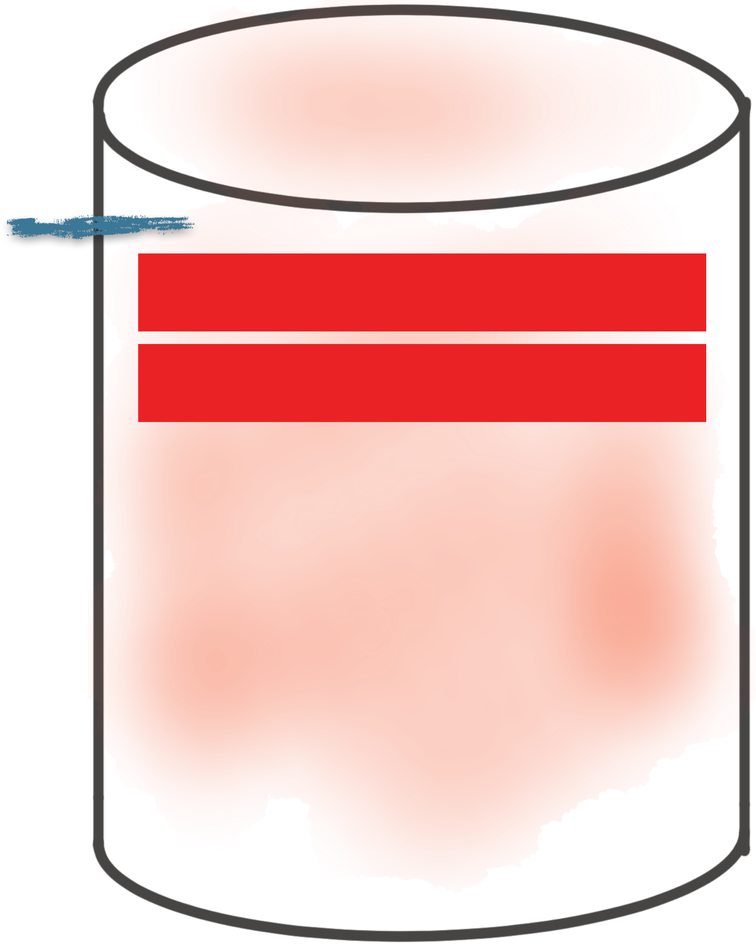
Query-based CDC



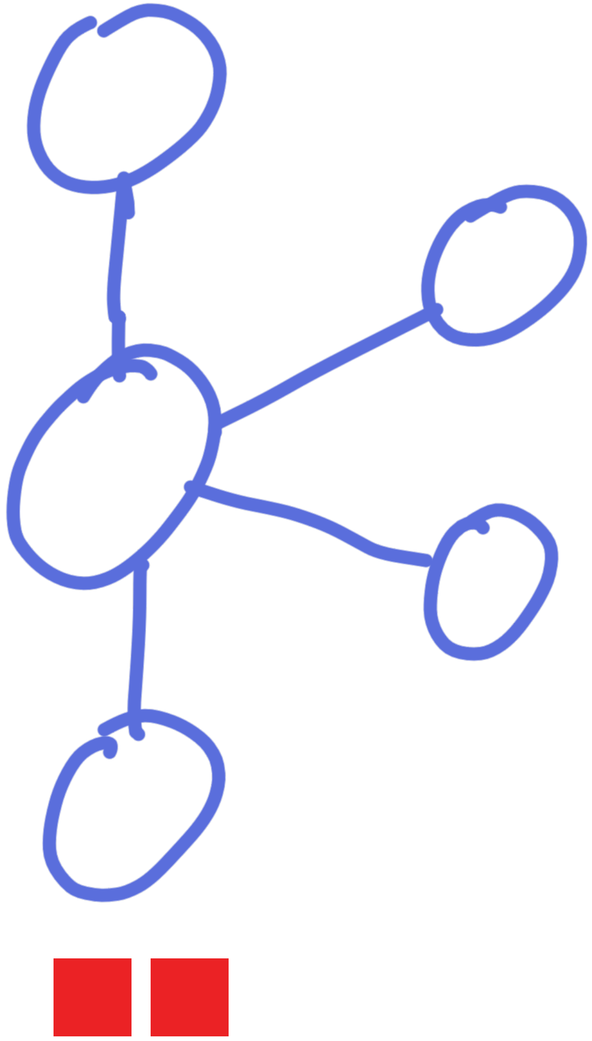
```
SELECT *
FROM   my_table
WHERE  ts_col > previous ts
```



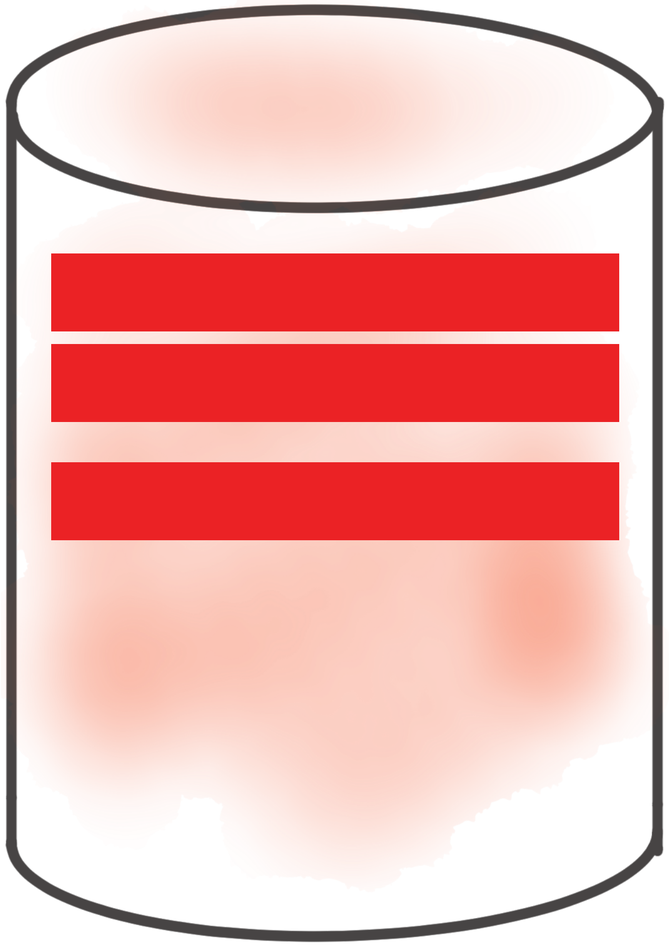
Query-based CDC



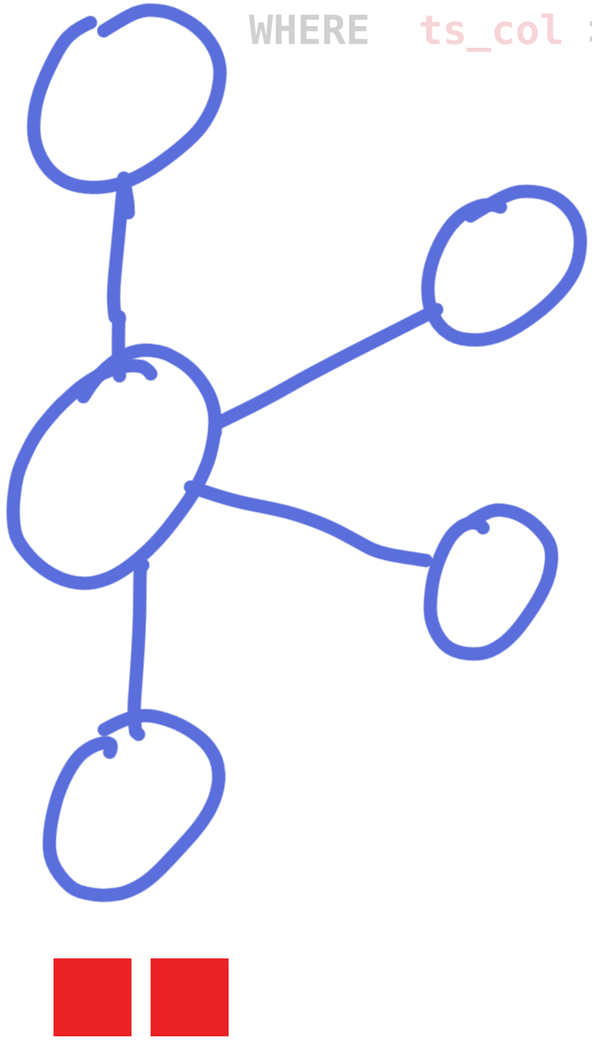
```
SELECT *
FROM   my_table
WHERE  ts_col >
         previous ts
```



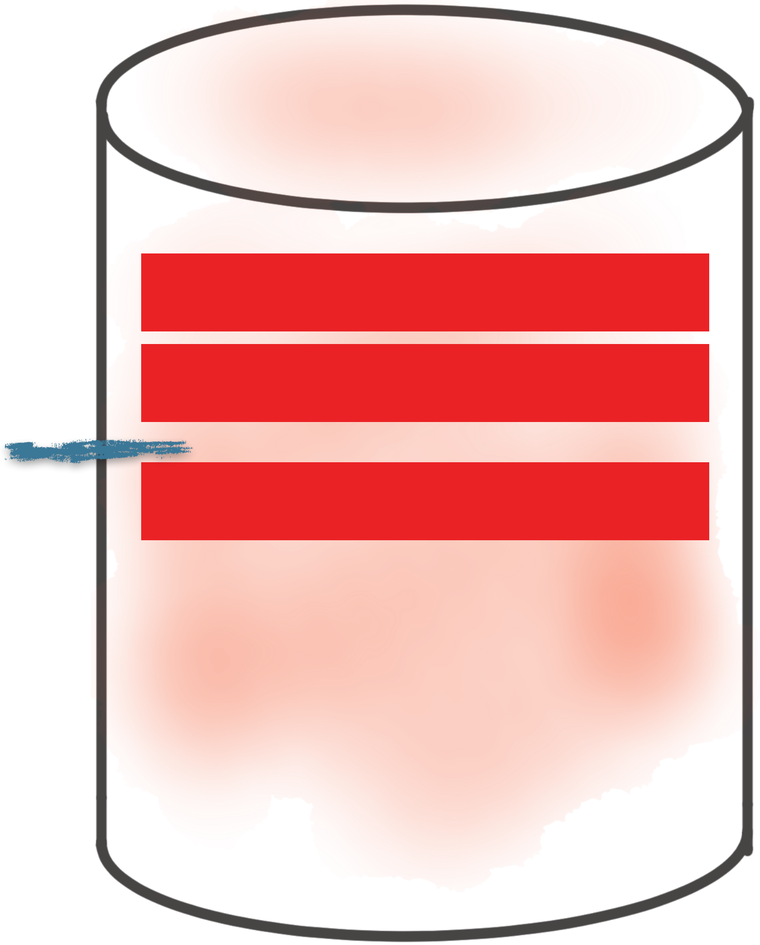
Query-based CDC



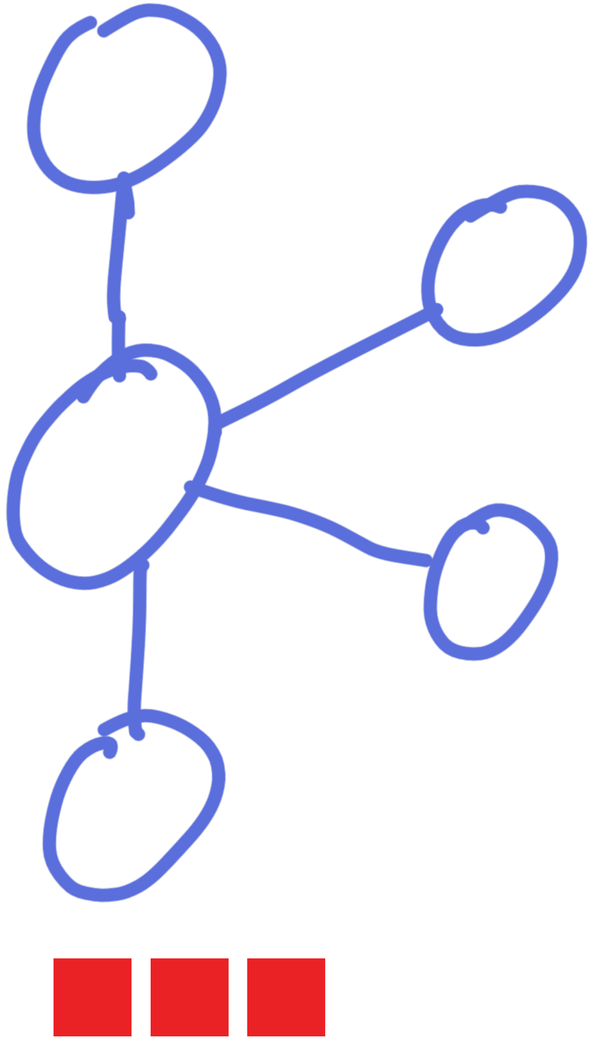
```
SELECT *
FROM   my_table
WHERE  ts_col > previous ts
```



Query-based CDC



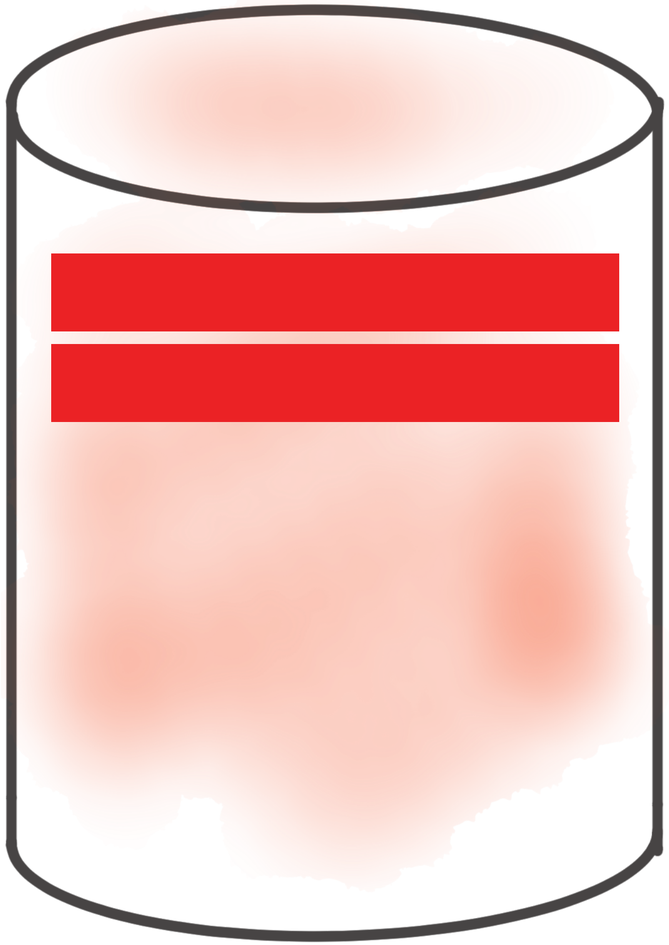
```
SELECT *
FROM   my_table
WHERE  ts_col >
          previous ts
```

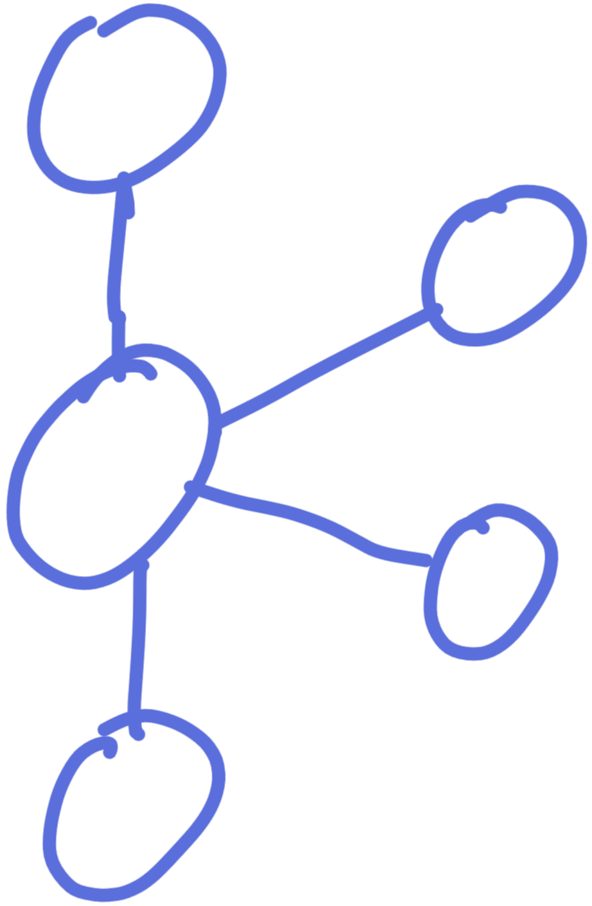


Log-based CDC

```
#051024 17:24:13 server id 1  end_log_pos 98
# Position  Timestamp   Type   Master ID        Size      Master Pos    Flags
# 00000004 9d fc 5c 43   0f   01 00 00 00   5e 00 00 00   62 00 00 00   00 00
# 00000017 04 00 35 2e 30 2e 31 35  2d 64 65 62 75 67 2d 6c |..5.0.15.debug.l|
# 00000027 6f 67 00 00 00 00 00 00  00 00 00 00 00 00 00 00 |og..............|
# 00000037 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 |................|
# 00000047 00 00 00 00 9d fc 5c 43  13 38 0d 00 08 00 12 00 |.......C.8......|
# 00000057 04 04 04 04 12 00 00 4b  00 04 1a               |.......K...|
```

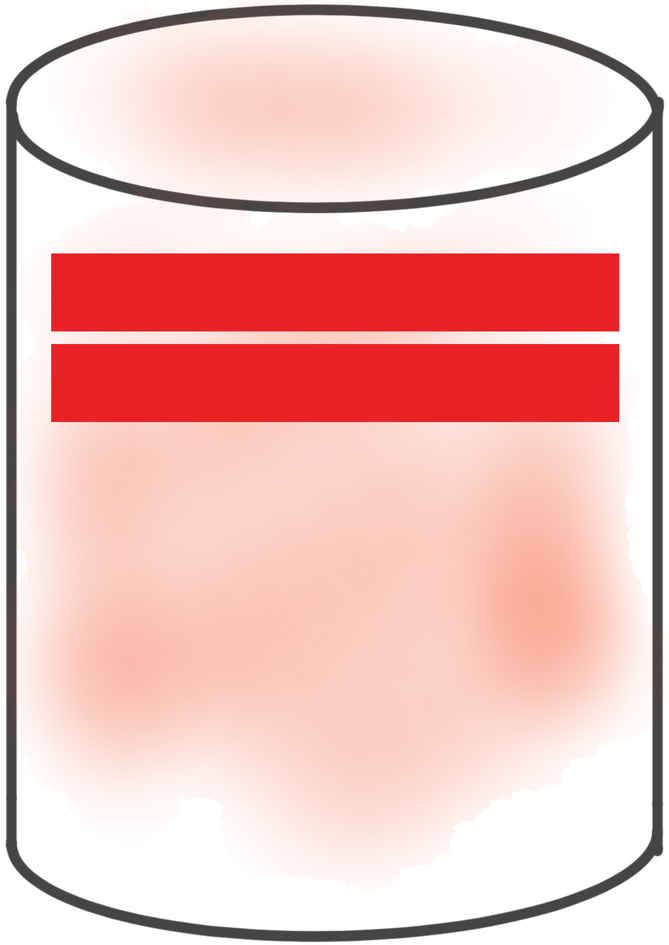
Log-based CDC

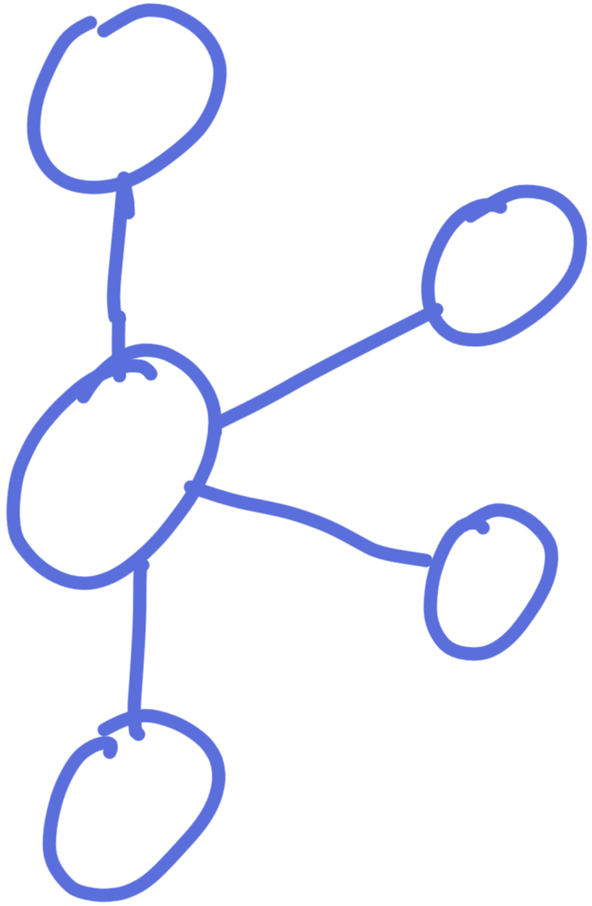




Transaction log

Log-based CDC

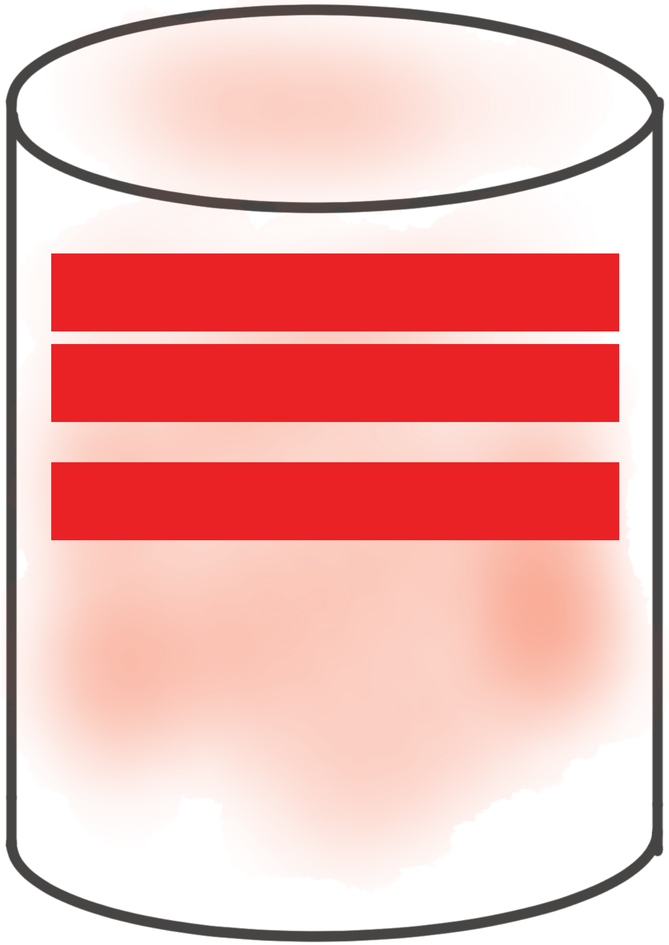


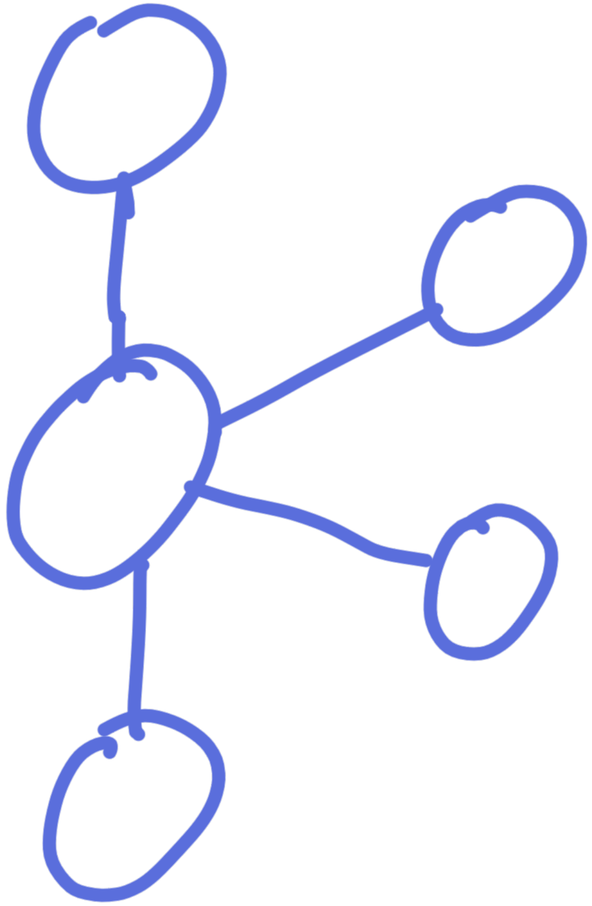




Transaction log

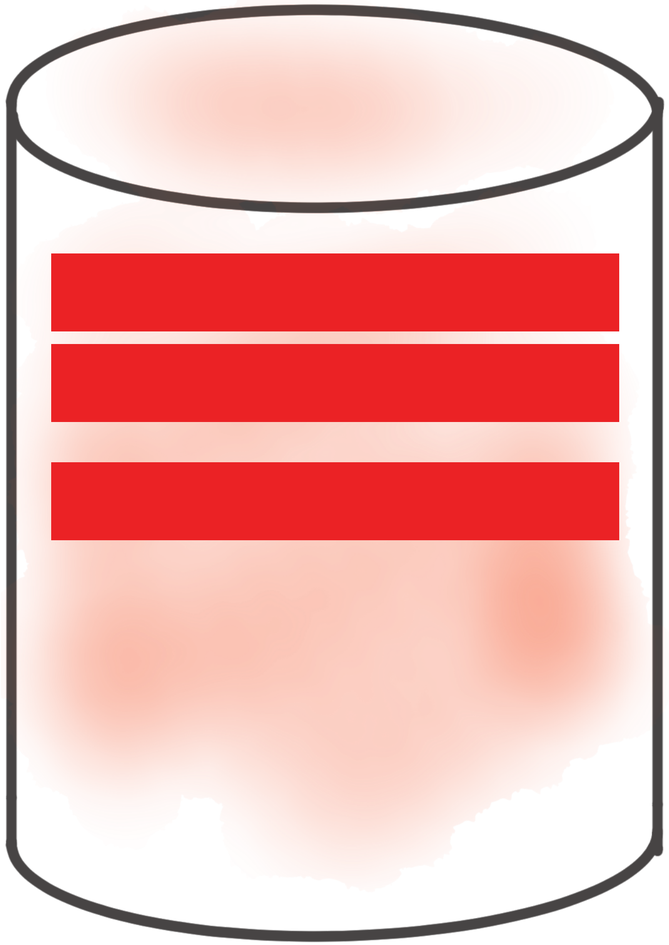
Log-based CDC

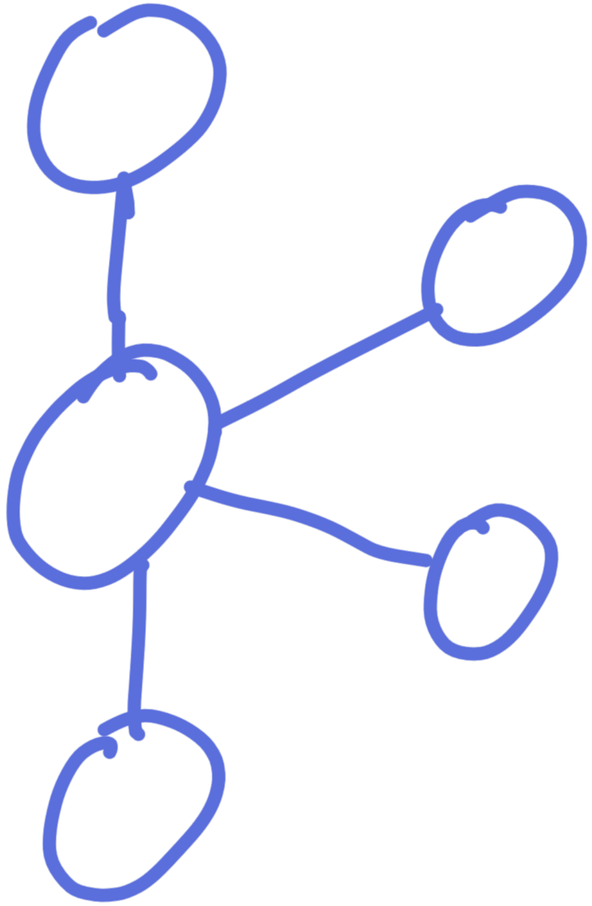




Transaction log

Log-based CDC







Transaction log

Demo Time!



https://rmoff.dev/cdc-demo


Query-based CDC
Log-based CDC
imgflip.com
JAKE-CLARK.TUMBLR

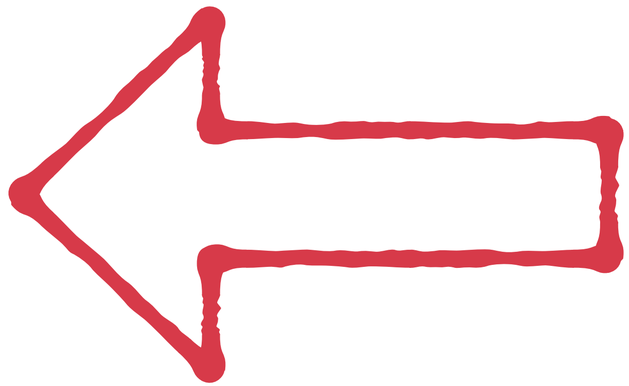
Query-based CDC

```
SELECT *
FROM   my_table
WHERE  ts_col > previous ts
```

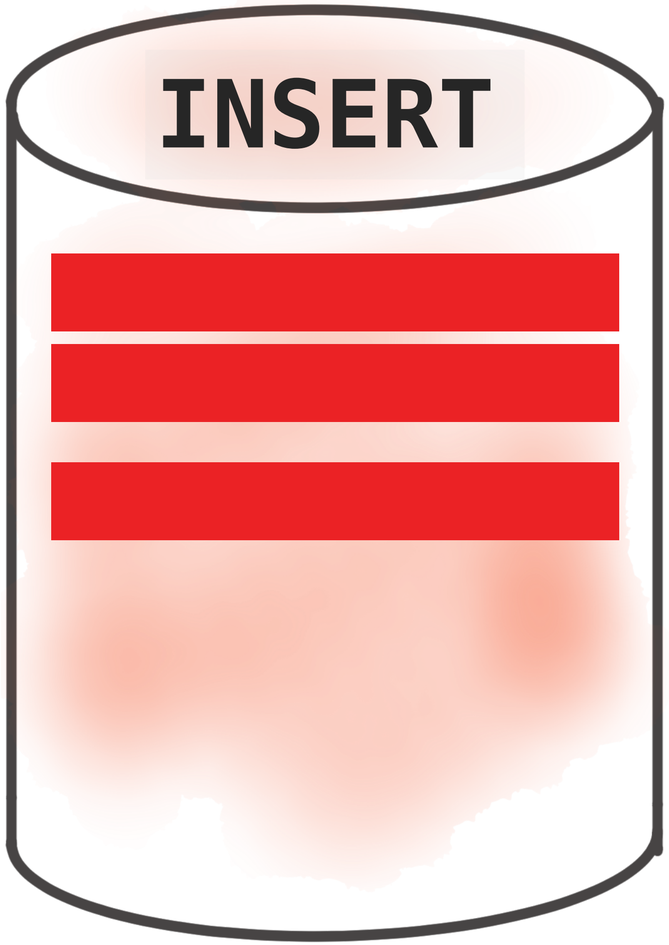
Query-based CDC

```
SELECT *
FROM   my_table
WHERE  ts_col > previous ts
```

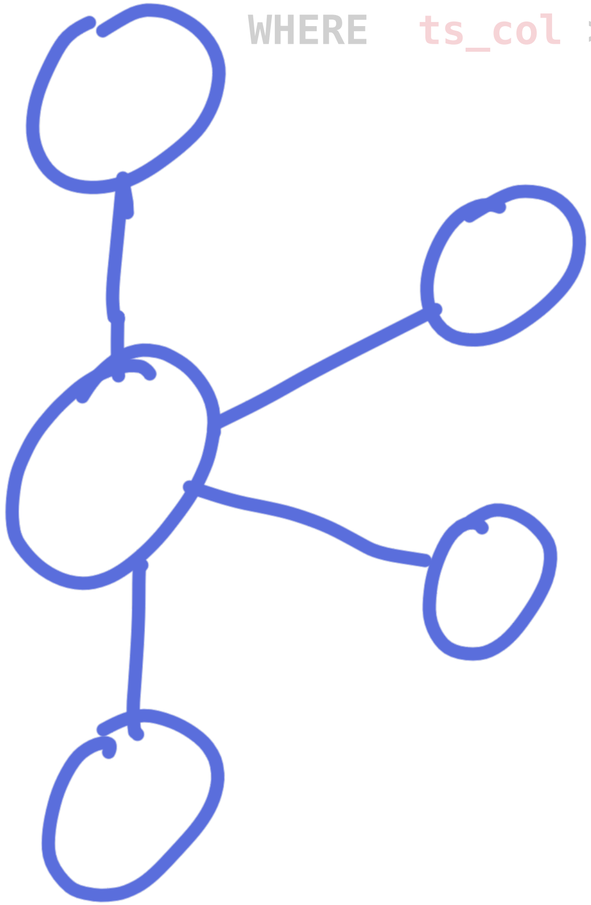
```
CREATE TABLE my_table (
    ID INT,
    FOO VARCHAR,
    BAR VARCHAR,
    WIBBLE VARCHAR,
    TS_COL TIMESTAMP
      DEFAULT CURRENT_TIMESTAMP
      ON UPDATE CURRENT_TIMESTAMP
)
```



Query-based CDC

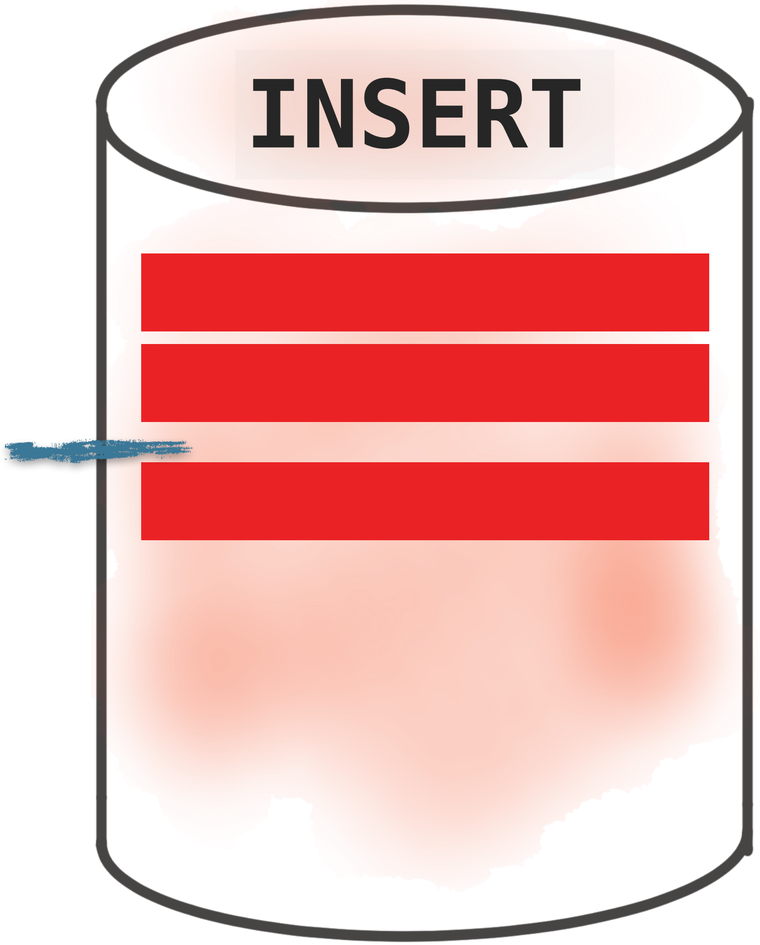



```
SELECT *
FROM   my_table
WHERE  ts_col > previous ts
```

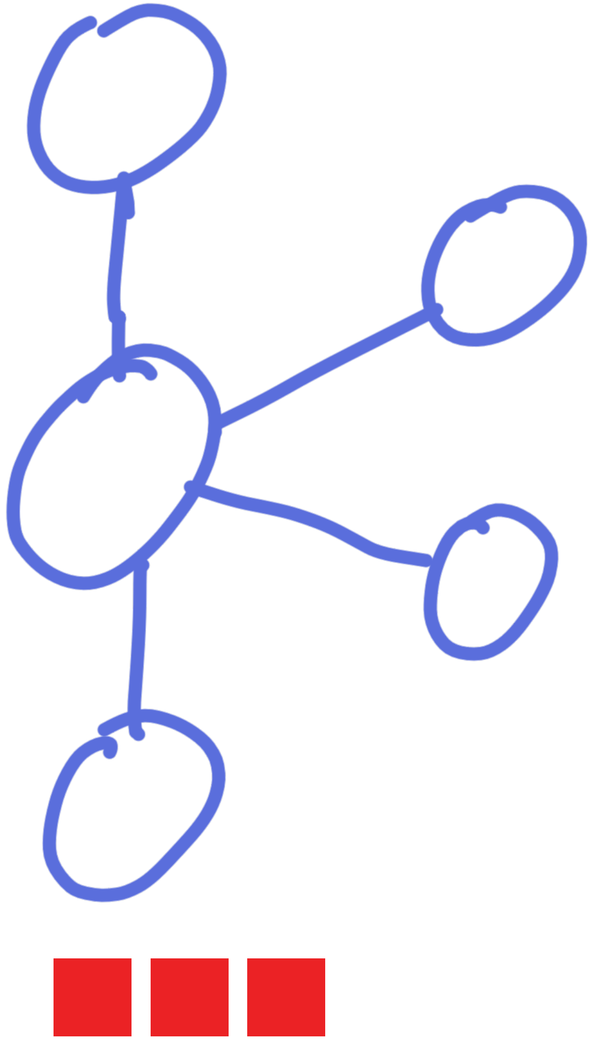




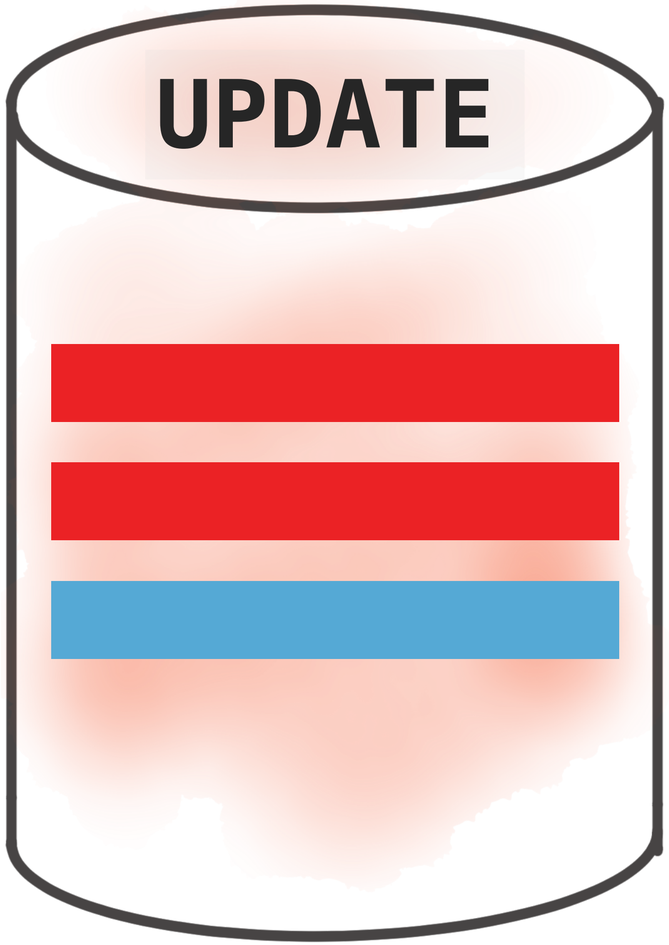
Query-based CDC

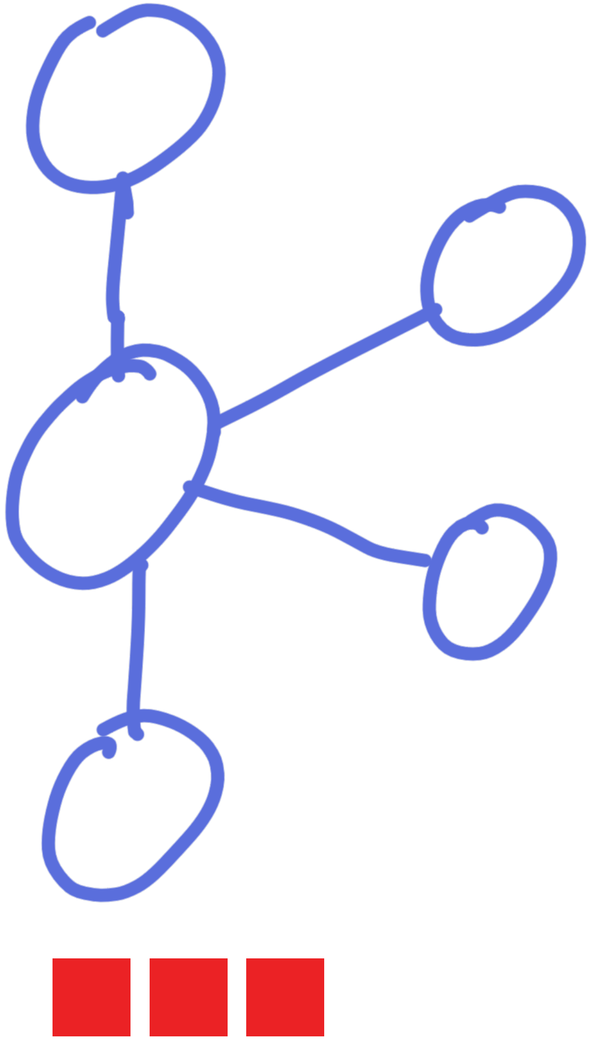



```
SELECT *
FROM   my_table
WHERE  ts_col >
          previous ts
```

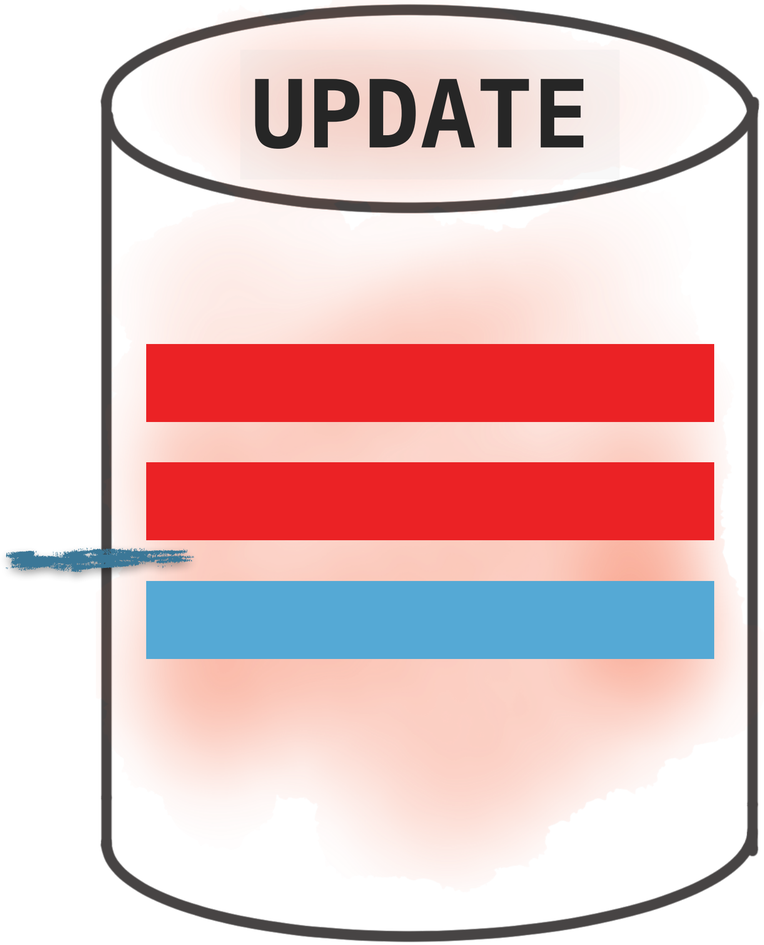


Query-based CDC

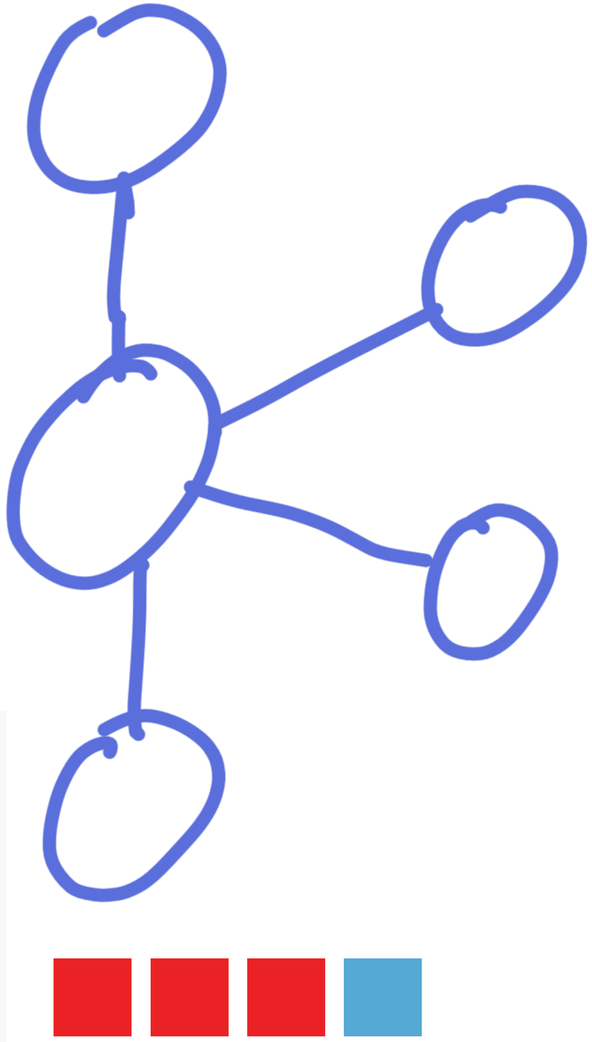




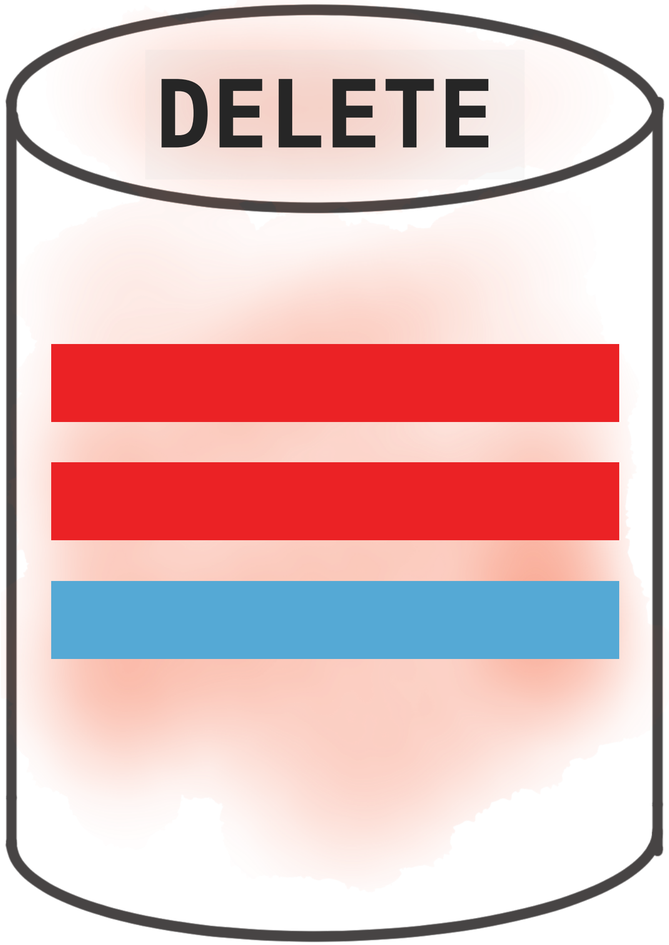
Query-based CDC

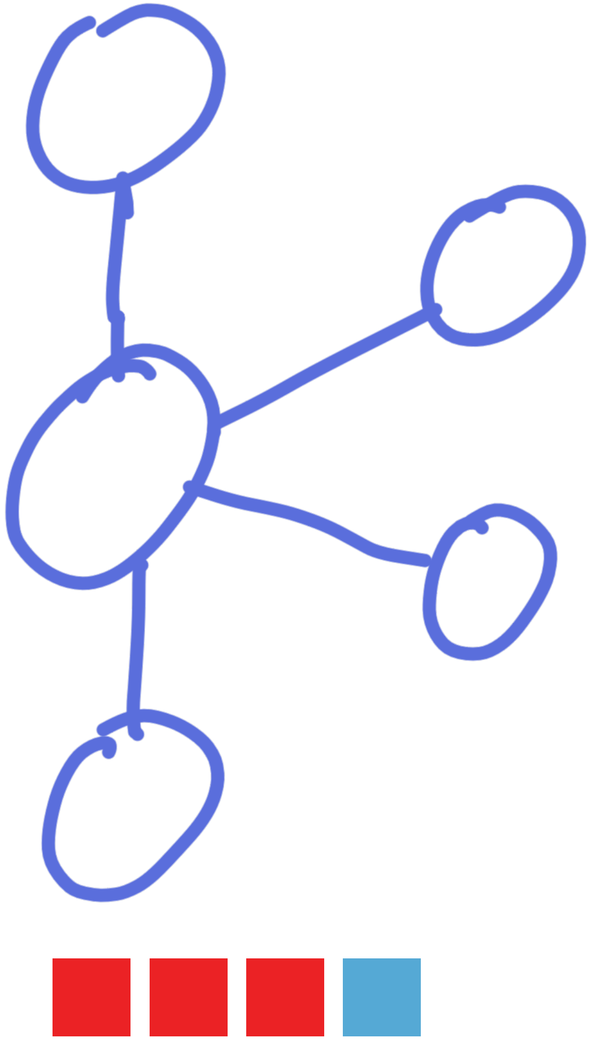



```
SELECT *
FROM   my_table
WHERE  ts_col >
         previous ts
```

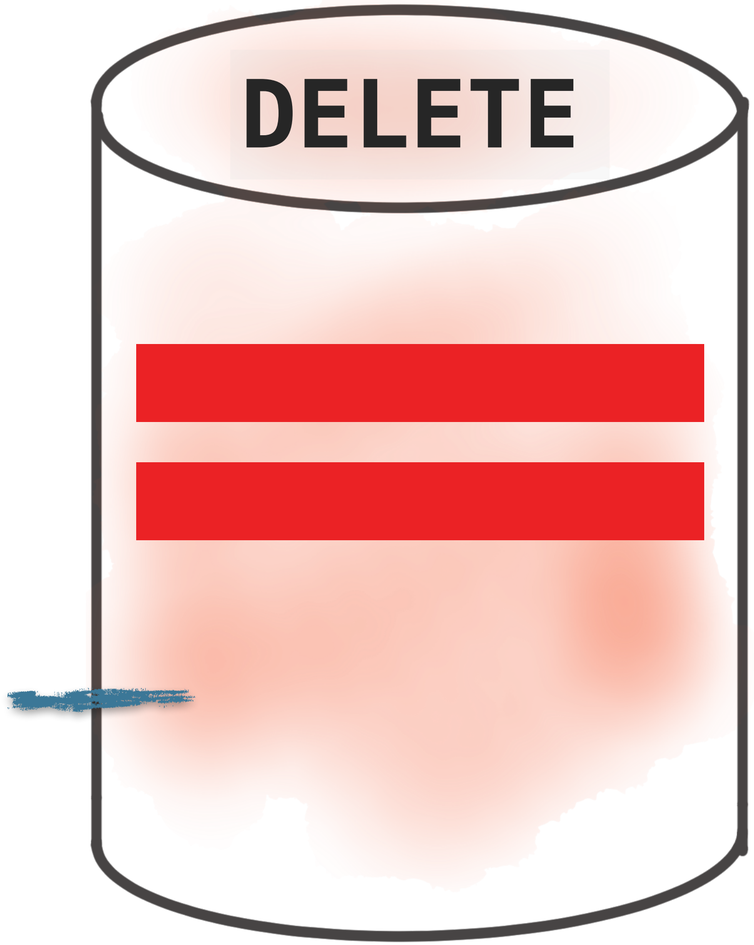


Query-based CDC

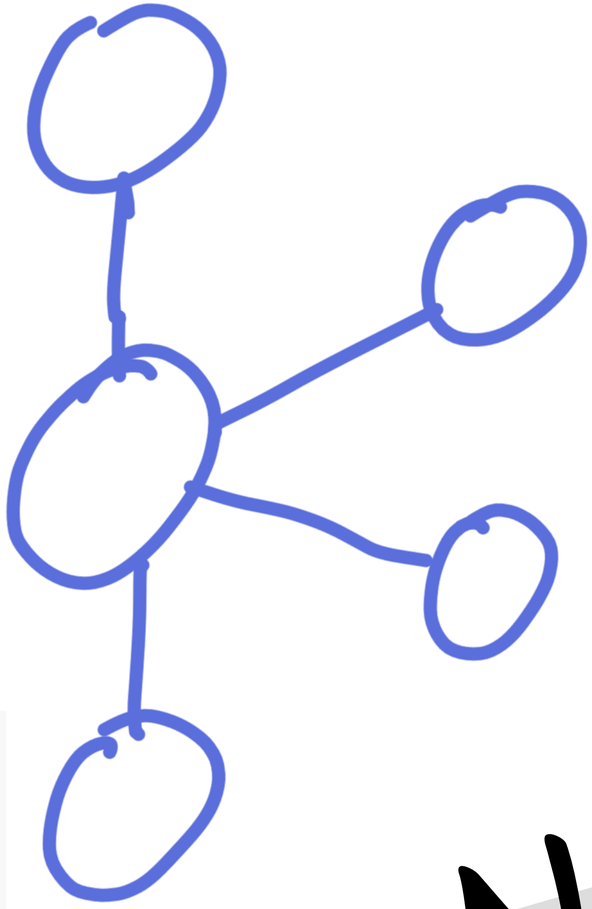




Query-based CDC




```
SELECT *
FROM   my_table
WHERE  ts_col >
         previous ts
```





Nope

Query-based CDC

orderID	status	address	updateTS



```
SELECT *
FROM   orders
WHERE  updateTS >
         previous ts
```

⏰now = 10:54:00
previous ts = 10:53:30

Query-based CDC

orderID	status	address	updateTS
42	SHIPPED	29 Acacia Road	10:54:29

```
SELECT *
FROM   orders
WHERE  updateTS >
         previous ts
```

⏰now = 10:54:30
previous ts = 10:54:00

```
{
  "orderID": 42,
  "status": "SHIPPED",
  "address": "29 Acacia Road",
  "updateTS": "10:54:29"
}
```

Query-based CDC

orderID	status	address	updateTS
42	PENDING	--	10:54:01

```
INSERT INTO orders (orderID, status)
VALUES (42, 'PENDING');
```

Query-based CDC

orderID	status	address	updateTS
42	PENDING	1640 Riverside Drive	10:54:20

```
UPDATE orders
SET address = '1640 Riverside Drive'
WHERE orderID = 42;
```

Query-based CDC

orderID	status	address	updateTS
42	PENDING	29 Acacia Road	10:54:25

```
UPDATE orders
SET address = '29 Acacia Road'
WHERE orderID = 42;
```

Query-based CDC

orderID	status	address	updateTS
42	SHIPPED	29 Acacia Road	10:54:29

```
UPDATE orders
SET status = 'SHIPPED'
WHERE orderID = 42;
```

Query-based CDC

orderID	status	address	updateTS
42	PENDING	--	10:54:01
42	PENDING	1640 Riverside Drive	10:54:20
42	PENDING	29 Acacia Road	10:54:25
42	SHIPPED	29 Acacia Road	10:54:29

Query-based CDC

orderID	status	address	updateTS
42	SHIPPED	29 Acacia Road	10:54:29

```
SELECT *
FROM   orders
WHERE  updateTS >
         previous ts
```

⏰now = 10:54:30
previous ts = 10:54:00

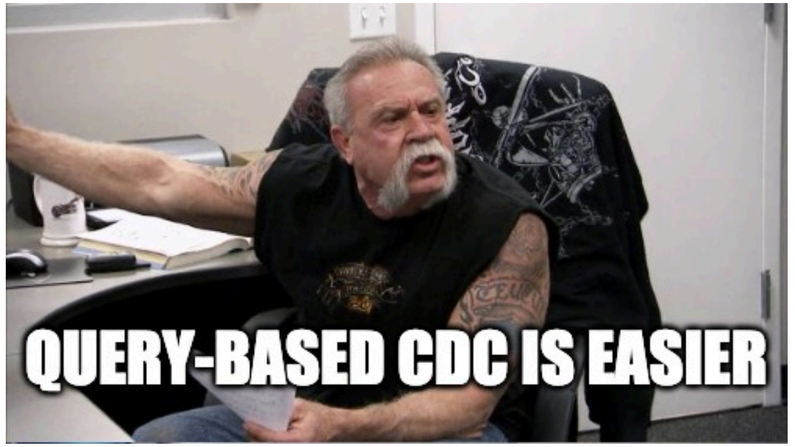
```
{
  "orderID": 42,
  "status": "SHIPPED",
  "address": "29 Acacia Road",
  "updateTS": "10:54:29"
}
```

Query-based CDC

- ✅ Usually easier to setup
- ✅ Requires fewer permissions
- 🛑 Needs specific columns in source schema to track changes
- 🛑 Impact of polling the DB (or higher latencies tradeoff)
- 🛑 Can't track deletes
- 🛑 Can't track multiple events between polling interval



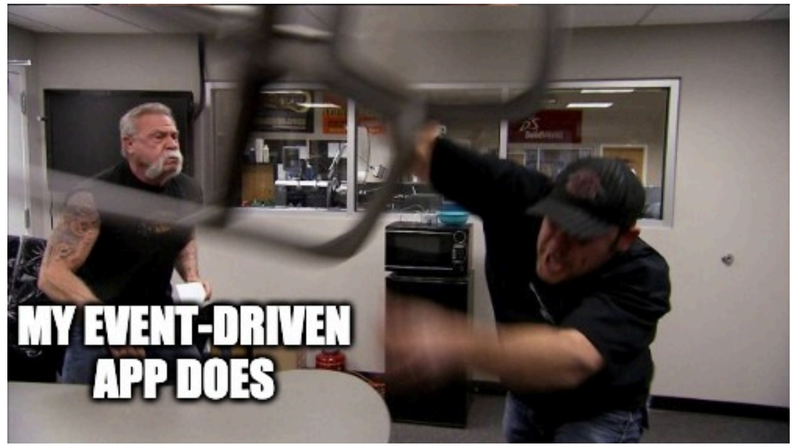
Photo by Matese Fields on Unsplash


QUERY-BASED CDC IS EASIER

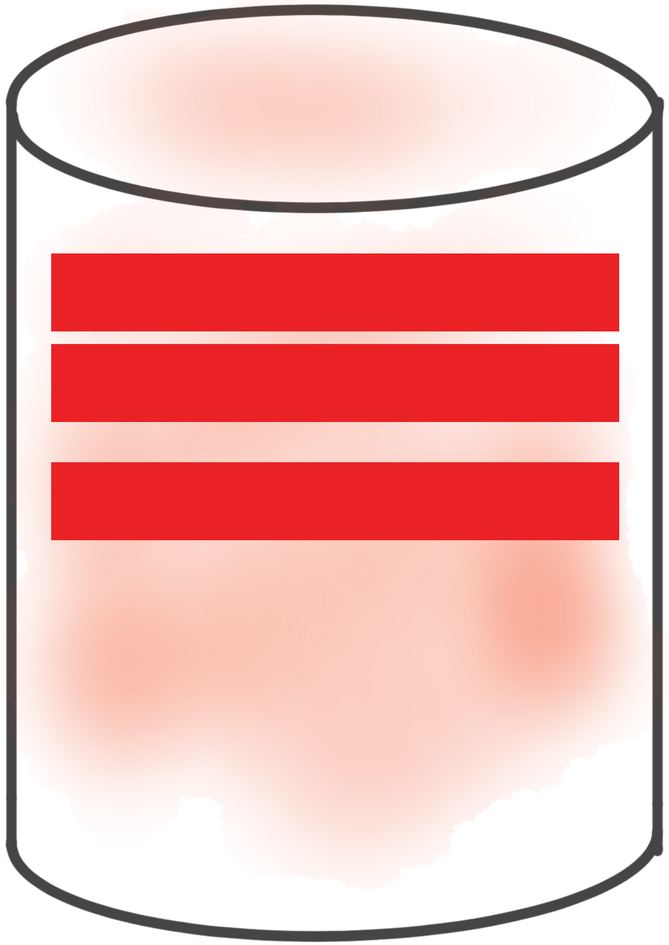

WHO NEEDS
ALL THE EVENTS?!

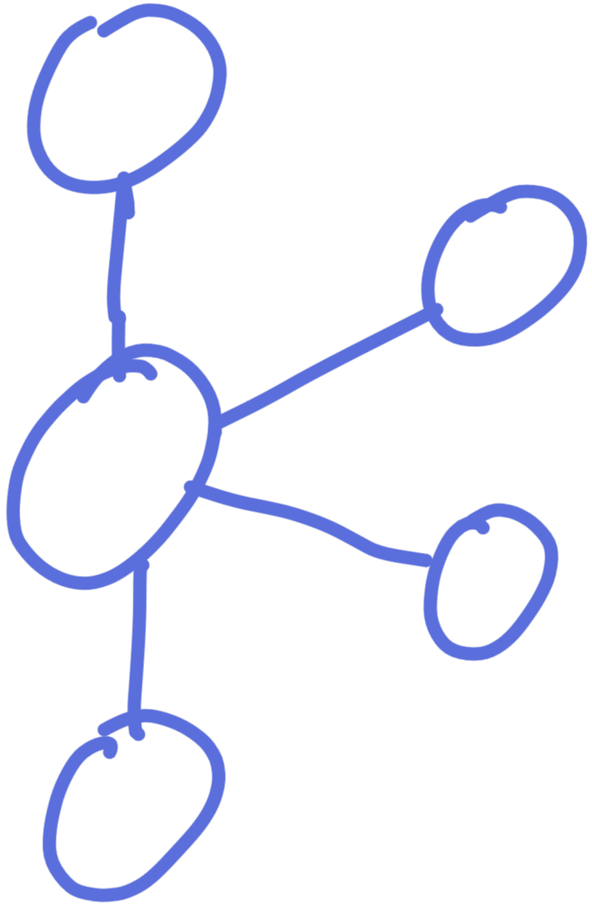

USE LOG-BASED CDC THEN!
imgflip.com


BUT IT DOESN'T
CAPTURE *ALL* THE EVENTS


MY EVENT-DRIVEN
APP DOES

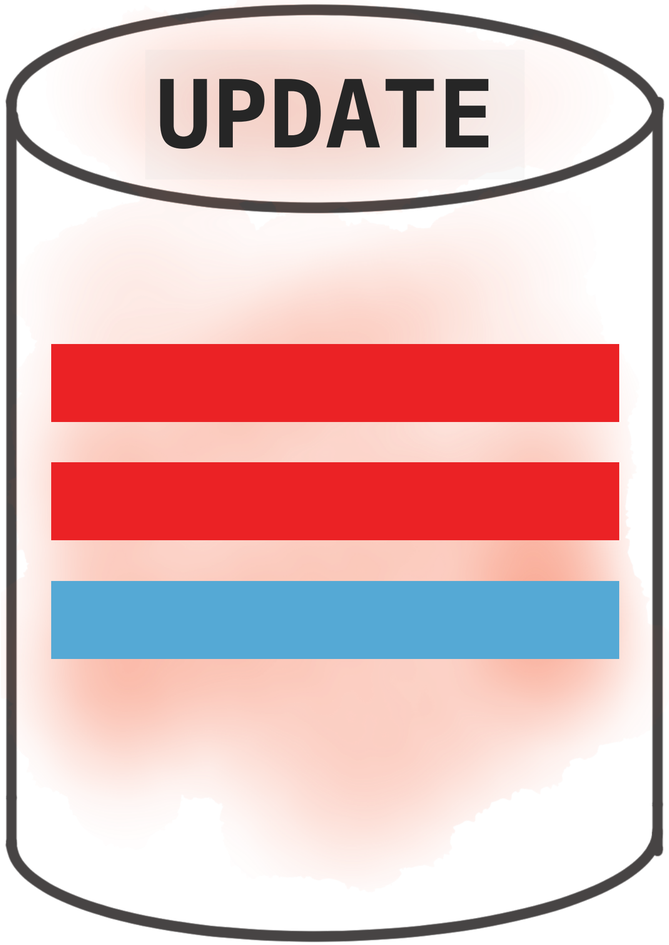
Log-based CDC





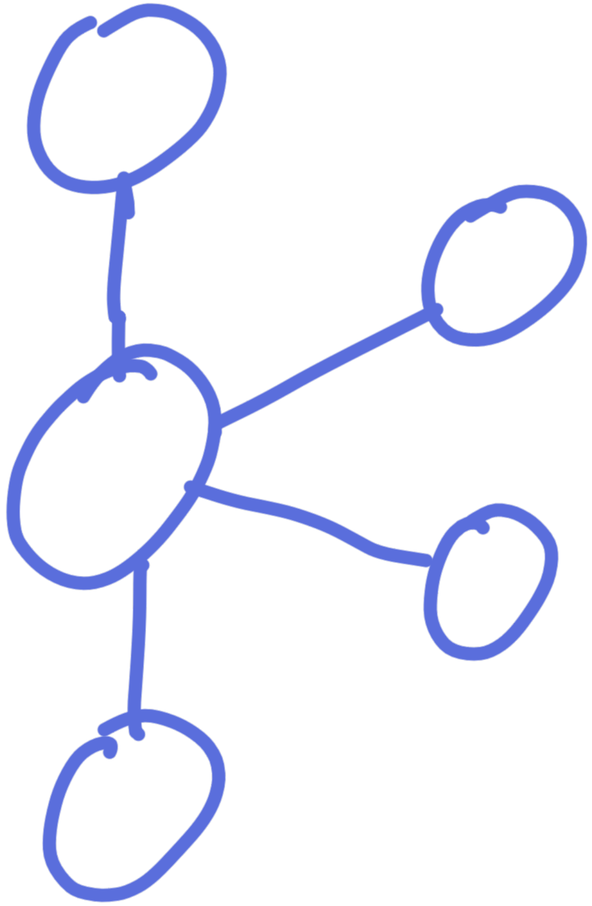
Transaction log

Log-based CDC



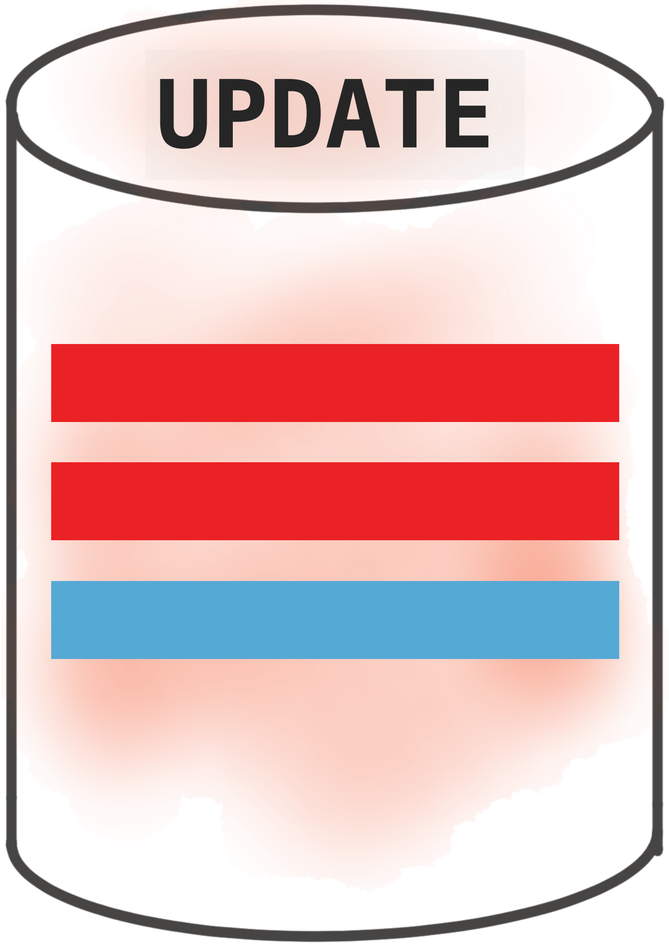


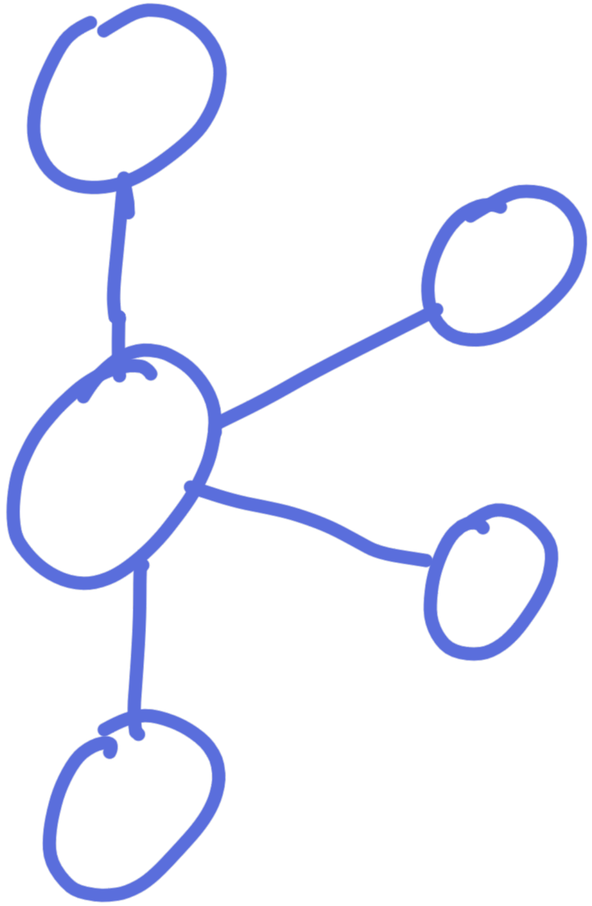
Transaction log





Log-based CDC





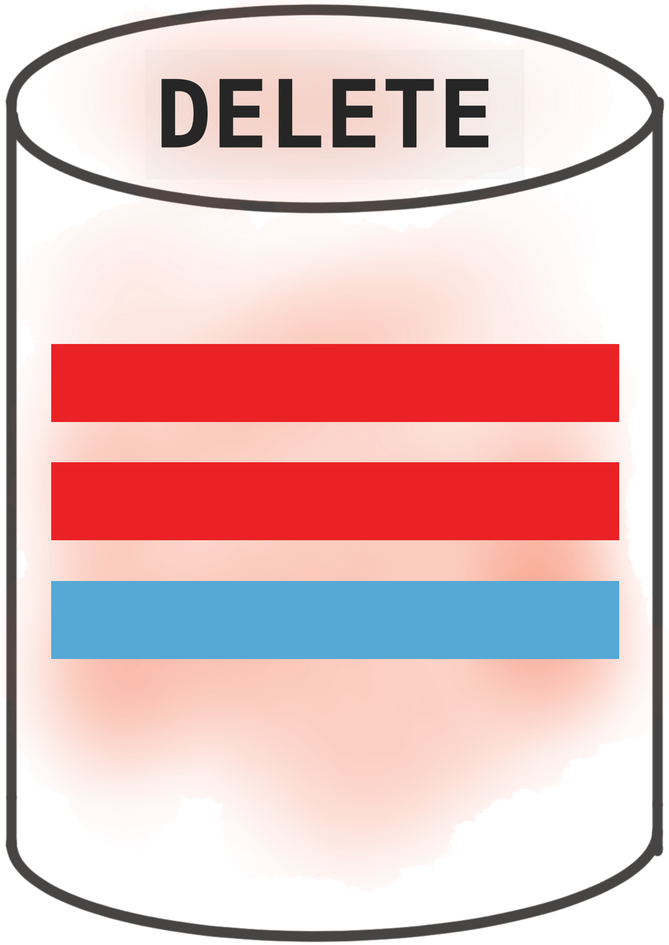


Transaction log



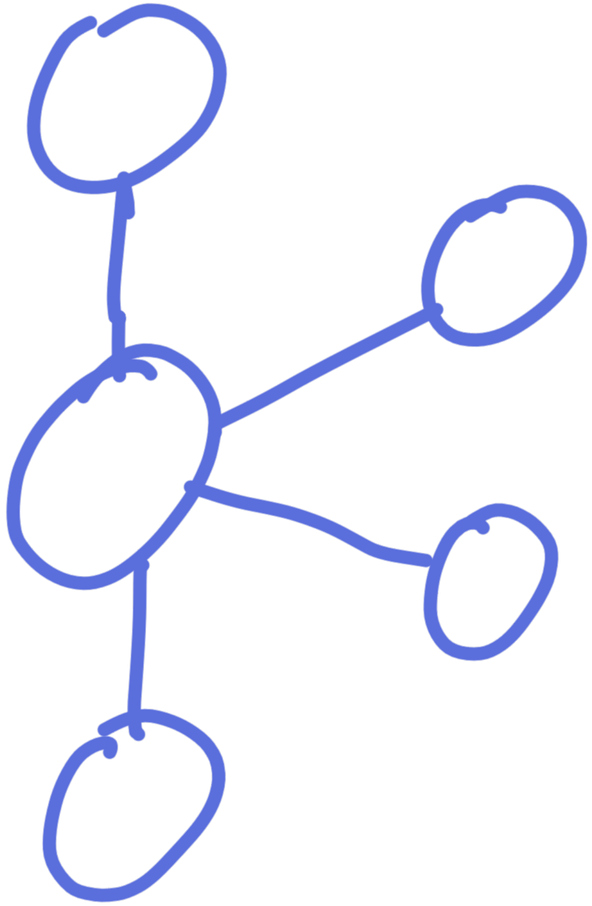


Log-based CDC



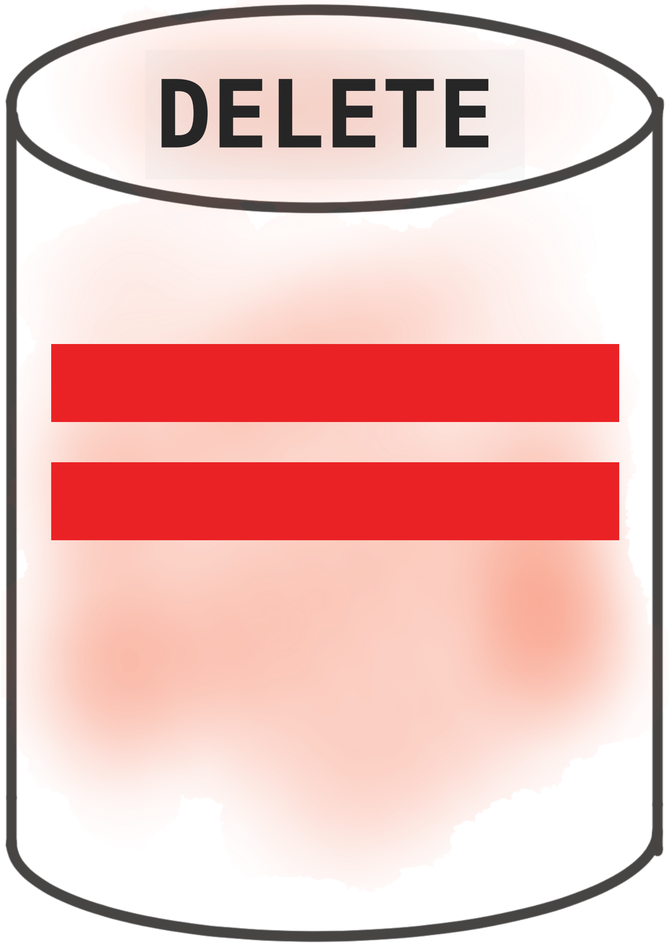


Transaction log



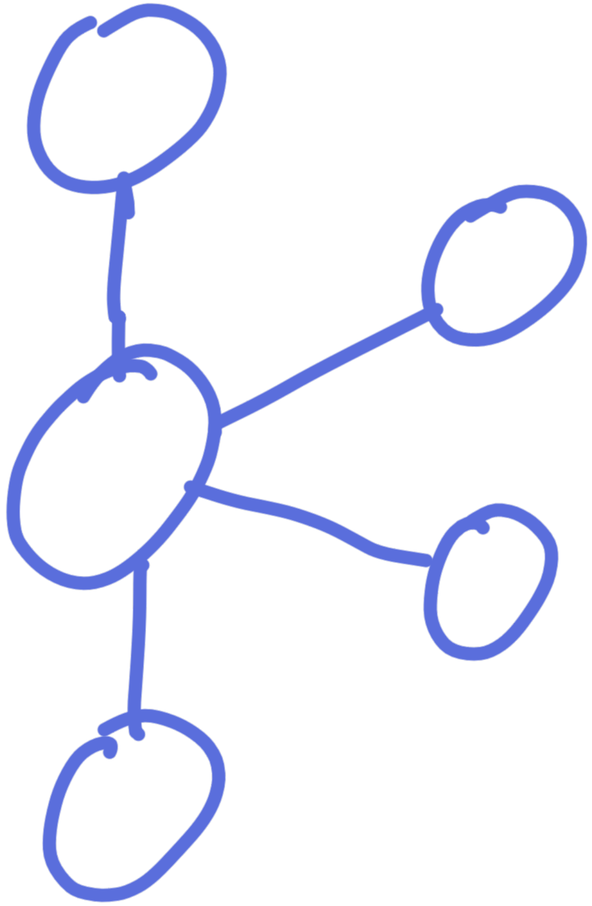


Log-based CDC



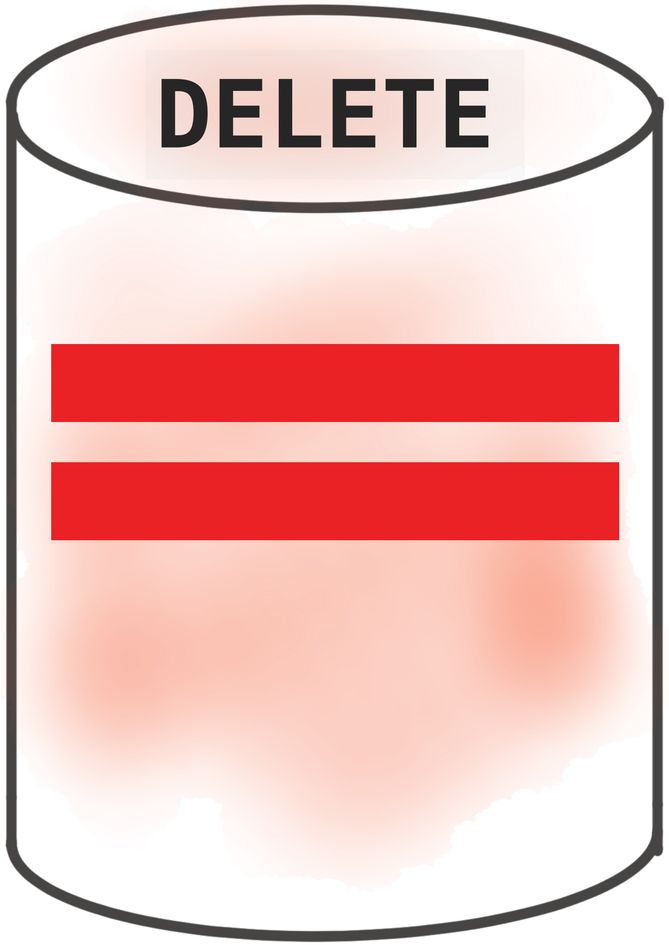


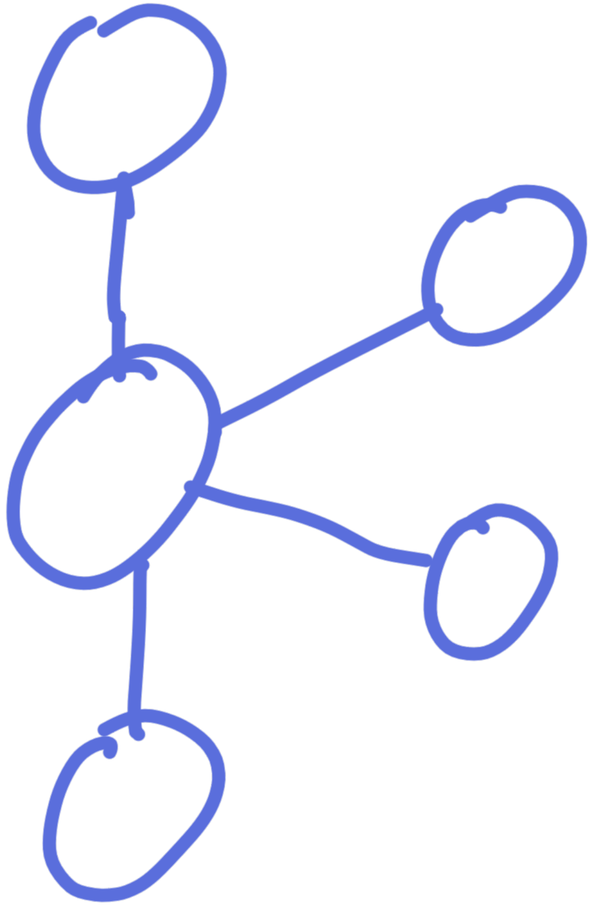
Transaction log





Log-based CDC





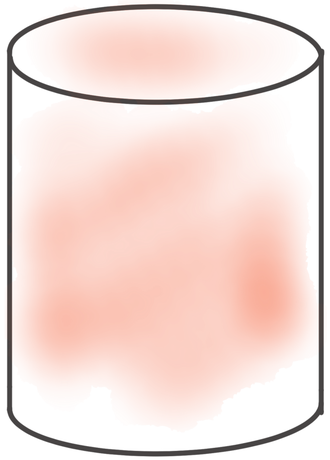


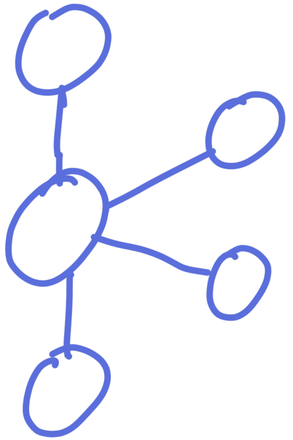
Transaction log





Log-based CDC









Commit log



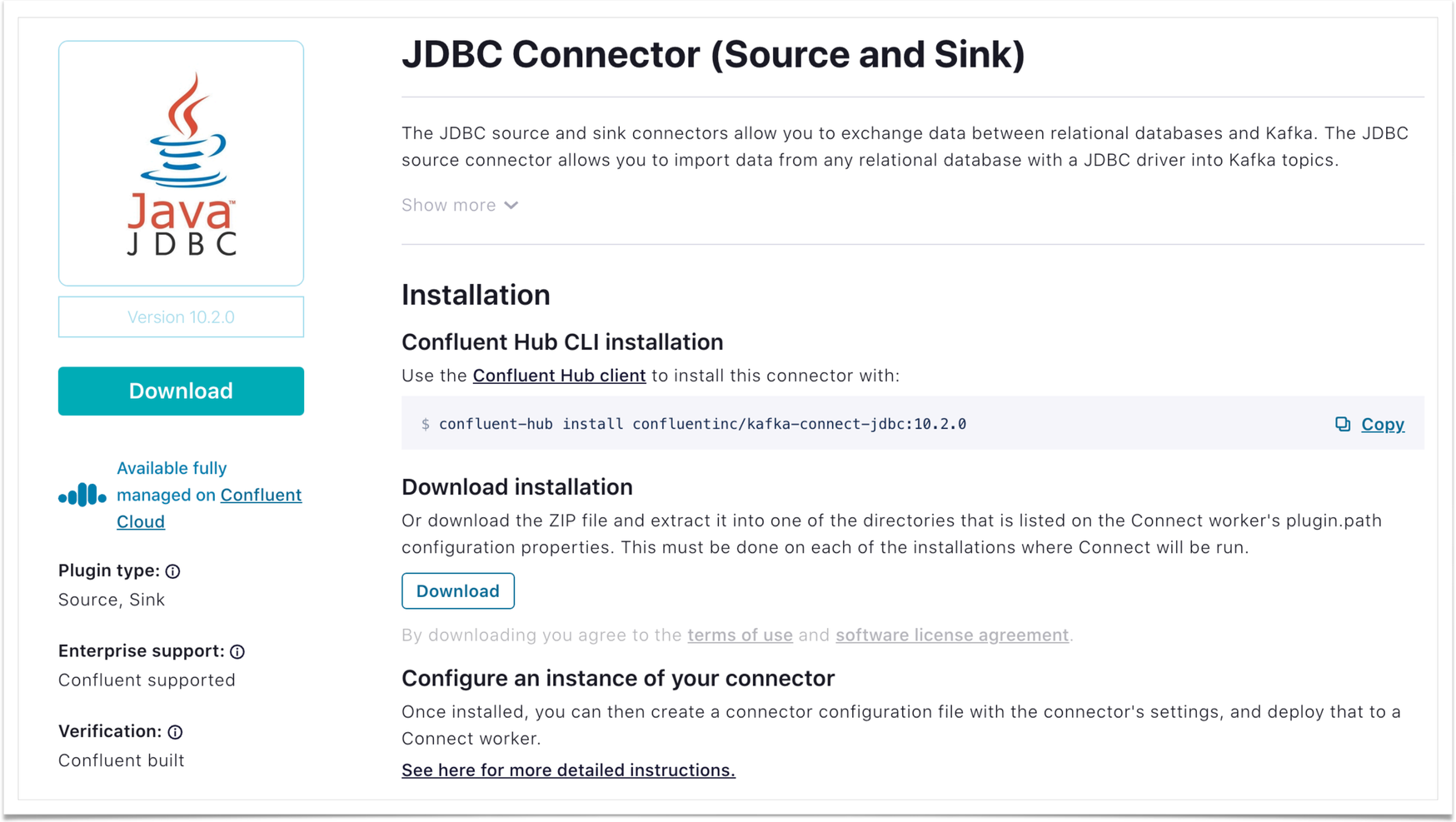
Log-based CDC

- ✅ 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

tl;dr : which tool do I use?

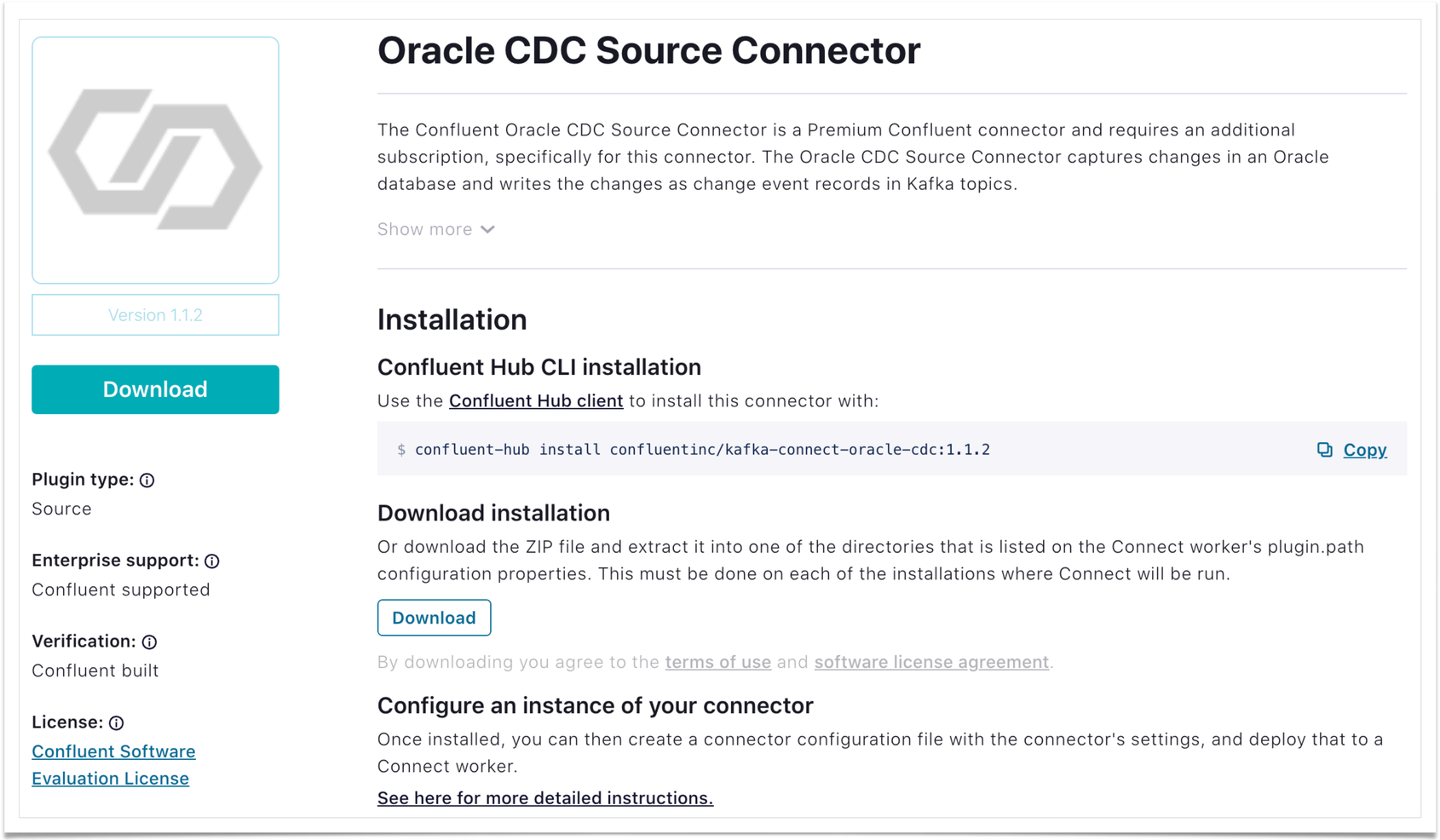
Query-based CDC:



confluent.io/hub

tl;dr : which tool do I use?

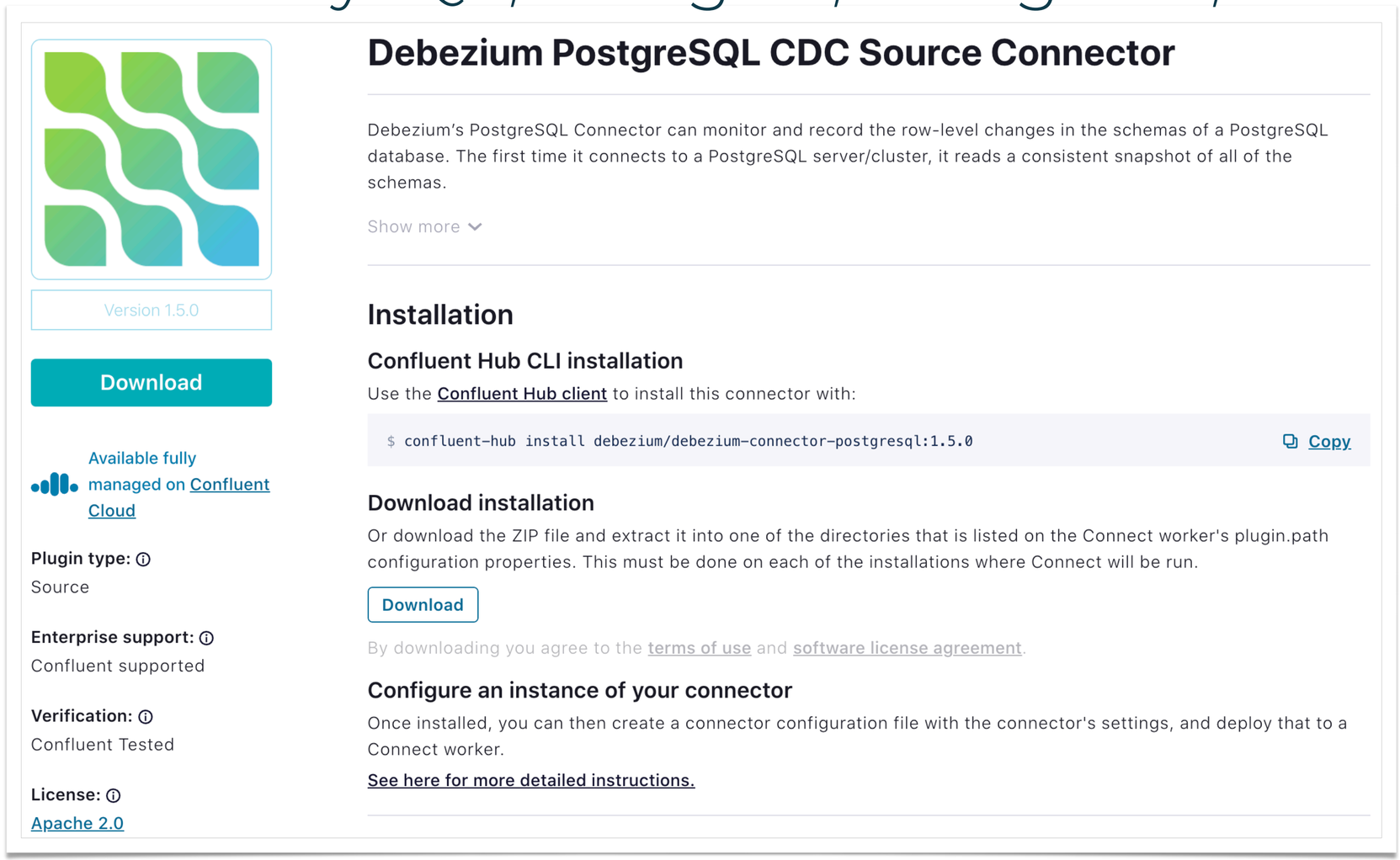
Log-based CDC - Oracle



confluent.io/hub

tl;dr : which tool do I use?

Log-based CDC - MySQL, Postgres, MongoDB, etc



confluent.io/hub

Want to learn more?



CONFLUENT

CTAs, not CATs
(sorry, not sorry)


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

#EOF

@rmoff rmoff.dev/talks youtube.com/rmoff