ATM Fraud Detection

with Apache Kafka and KSQL

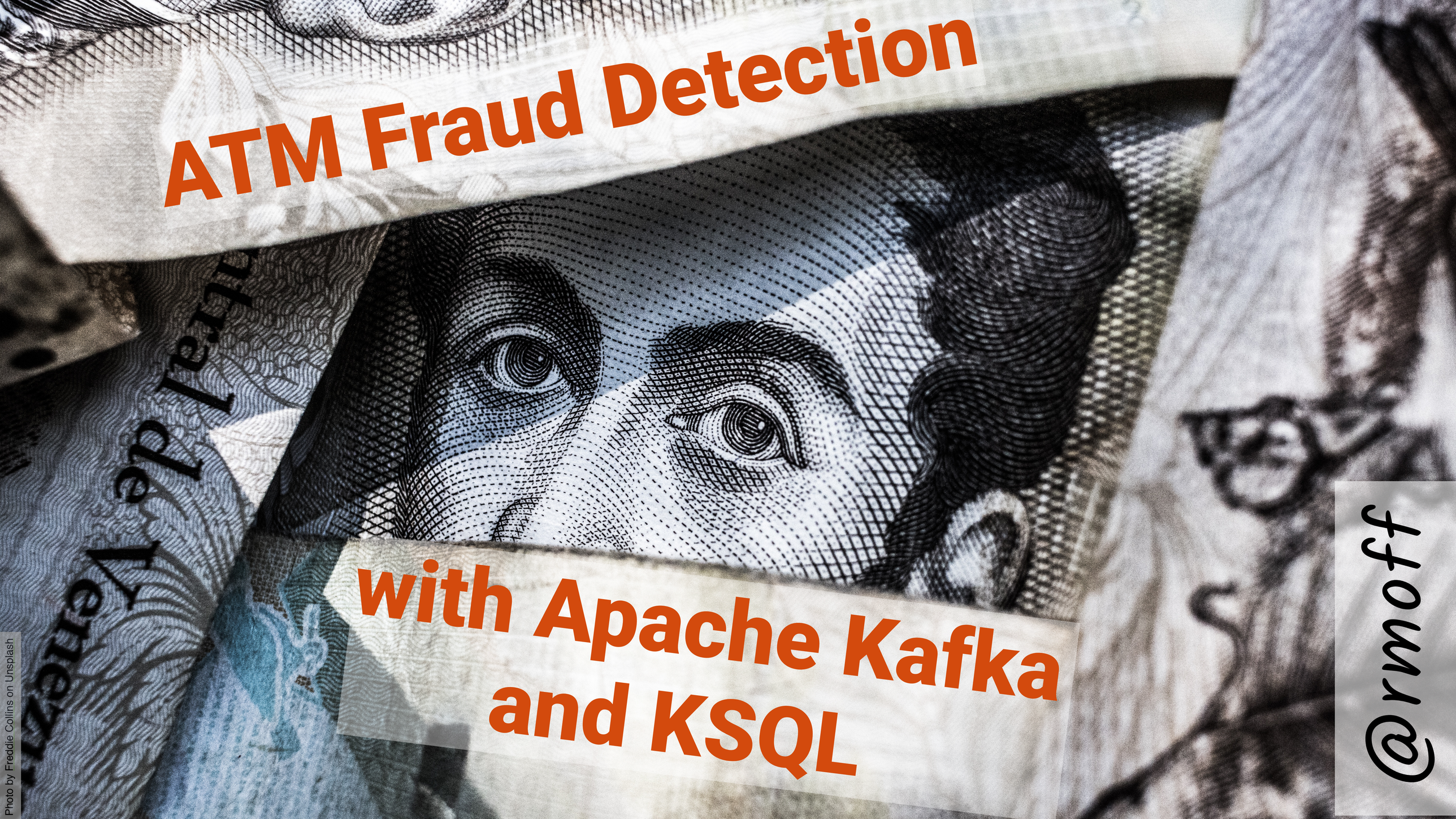


Photo by Freddie Collins on Unsplash

@rmoff

Photo by Mirza Babic on Unsplash

Photo by Lasaye Hommes on Unsplash

{"account_id": "a267", "timestamp": "2018-11-23 17:10:07 +0000", "atm": "ATM : 5229255383", "amount": 400, "location
: {"lat": "53.8081494", "lon": "-1.7311409"}, "transaction_id": "a07a8f7a-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a366", "timestamp": "2018-11-23 17:10:08 +0000", "atm": "Yorkshire Building Society", "amount": 200,
"location": {"lat": "53.7961087", "lon": "-1.5423515"}, "transaction_id": "a0ed5aa0-ef42-11e8-b79a-0242c0a80003"}
% Reached end of topic atm_txns_gess [0] at offset 27195
{"account_id": "a892", "timestamp": "2018-11-23 17:10:08 +0000", "atm": "ATM : 333467278", "amount": 20, "location":
{"lat": "53.7972381", "lon": "-1.542482"}, "transaction_id": "a160558c-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a515", "timestamp": "2018-11-23 17:10:09 +0000", "atm": "Barclays Bank PLC", "amount": 20, "location"
: {"lat": "53.7250925", "lon": "-1.8882749"}, "transaction_id": "a1d314e6-ef42-11e8-b79a-0242c0a80003"}
% Reached end of topic atm_txns_gess [0] at offset 27197
{"account_id": "a699", "timestamp": "2018-11-23 17:10:10 +0000", "atm": "Post Office", "amount": 50, "location": {"la
t": "53.87419", "lon": "-1.7128211"}, "transaction_id": "a245d1b6-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a499", "timestamp": "2018-11-23 17:10:11 +0000", "atm": "ATM : 5792055772", "amount": 400, "location"
: {"lat": "53.8357238", "lon": "-1.7965487"}, "transaction_id": "a2b8ab0a-ef42-11e8-b79a-0242c0a80003"}
% Reached end of topic atm_txns_gess [0] at offset 27199
{"account_id": "a536", "timestamp": "2018-11-23 17:10:11 +0000", "atm": "RBS", "amount": 20, "location": {"lat": "53.
7988921", "lon": "-1.547186"}, "transaction_id": "a32b8472-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a437", "timestamp": "2018-11-23 17:10:12 +0000", "atm": "Yorkshire Bank", "amount": 20, "location": {
"lat": "53.8203297", "lon": "-1.5765871"}, "transaction_id": "a39e48a4-ef42-11e8-b79a-0242c0a80003"}
% Reached end of topic atm_txns_gess [0] at offset 27201
{"account_id": "a437", "timestamp": "2018-11-23 17:01:18 +0000", "atm": "ATM : 1414423006", "amount": 400, "location"
: {"lat": "53.710754", "lon": "-2.101069"}, "transaction_id": "xxxa39e48a4-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a884", "timestamp": "2018-11-23 17:10:13 +0000", "atm": "NatWest", "amount": 50, "location": {"lat":
"53.6554012", "lon": "-1.8164935"}, "transaction_id": "a41186e8-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a823", "timestamp": "2018-11-23 17:10:14 +0000", "atm": "ATM : 1704973341", "amount": 200, "location"
: {"lat": "53.6103183", "lon": "-1.7028805"}, "transaction_id": "a47f3b84-ef42-11e8-b79a-0242c0a80003"}
% Reached end of topic atm_txns_gess [0] at offset 27204

Demo!

Spot patterns within this stream

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

Spot patterns within this stream

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

Legit

Legit

Spot patterns within this stream

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

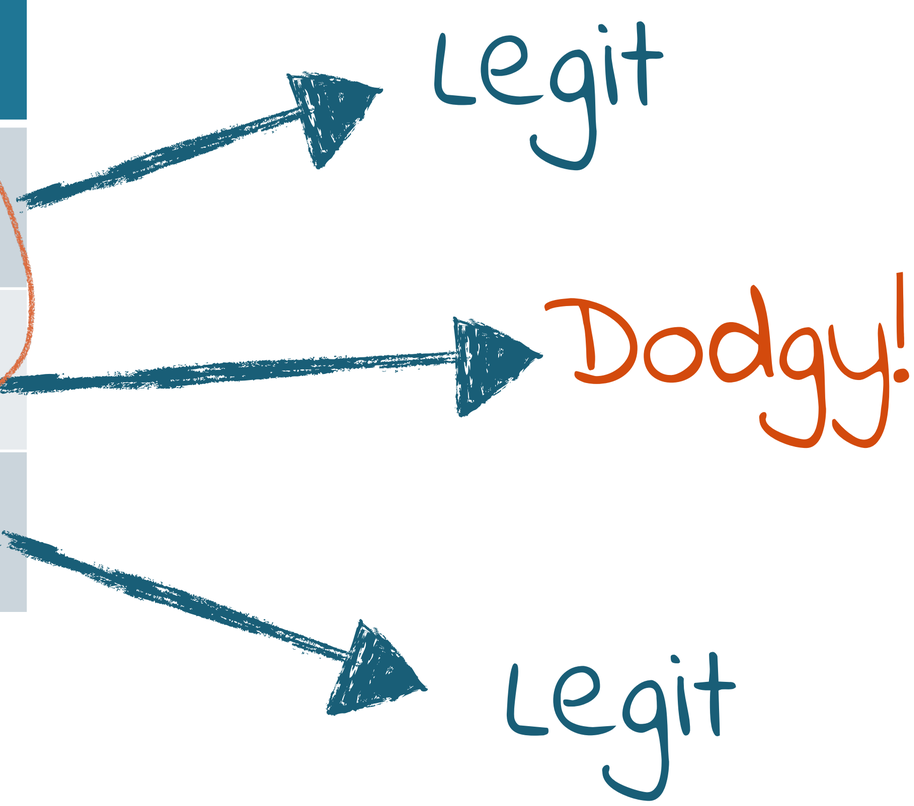
Legit

Dodgy!

Legit

Spot patterns within this stream

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds



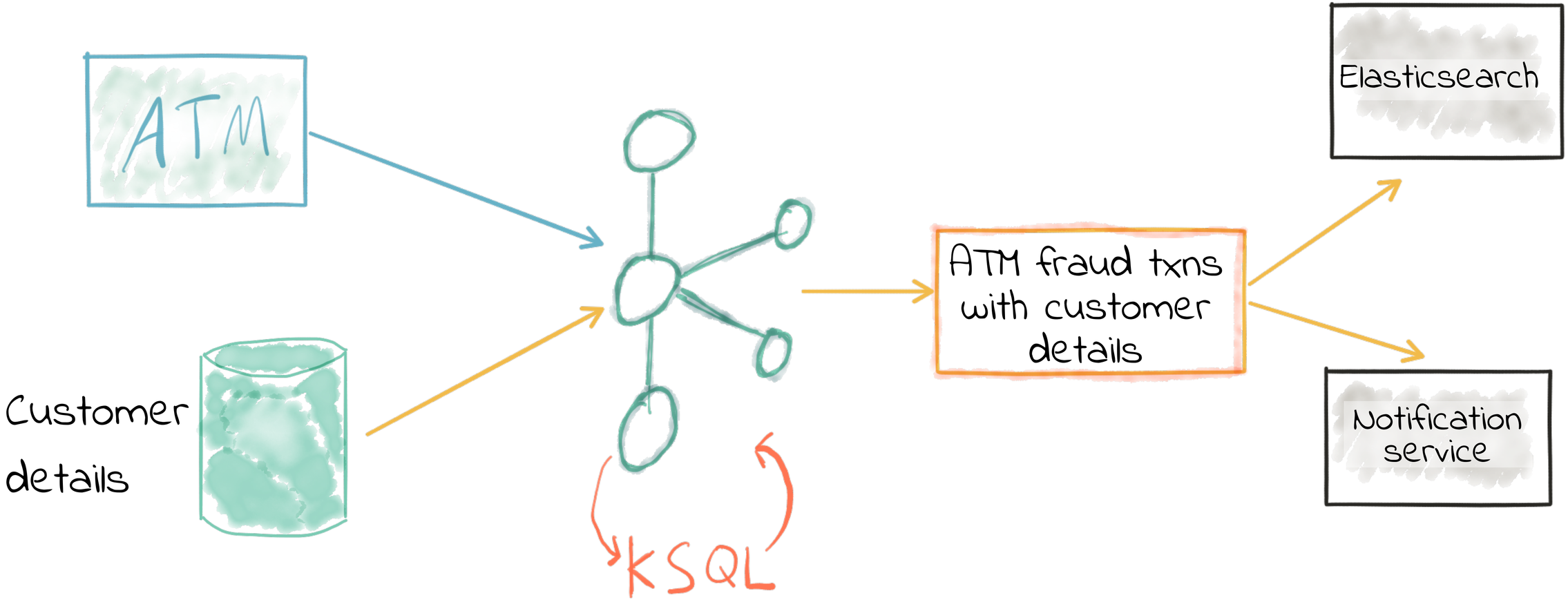
{"account_id": "a267", "timestamp": "2018-11-23 17:10:07 +0000", "atm": "ATM : 5229255383", "amount": 400, "location
: {"lat": "53.8081494", "lon": "-1.7311409"}, "transaction_id": "a07a8f7a-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a366", "timestamp": "2018-11-23 17:10:08 +0000", "atm": "Yorkshire Building Society", "amount": 200,
"location": {"lat": "53.7961087", "lon": "-1.5423515"}, "transaction_id": "a0ed5aa0-ef42-11e8-b79a-0242c0a80003"}
% Reached end of topic atm_txns_gess [0] at offset 27195
{"account_id": "a892", "timestamp": "2018-11-23 17:10:08 +0000", "atm": "ATM : 333467278", "amount": 20, "location":
{"lat": "53.7972381", "lon": "-1.542482"}, "transaction_id": "a160558c-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a515", "timestamp": "2018-11-23 17:10:09 +0000", "atm": "Barclays Bank PLC", "amount": 20, "location"
: {"lat": "53.7250925", "lon": "-1.8882749"}, "transaction_id": "a1d314e6-ef42-11e8-b79a-0242c0a80003"}
% Reached end of topic atm_txns_gess [0] at offset 27197
{"account_id": "a699", "timestamp": "2018-11-23 17:10:10 +0000", "atm": "Post Office", "amount": 50, "location": {"la
t": "53.87419", "lon": "-1.7128211"}, "transaction_id": "a245d1b6-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a499", "timestamp": "2018-11-23 17:10:11 +0000", "atm": "ATM : 5792055772", "amount": 400, "location"
: {"lat": "53.8357238", "lon": "-1.7965487"}, "transaction_id": "a2b8ab0a-ef42-11e8-b79a-0242c0a80003"}
% Reached end of topic atm_txns_gess [0] at offset 27199
{"account_id": "a536", "timestamp": "2018-11-23 17:10:11 +0000", "atm": "RBS", "amount": 20, "location": {"lat": "53.
7988921", "lon": "-1.547186"}, "transaction_id": "a32b8472-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a437", "timestamp": "2018-11-23 17:10:12 +0000", "atm": "Yorkshire Bank", "amount": 20, "location": {
"lat": "53.8203297", "lon": "-1.5765871"}, "transaction_id": "a39e48a4-ef42-11e8-b79a-0242c0a80003"}
% Reached end of topic atm_txns_gess [0] at offset 27201
{"account_id": "a437", "timestamp": "2018-11-23 17:01:18 +0000", "atm": "ATM : 1414423006", "amount": 400, "location"
: {"lat": "53.710754", "lon": "-2.101069"}, "transaction_id": "xxxa39e48a4-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a884", "timestamp": "2018-11-23 17:10:13 +0000", "atm": "NatWest", "amount": 50, "location": {"lat":
"53.6554012", "lon": "-1.8164935"}, "transaction_id": "a41186e8-ef42-11e8-b79a-0242c0a80003"}
{"account_id": "a823", "timestamp": "2018-11-23 17:10:14 +0000", "atm": "ATM : 1704973341", "amount": 200, "location"
: {"lat": "53.6103183", "lon": "-1.7028805"}, "transaction_id": "a47f3b84-ef42-11e8-b79a-0242c0a80003"}
% Reached end of topic atm_txns_gess [0] at offset 27204

KSQL : Stream Processing with SQL

@rmoff

```
SELECT TXN_ID, ATM,
       CUSTOMER_NAME,
       CUSTOMER_PHONE
FROM   ATM_POSSIBLE_FRAUD;
```

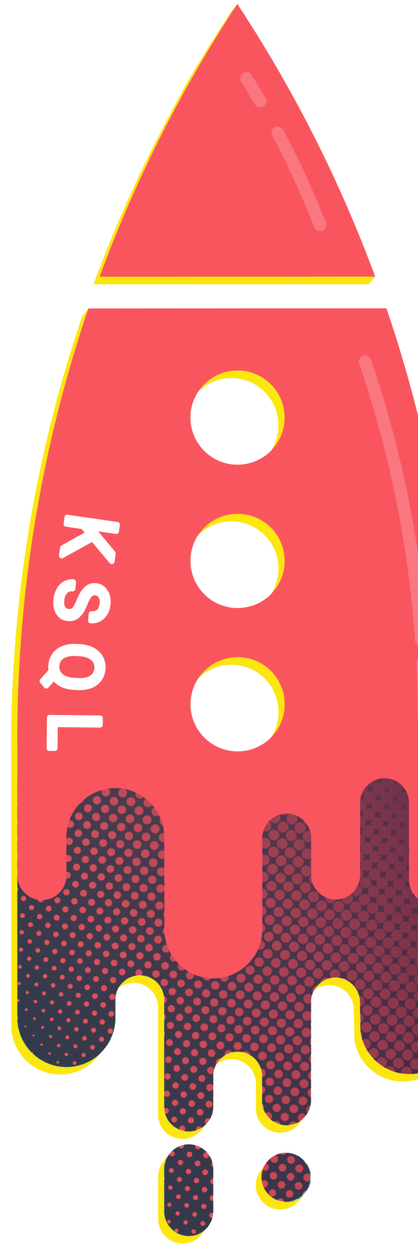
	ACCOUNT_ID	CUSTOMER_NAME	CUSTOMER_PHONE	TX1_ATM	TX2_ATM	MINUTES_DIFFERENC
13:55:37.000	a275	Willa Myderscough	+44 872 347 2824	NatWest	NatWest	0.167
13:54:35.000	a825	Matias Ciubutaro	+44 564 988 3000	ATM : 2149405513	Yorkshire Bank	0.15
13:53:48.000	a694	Garrik Murrell	+44 984 850 3016	Sainsbury	RBS	1.567
13:53:43.000	a616	Emile Henzer	+44 777 131 5530	Alliance & Leicester	ATM : 2377793700	0.617
13:53:41.000	a820	Elihu Beadon	+44 459 876 4518	ATM : 3612929539	ATM : 5156248736	1.417
13:53:36.000	a370	Chester Kinze	+44 459 263 5819	ATM : 2149405513	ATM : 1323671073	0.717
13:53:34.000	a889	Roarke Mordan	+44 459 922 2787	Yorkshire Bank	ATM : 3612929539	0.45
13:53:26.000	a244	Cathryn Burress	+44 434 738 8955	Yorkshire Bank	RBS	2.5
13:53:24.000	a367	Kiah Brockway	+44 236 318 0733	ATM : 4275471693	Yorkshire Bank	2.333
13:53:16.000	a382	Gib Embling	+44 203 139 3072	Cooperative	Barclays Bank PLC	2.3
13:53:14.000	a235	Teresina Pilch	+44 755 186 3106	ATM : 1414423006	TSB	2.367
13:53:13.000	a936	Courtney Trye	+44 874 560 4644	NatWest	ATM : 638692233	2
13:53:03.000	a640	Roderic Tolumello	+44 847 616 4985	ATM : 3634830654	Santander	2.5
13:52:50.000	a945	Steve Shiliton	+44 594 936 8109	HSBC	Co-operative Bank	2.517
13:52:45.000	a253	Eugenie Benoit	+44 960 103 3227	Yorkshire Bank	Co-operative Bank	0.05
13:52:44.000	a288	Sheilah Langthorne	+44 763 737 7574	ATM : 3634830654	ATM : 4051026380	1.7



1. Spot fraud in stream of transactions
2. Enrich transaction events with customer data

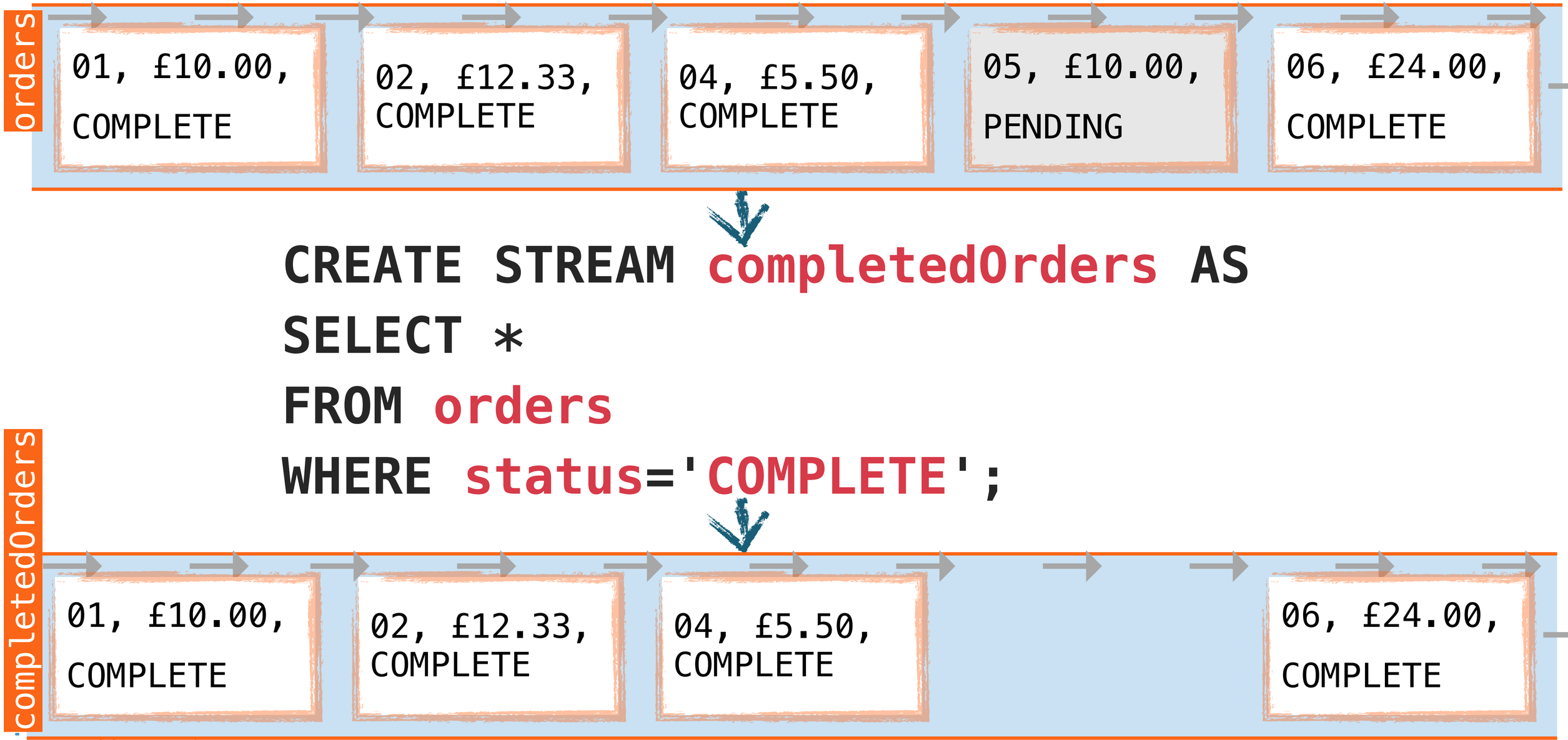
@rmoff

KSQL
is the
Streaming
SQL Engine
for
Apache Kafka



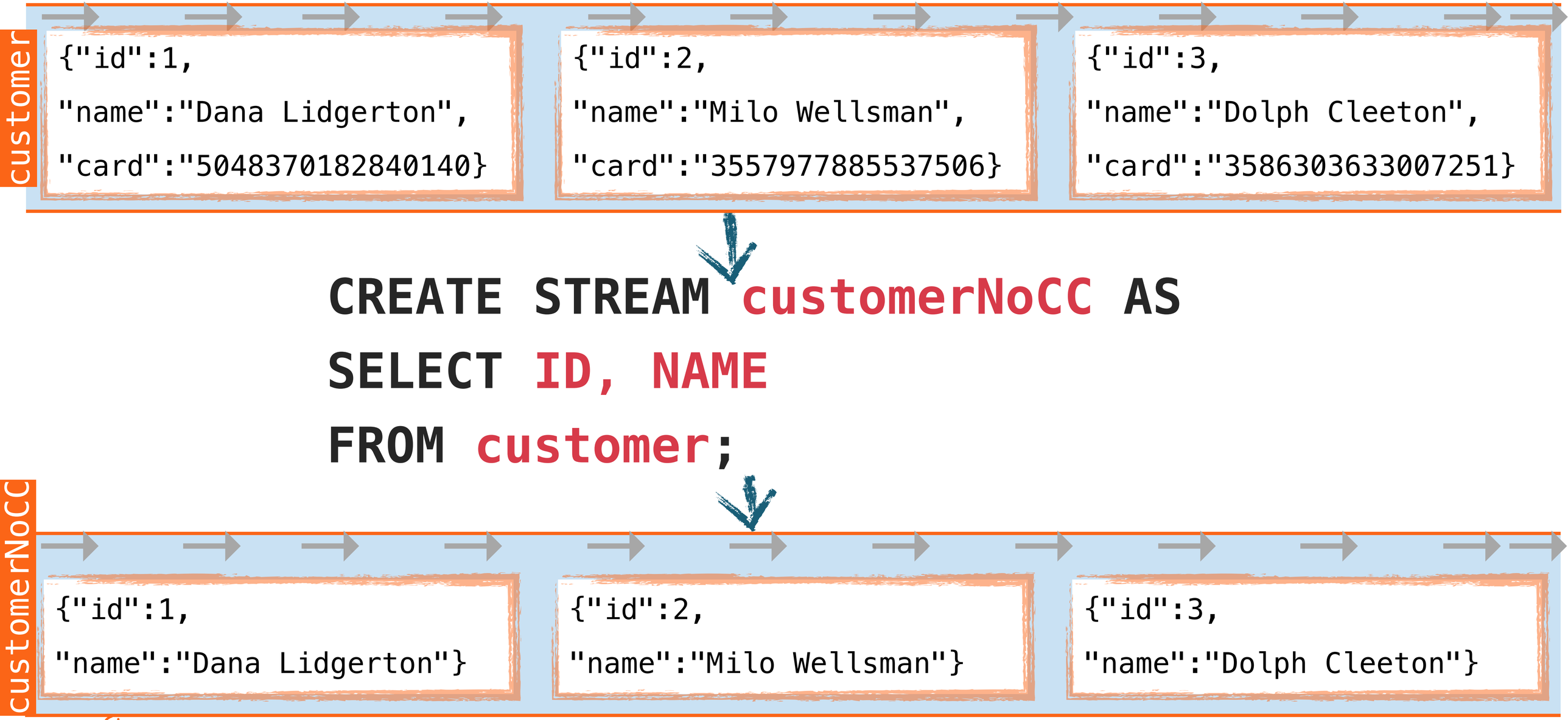
confluent

Filter messages with KSQL




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

Drop columns with KSQL



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;
```

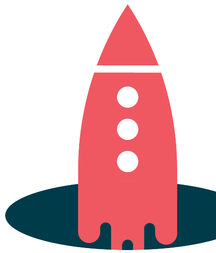
customersByCountry

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

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

KSQL for Data Transformation

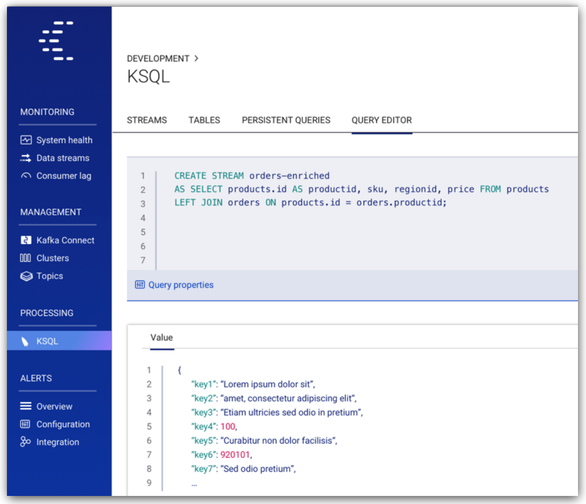
```
CREATE STREAM pageviews
  WITH (PARTITIONS=4,
        VALUE_FORMAT='AVRO') AS
  SELECT * FROM pageviews_json;
```



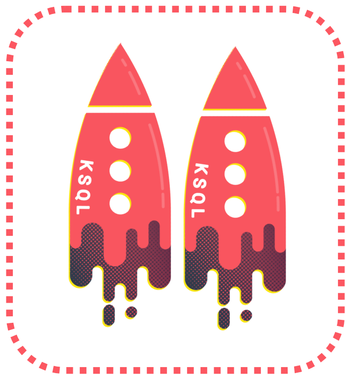
KSQL in Development and Production

Interactive KSQL
for development and testing

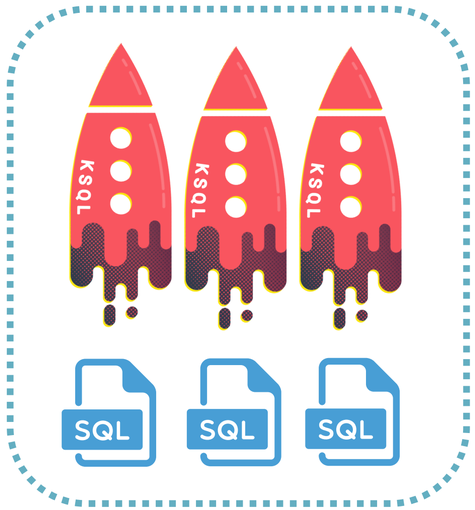
Headless KSQL
for Production



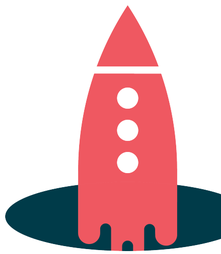
REST



Desired KSQL queries have been identified

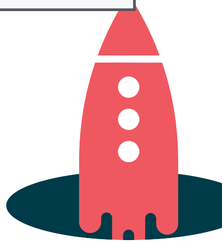


"Hmm, let me try out this idea..."

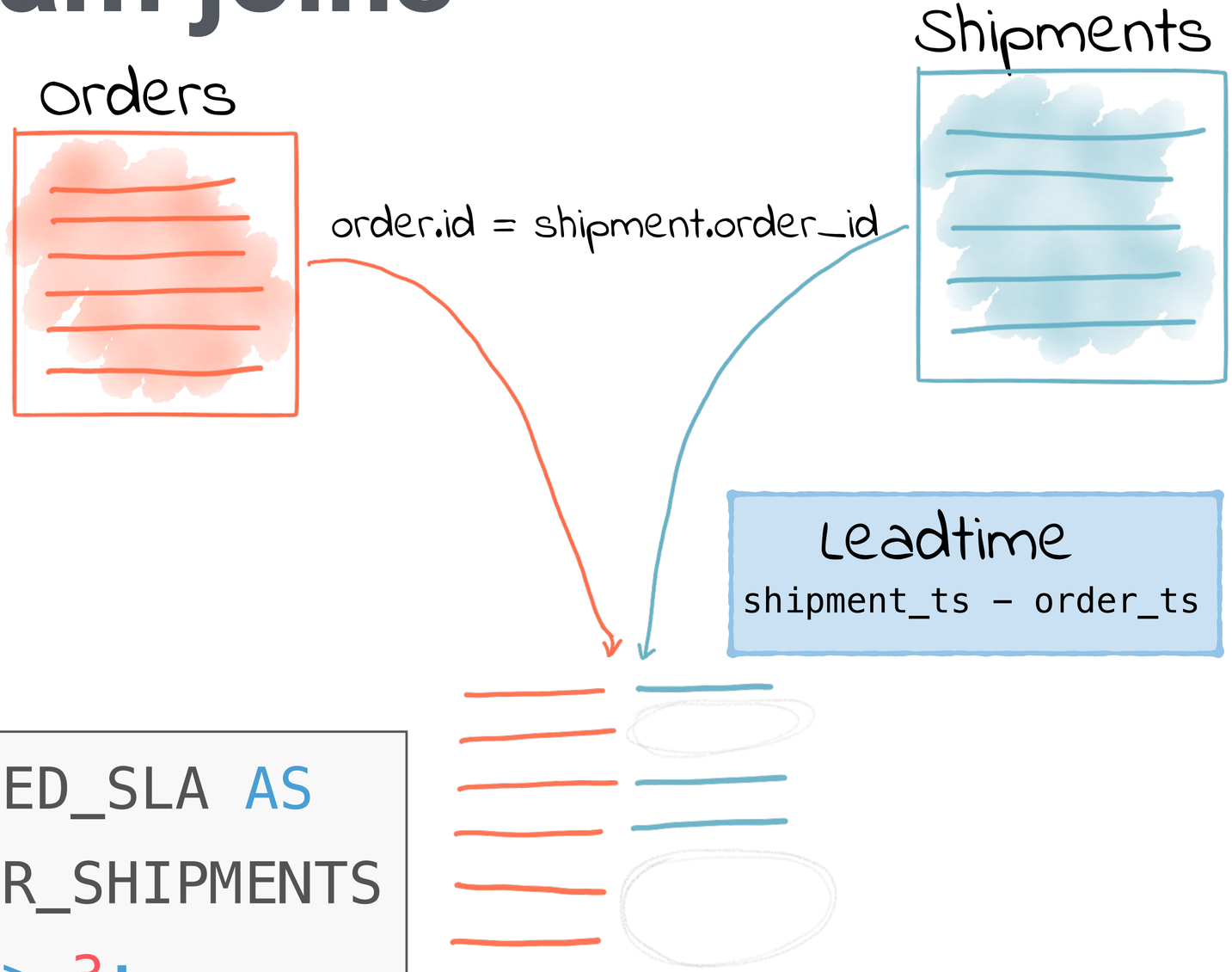


KSQL supports JOINs

```
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';
```

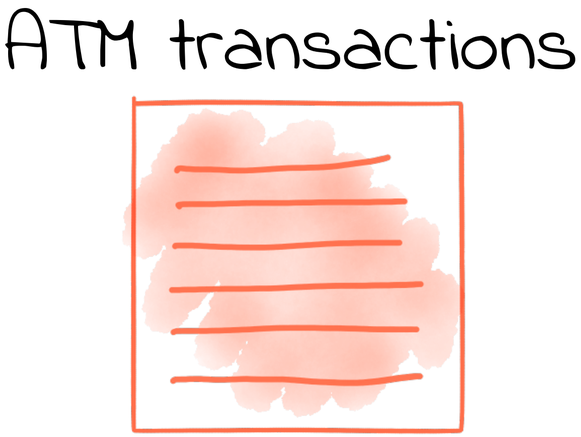


Stream Stream joins

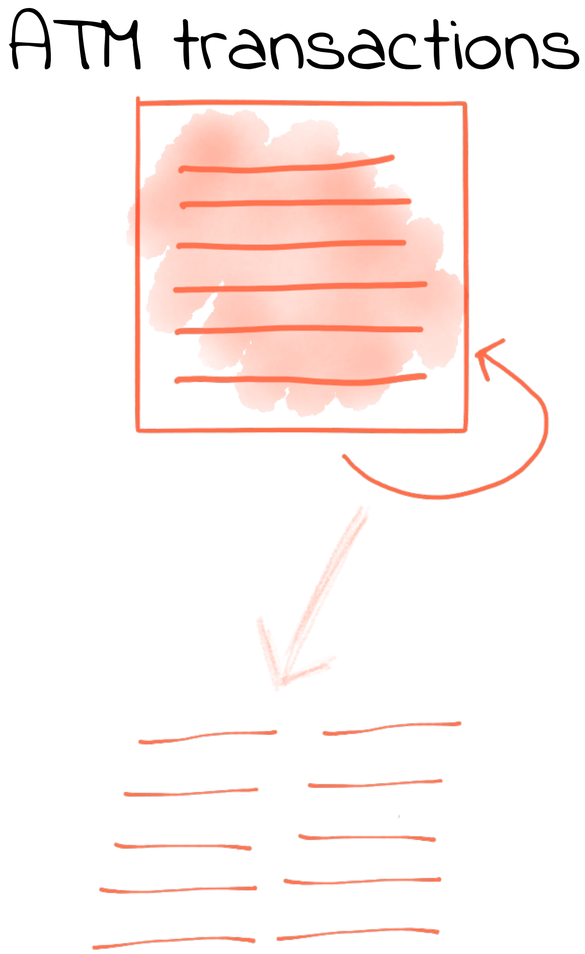



```
CREATE STREAM MISSED_SLA AS
SELECT * FROM ORDER_SHIPMENTS
WHERE LEADTIME_HR > 3;
```

Stream Stream joins



Stream Stream joins

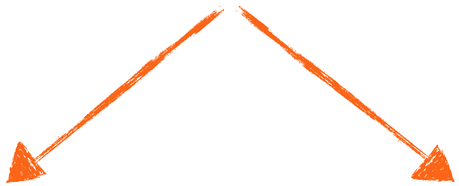


Demo!

Self-Join (Cartesian product)

T

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds



T1

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

T2

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

Self-Join (Cartesian product)

T1

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

T2

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

```
                ATM_TXNS T1
INNER JOIN ATM_TXNS T2

ON T1.ACCOUNT_ID = T2.ACCOUNT_ID
```

Self-Join (Cartesian product)

T1

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

T2

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

```
FROM          ATM_TXNS T1
INNER JOIN ATM_TXNS T2
                WITHIN 10 MINUTES
ON T1.ACCOUNT_ID = T2.ACCOUNT_ID
```

Self-Join

T1 Txn ID	T2 Txn ID	T1 Time	T2 Time	T1 ATM	T2 ATM
xxx116d91d6-ef17	116d91d6-ef17	11:56:58	11:58:19	Midland	Halifax
116d91d6-ef17	xxx116d91d6-ef17	11:58:19	11:56:58	Halifax	Midland
xxx116d91d6-ef17	xxx116d91d6-ef17	11:56:58	11:56:58	Midland	Midland
116d91d6-ef17	116d91d6-ef17	11:58:19	11:58:19	Halifax	Halifax

Self-Join

T1 Txn ID	T2 Txn ID	T1 Time	T2 Time	T1 ATM	T2 ATM
xxx116d91d6-ef17	116d91d6-ef17	11:56:58	11:58:19	Midland	Halifax
116d91d6-ef17	xxx116d91d6-ef17	11:58:19	11:56:58	Halifax	Midland
xxx116d91d6-ef17	xxx116d91d6-ef17	11:56:58	11:56:58	Midland	Midland
116d91d6-ef17	116d91d6-ef17	11:58:19	11:58:19	Halifax	Halifax

Self join on same txn IDs

Exclude joins on the same txn

T1 Txn ID	T2 Txn ID	T1 Time	T2 Time	T1 ATM	T2 ATM
xxx116d91d6-ef17	116d91d6-ef17	11:56:58	11:58:19	Midland	Halifax
116d91d6-ef17	xxx116d91d6-ef17	11:58:19	11:56:58	Halifax	Midland

```
WHERE   T1.TRANSACTION_ID !=
        T2.TRANSACTION_ID
```

Exclude joins on the same txn

T1 Txn ID	T2 Txn ID	T1 Time	T2 Time	T1 ATM	T2 ATM
xxx116d91d6-ef17	116d91d6-ef17	11:56:58	11:58:19	Midland	Halifax
116d91d6-ef17	xxx116d91d6-ef17	11:58:19	11:56:58	Halifax	Midland

Duplicate results (A:B / B:A)

Join Windows

T1

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

T2

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

```
                WITHIN 10 MINUTES
WHERE           T1.TRANSACTION_ID !=
                T2.TRANSACTION_ID
```

Join Windows

T1

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

T2

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

```
WITHIN 10 MINUTES
WHERE   T1.TRANSACTION_ID != T2.TRANSACTION_ID
```

Join Windows

T1

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

T2

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

```
WITHIN 10 MINUTES
WHERE   T1.TRANSACTION_ID !=
        T2.TRANSACTION_ID
```

Only join *forward*

T1

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

T2

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

```
    WITHIN (0 MINUTES, 10 MINUTES)
WHERE   T1.TRANSACTION_ID !=
        T2.TRANSACTION_ID
```

Only join *forward*

T1

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

T2

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

```
    WITHIN (0 MINUTES, 10 MINUTES)
WHERE   T1.TRANSACTION_ID !=
        T2.TRANSACTION_ID
```

Only join *forward*

T1 Txn ID	T2 Txn ID	T1 Time	T2 Time	T1 ATM	T2 ATM
xxx116d91d6-ef17	116d91d6-ef17	11:56:58	11:58:19	Midland	Halifax

```
WITHIN (0 MINUTES, 10 MINUTES)
```

Ignore events in the right-hand stream prior to those in the left

Only join *forward*

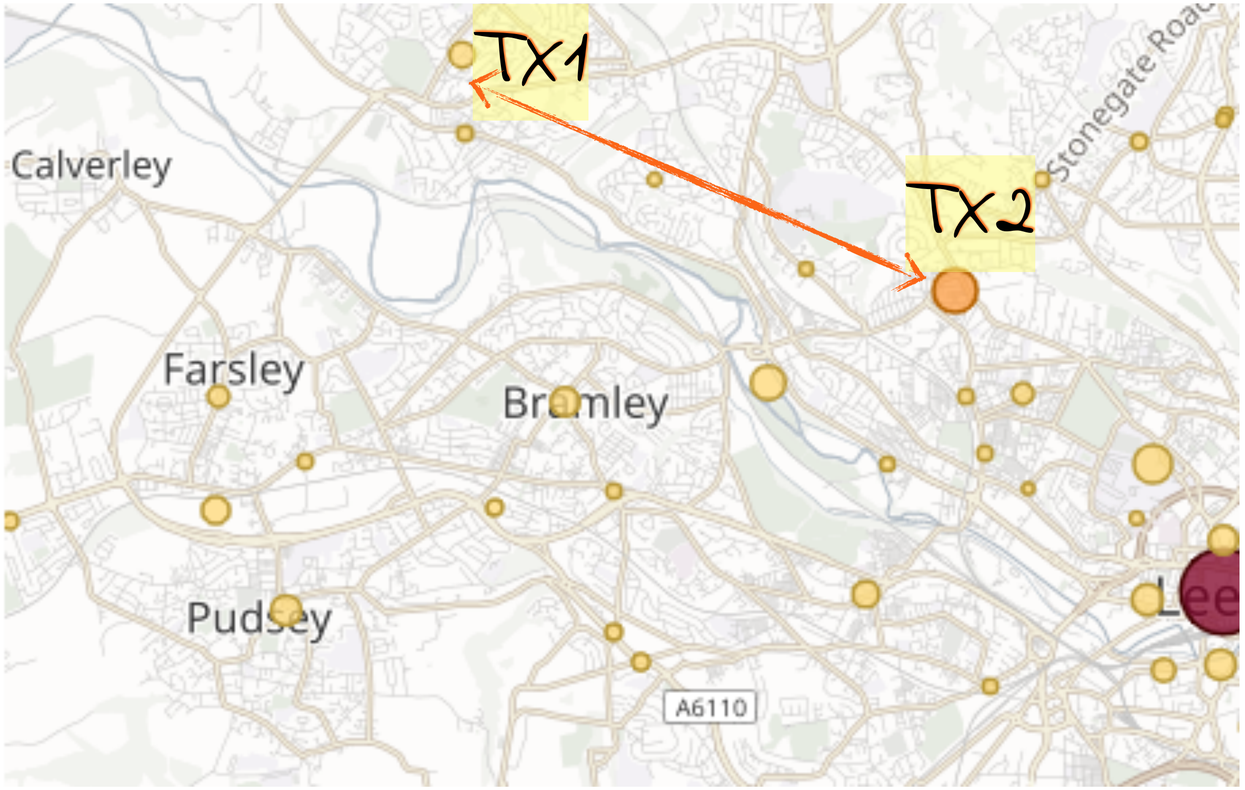
T1 Txn ID	T2 Txn ID	T1 Time	T2 Time	T1 ATM	T2 ATM
xxx116d91d6-ef17	116d91d6-ef17	11:56:58	11:58:19	Midland	Halifax

Legit

Dodgy!

Photo by Esteban Lopez on Unsplash

Calcuate distance between ATMs




```
GEO_DISTANCE(TX1.location->lat, TX1.location->lon,
             TX2.location->lat, TX2.location->lon,
             'KM')
```

Calculate time between transactions

@rmoff

```
TX2.ROWTIME - TX1.ROWTIME AS
 MILLISECONDS_DIFFERENCE


(TX2.ROWTIME - TX1.ROWTIME)
 / 1000 / 60 / 60 AS HOURS_DIFFERENCE
```

```
GEO_DISTANCE(…) / HOURS_DIFFERENCE
AS KMH_REQUIRED
```

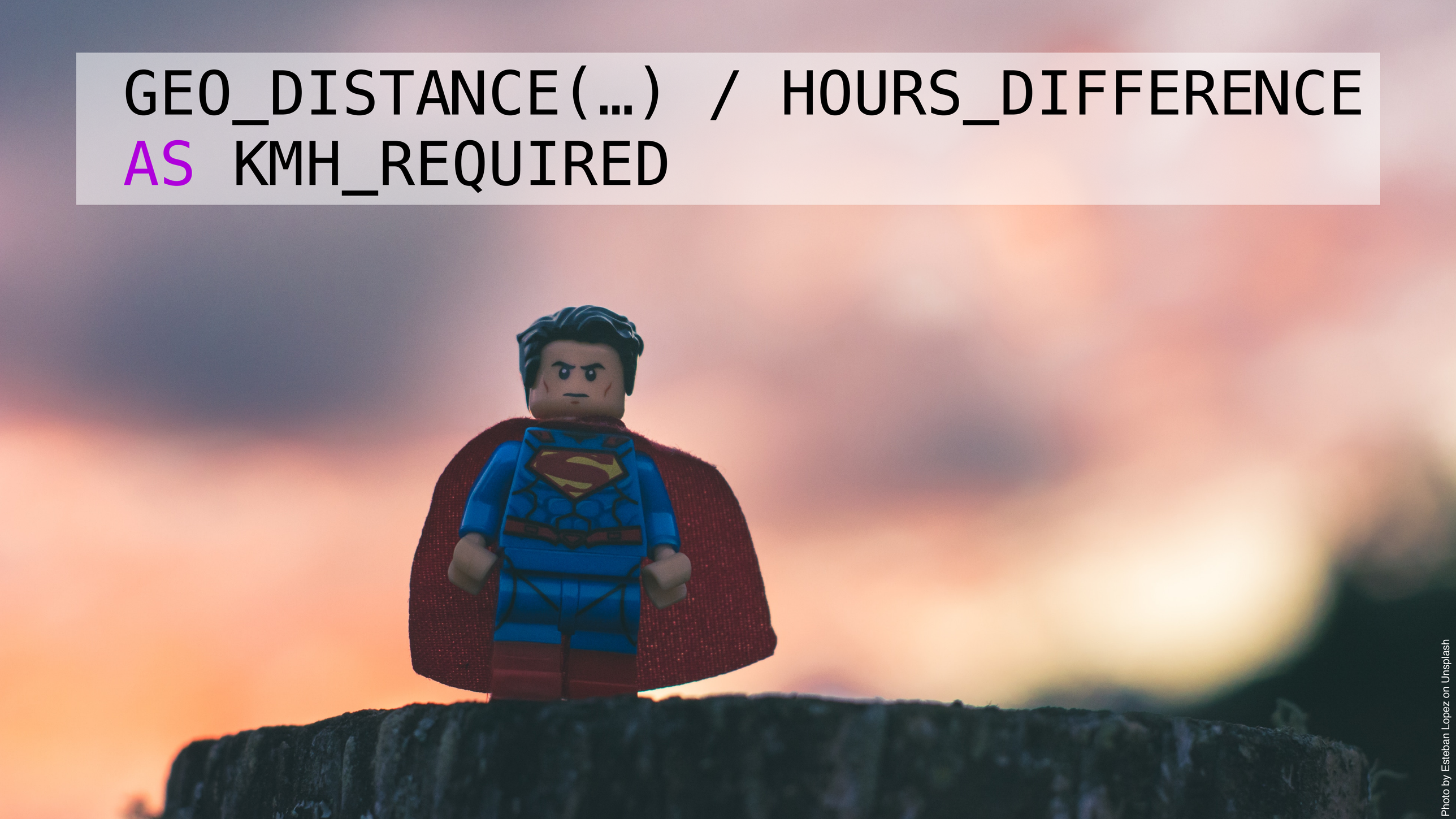


Photo by Esteban Lopez on Unsplash

So speaking of time…

```
ksql> PRINT 'atm_txns_gess' ;
Format:JSON
{
   "ROWTIME": 1544116309152,
   "ROWKEY": "null",
   "account_id": "a218",
   "timestamp": "2018-12-06 17:09:58 +0000",
   "atm": "HSBC",
…}
```

Kafka message timestamp

2018-12-06 17:11:49

Event time

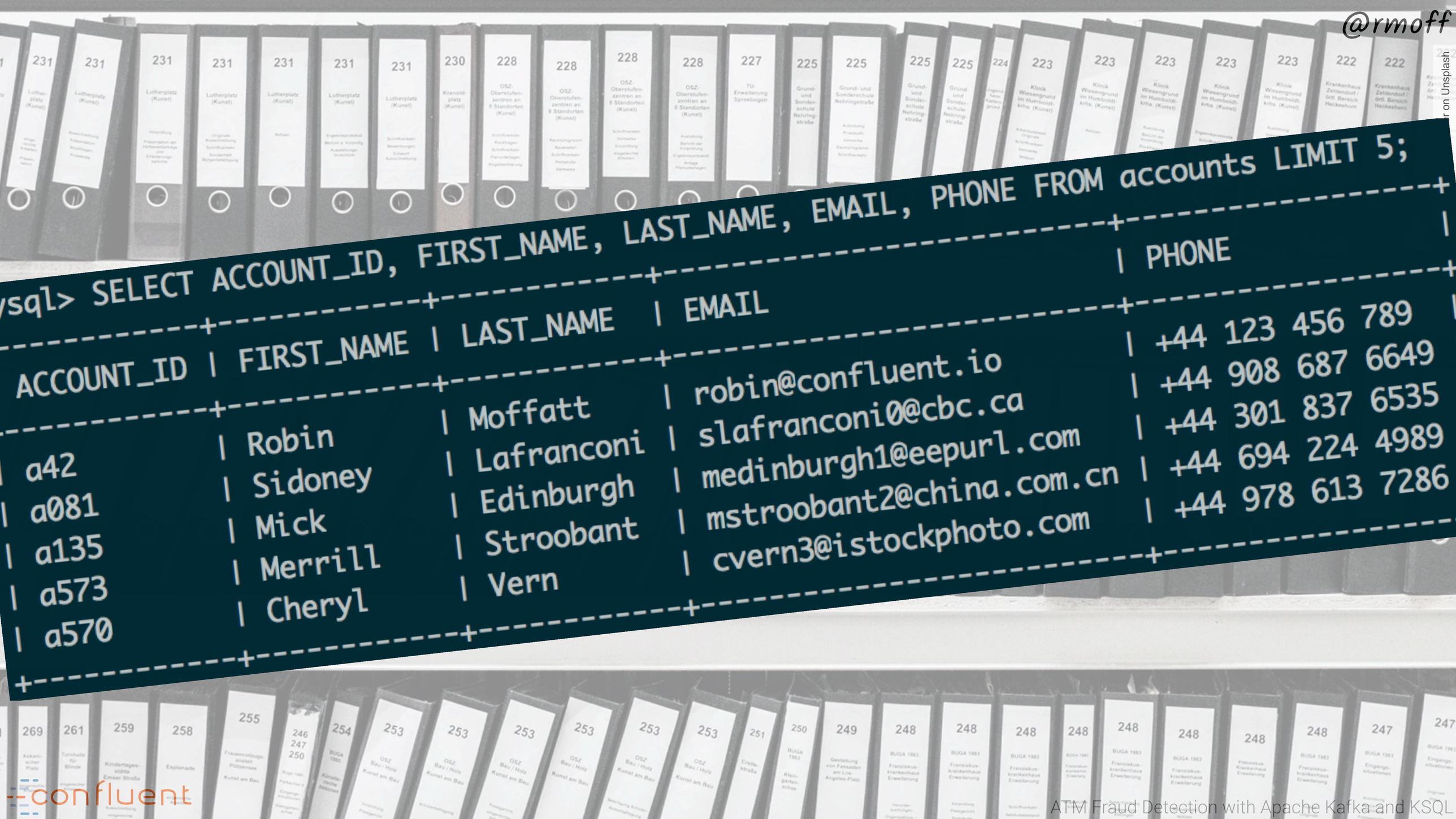
```
CREATE STREAM ATM_TXNS_GESS
(account_id VARCHAR, timestamp VARCHAR, …
WITH (KAFKA_TOPIC='atm_txns_gess',
TIMESTAMP='timestamp',
TIMESTAMP_FORMAT=
    'yyyy-MM-dd HH:mm:ss X');
```

```
ksql> PRINT 'atm_txns_gess' ;
Format:JSON
{
  "ROWTIME": 1544116309152,
  "ROWKEY": "null",
  "account_id": "a218",
  "timestamp": "2018-12-06 17:09:58 +0000",
  "atm": "USBC",
```

@rmoff

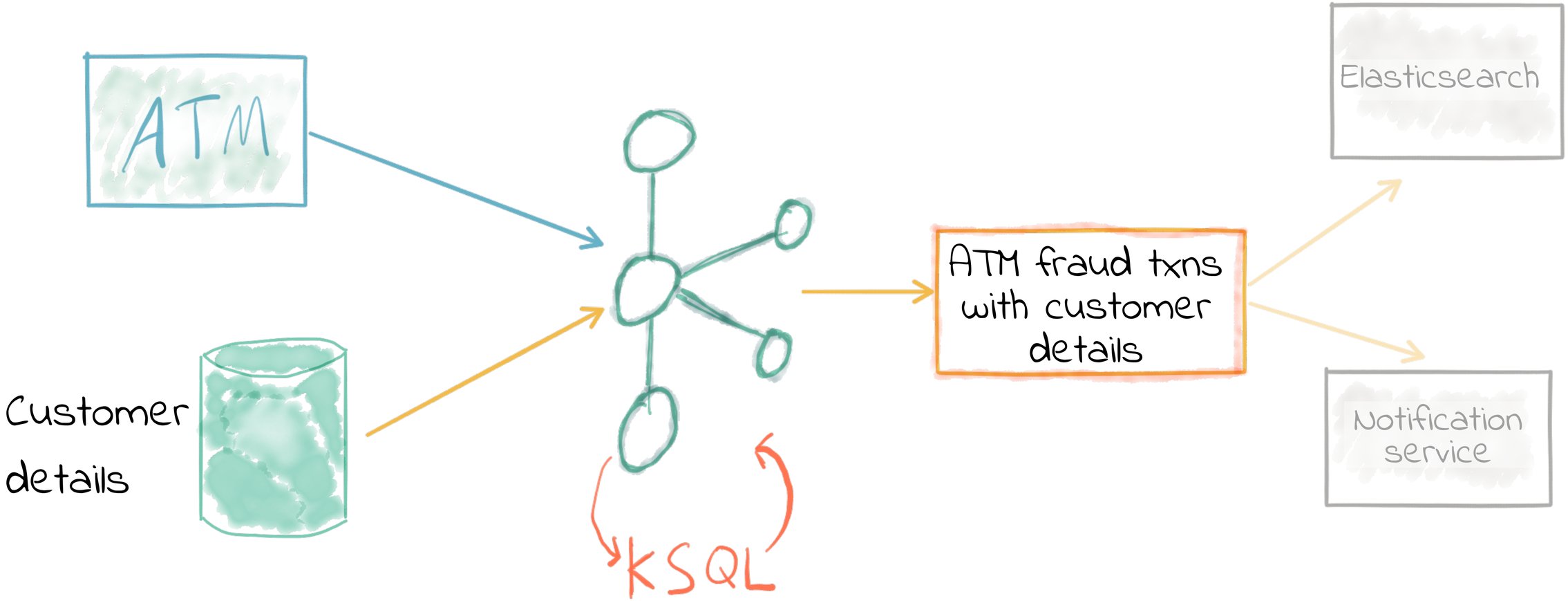
But what about the account holder?

@rmoff



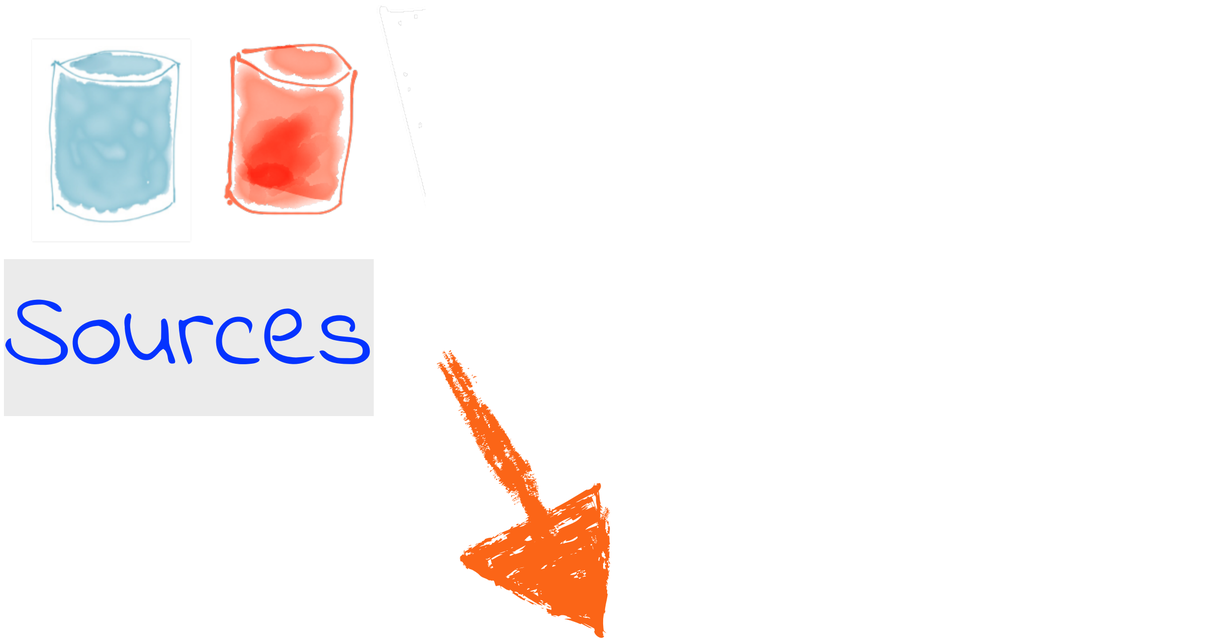
```
mysql> SELECT ACCOUNT_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE FROM accounts LIMIT 5;
+------------+------------+-----------+--------------------------+-----------------+
| ACCOUNT_ID | FIRST_NAME | LAST_NAME | EMAIL                    | PHONE           |
+------------+------------+-----------+--------------------------+-----------------+
| a42        | Robin      | Moffatt   | robin@confluent.io       | +44 123 456 789 |
| a081       | Sidoney    | Lafranconi | slafranconi0@cbc.ca     | +44 908 687 6649 |
| a135       | Mick       | Edinburgh | medinburgh1@eepurl.com   | +44 301 837 6535 |
| a573       | Merrill    | Stroobant | mstroobant2@china.com.cn | +44 694 224 4989 |
| a570       | Cheryl     | Vern      | cvern3@istockphoto.com   | +44 978 613 7286 |
+------------+------------+-----------+--------------------------+-----------------+
```

confluent



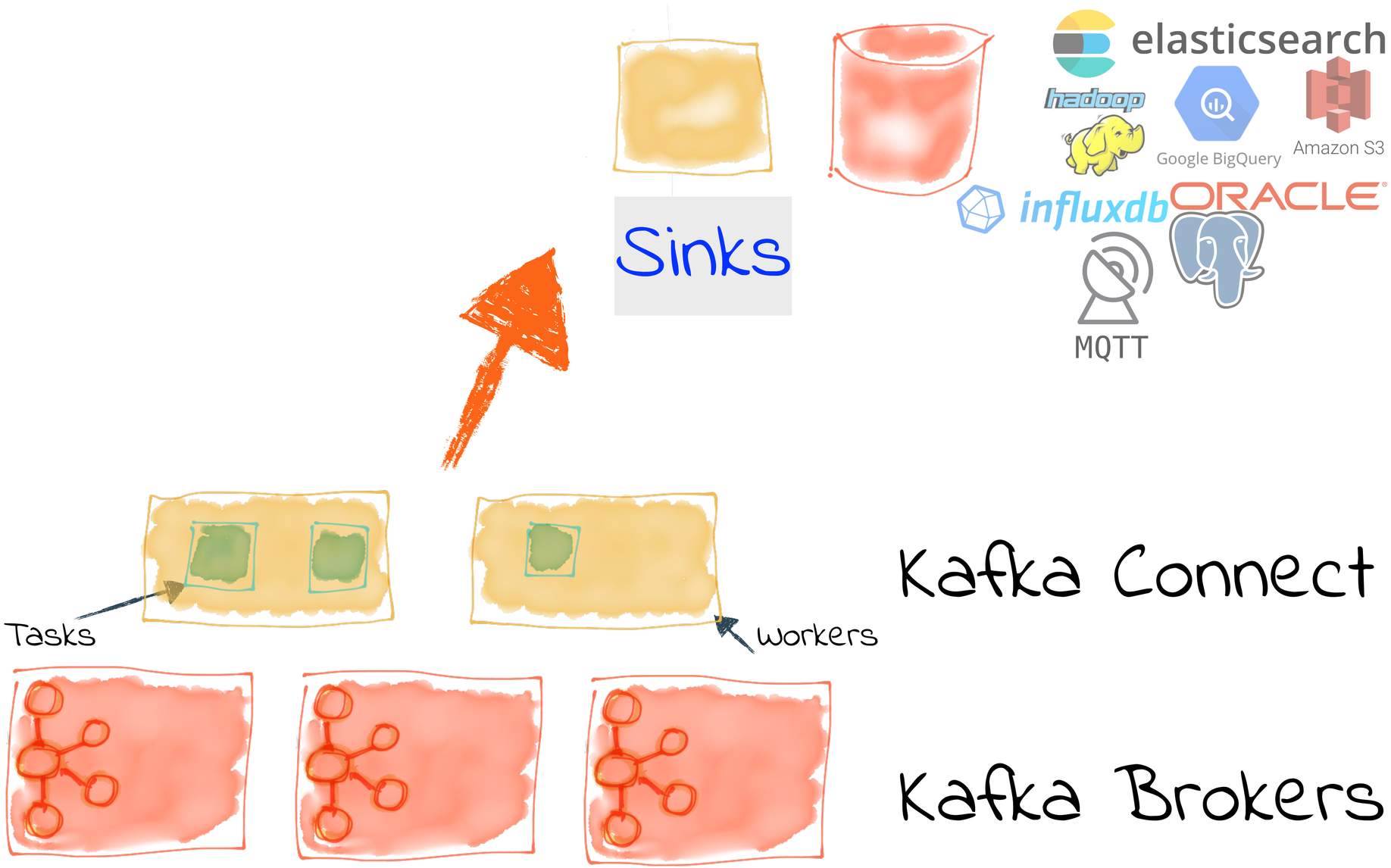
1. Enrich transaction events with customer data

Streaming Integration with Kafka Connect



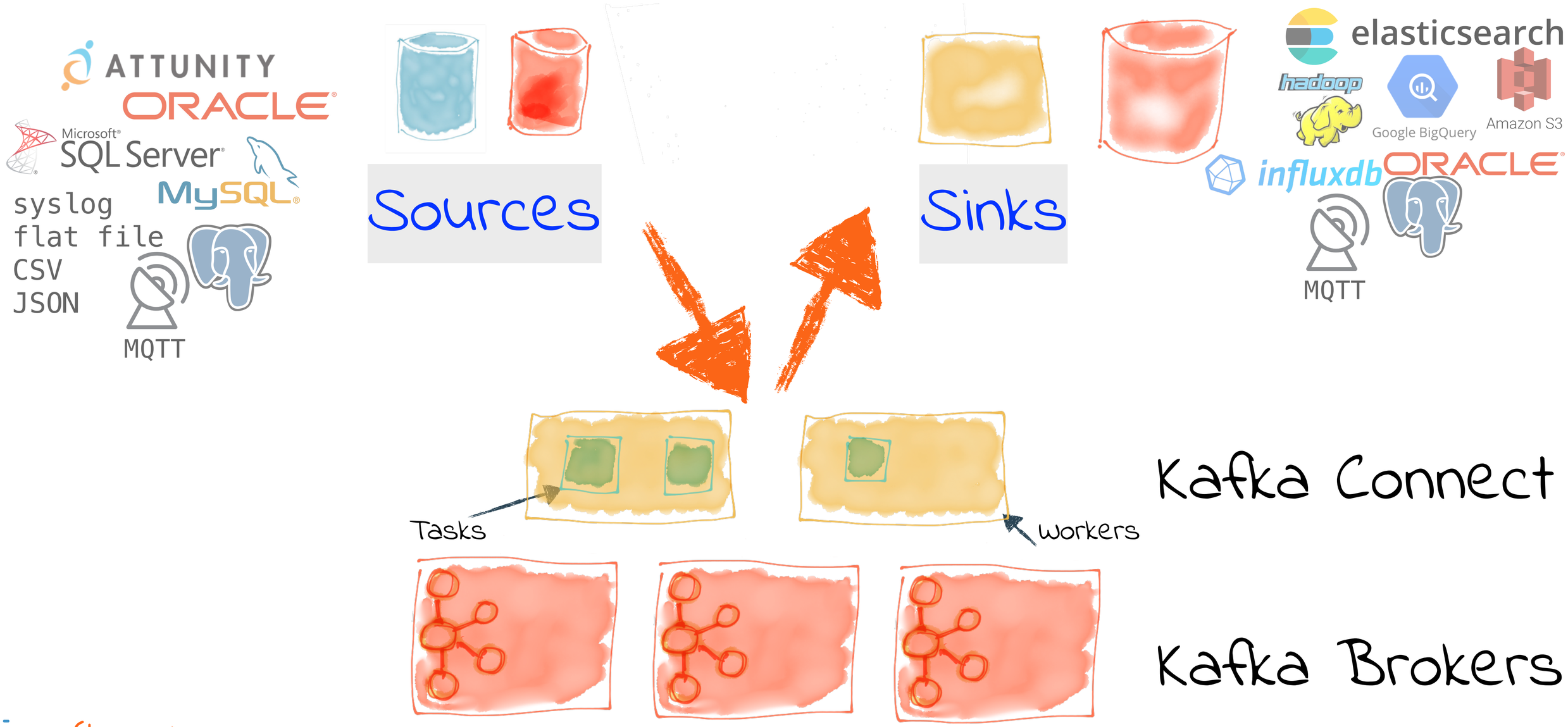
@rmoff

Streaming Integration with Kafka Connect



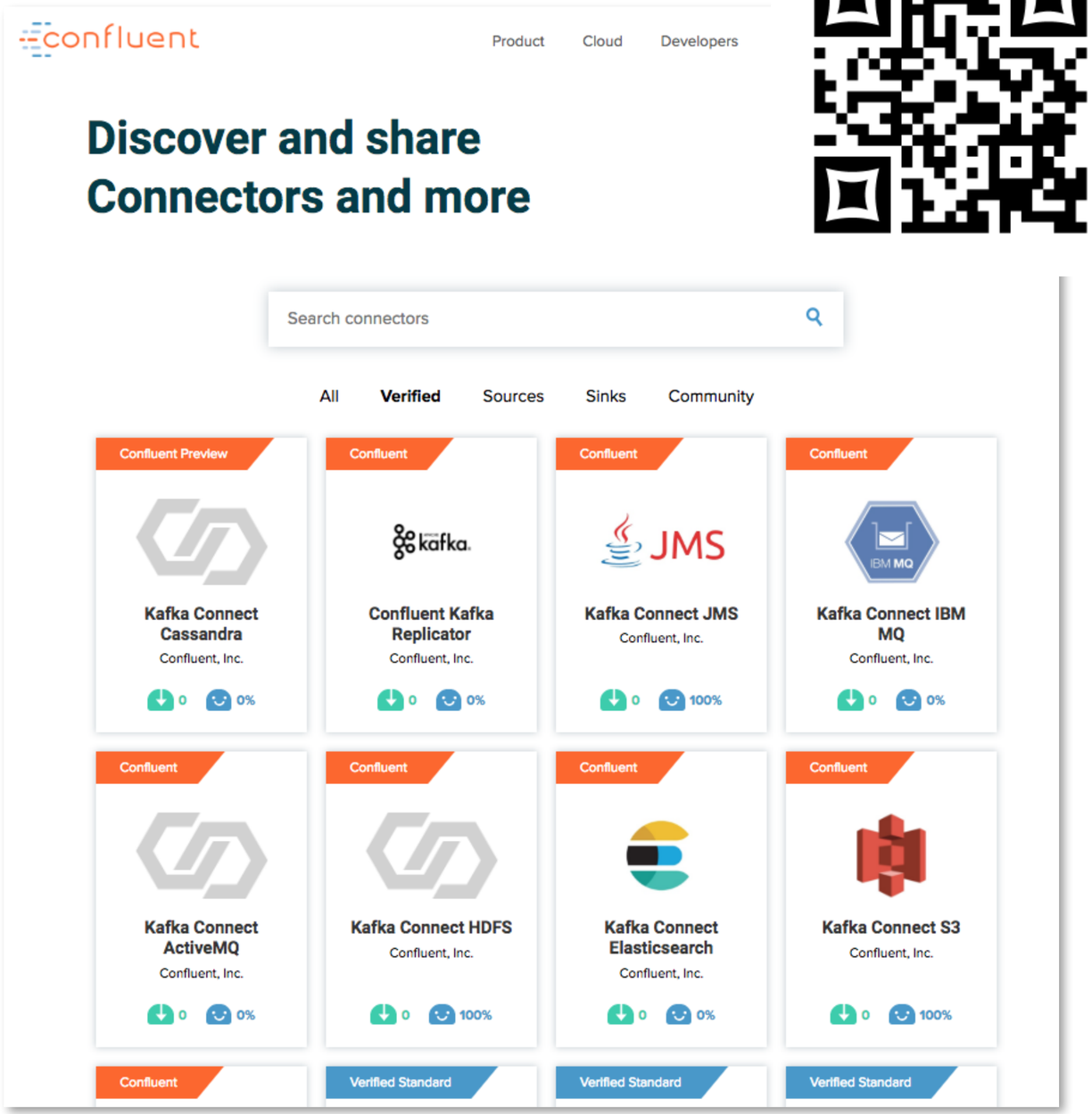
@rmoff

Streaming Integration with Kafka Connect



Confluent Hub

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



hub.confluent.io

The Table Stream Duality

Stream

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

Time

Table

Account ID	Balance
12345	€50

Account ID	Balance
12345	€75

Account ID	Balance
12345	€15

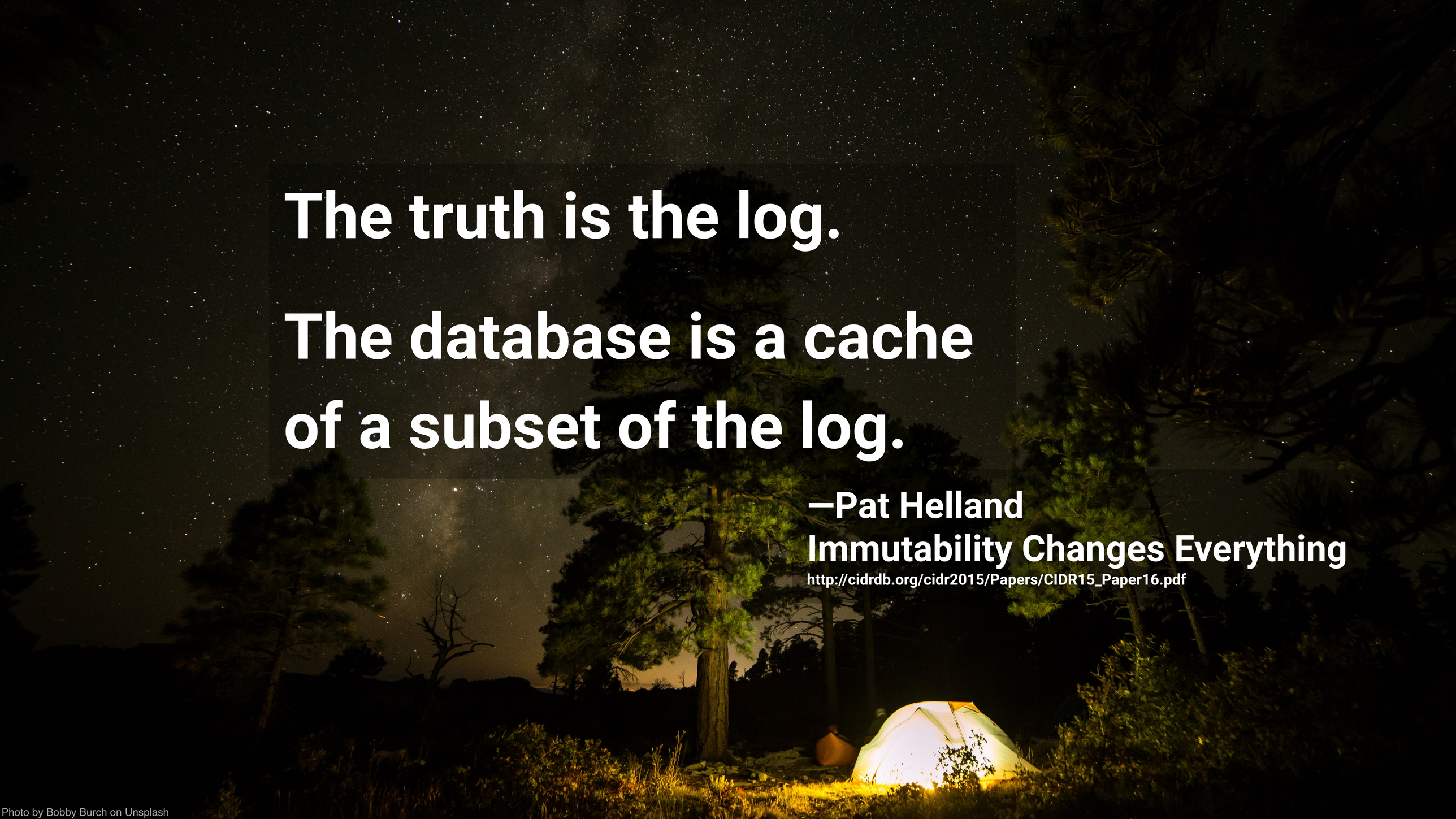
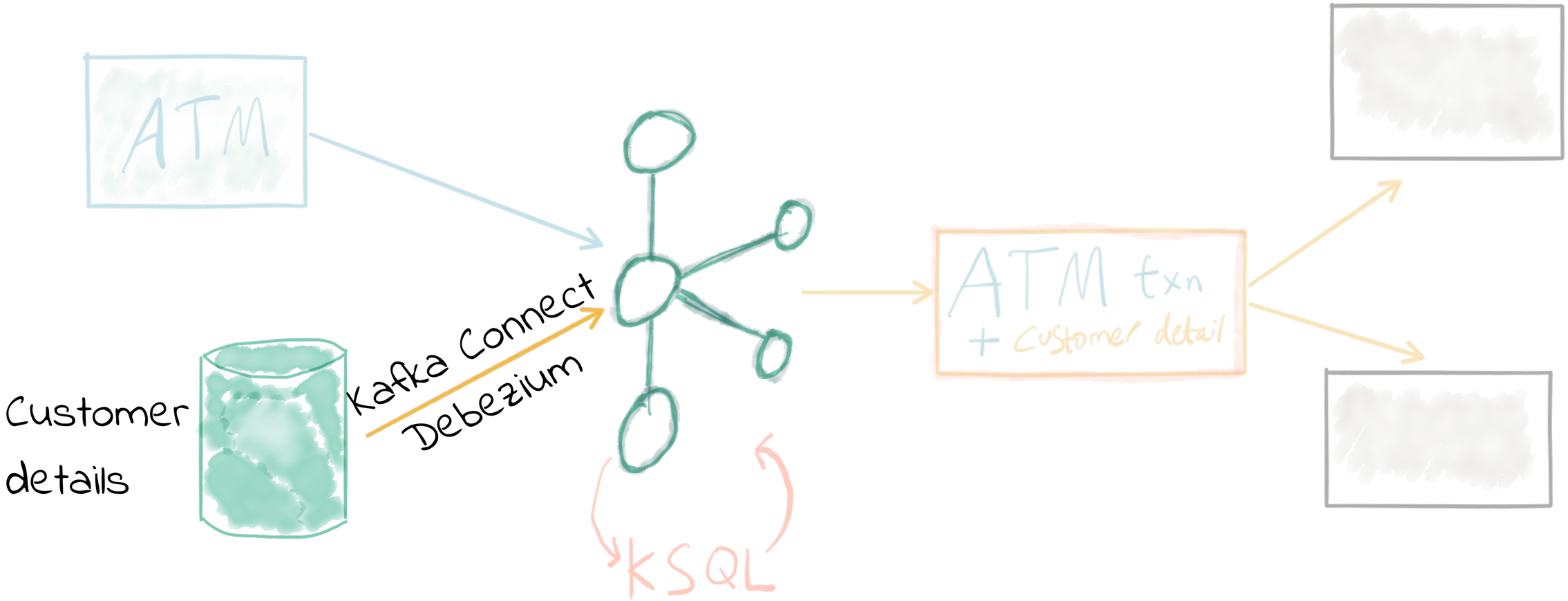


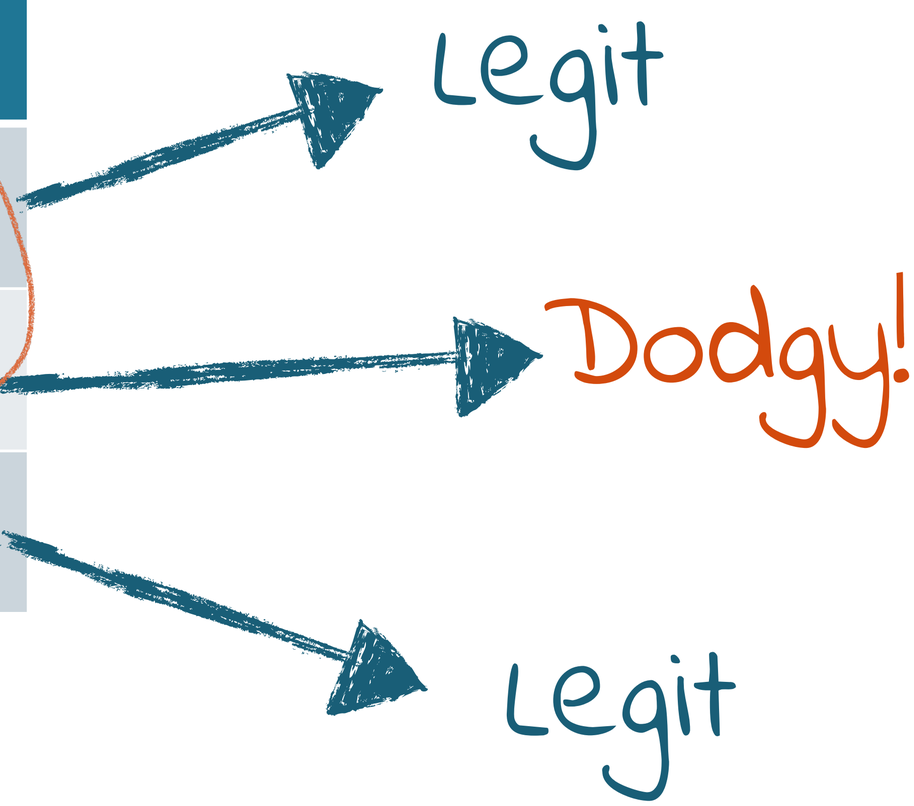
Photo by Bobby Burch on Unsplash

Demo Time!


ATM
Customer details
Kafka Connect
Debezium
KSQL
ATM txn
+ customer detail

Spot patterns within this stream

Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds

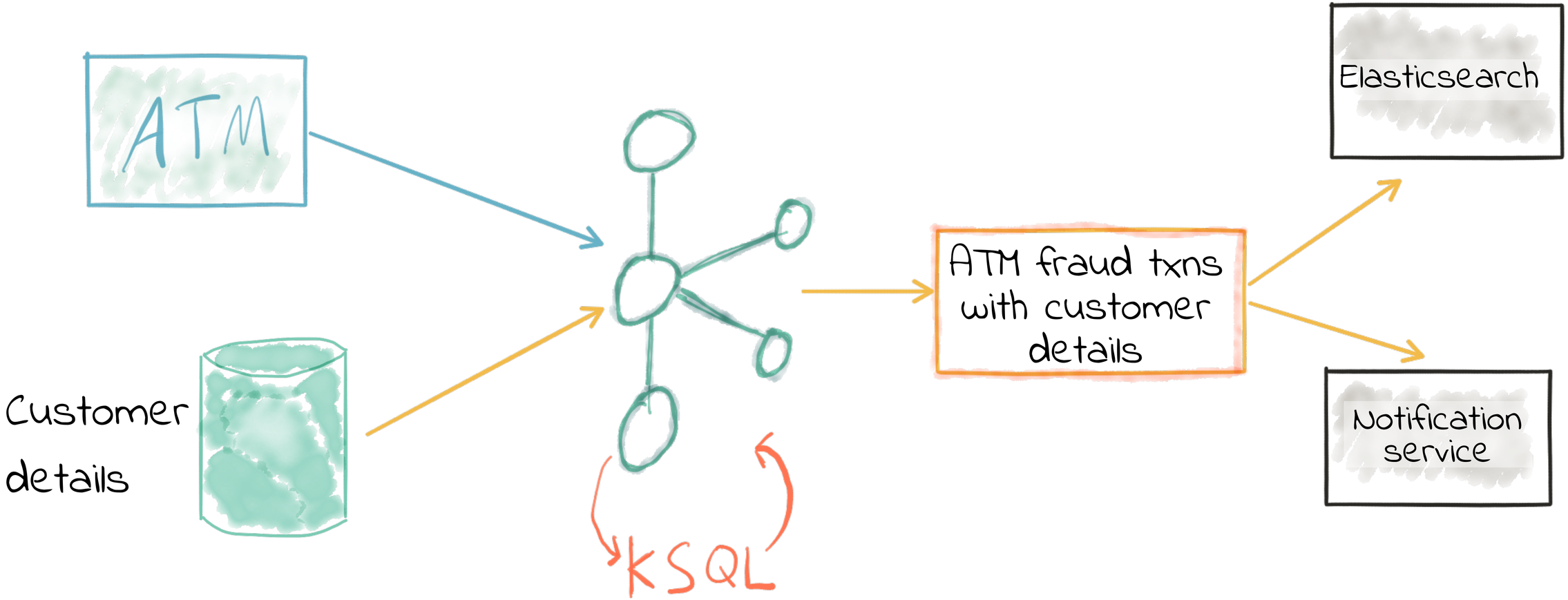


Suspect Transactions

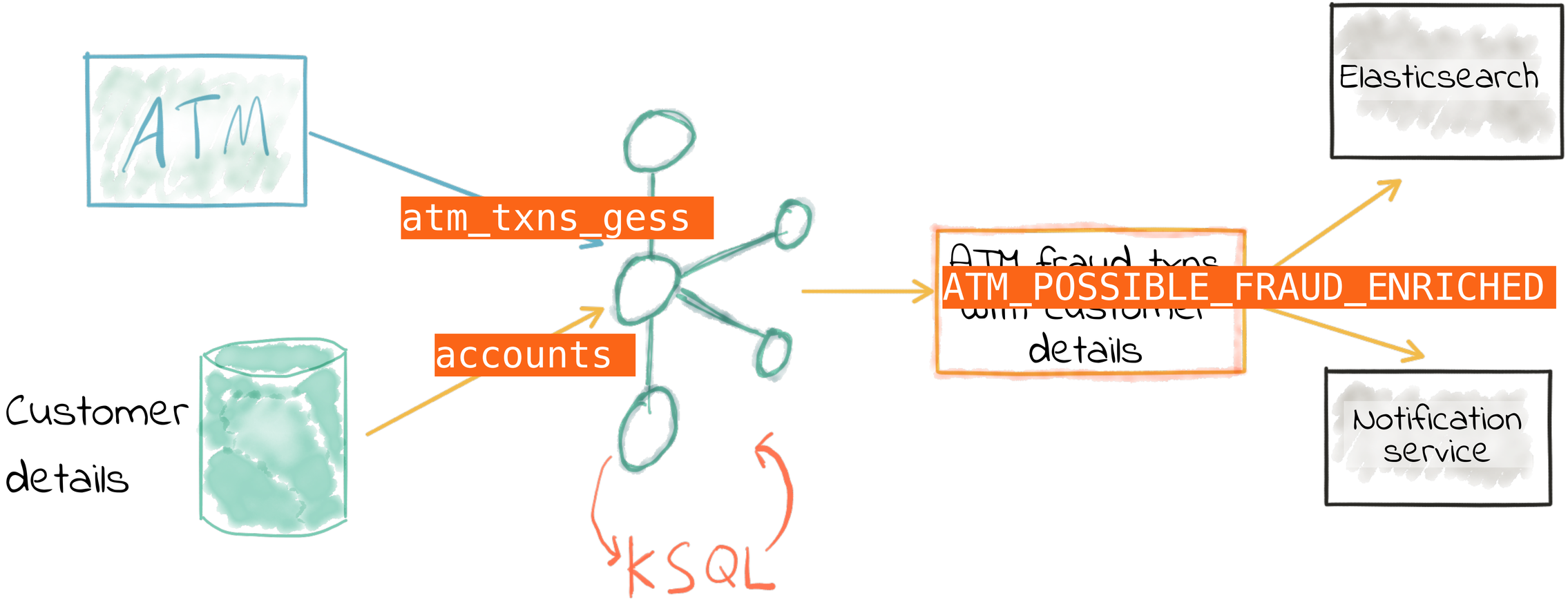
Ac. ID	T1 Time	ATM	T2 Time	ATM
A42	11:56:58	Midland	11:58:19	Halifax

Suspect Transactions

Name	Phone	Ac. ID	T1 Time	ATM	T2 Time	ATM
Robin M	1234 567	A42	11:56:58	Midland	11:58:19	Halifax

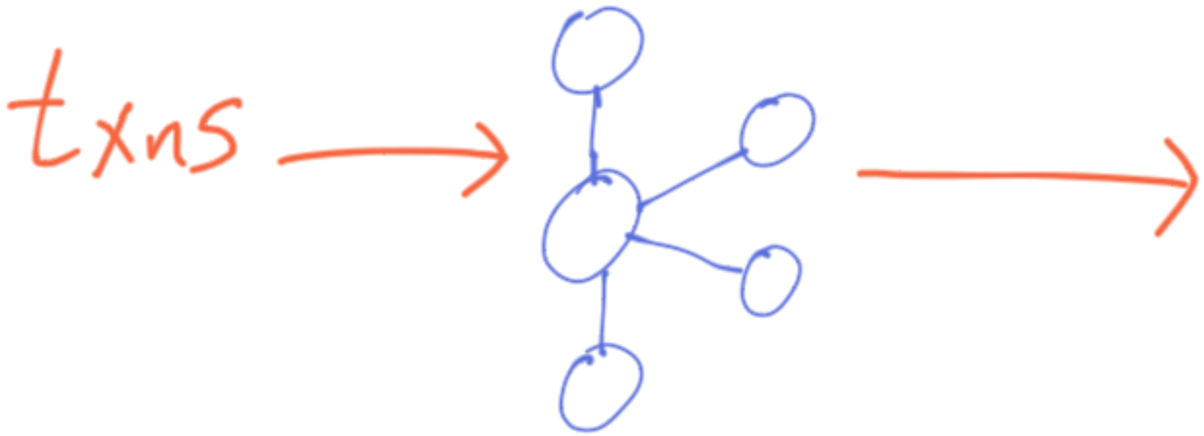


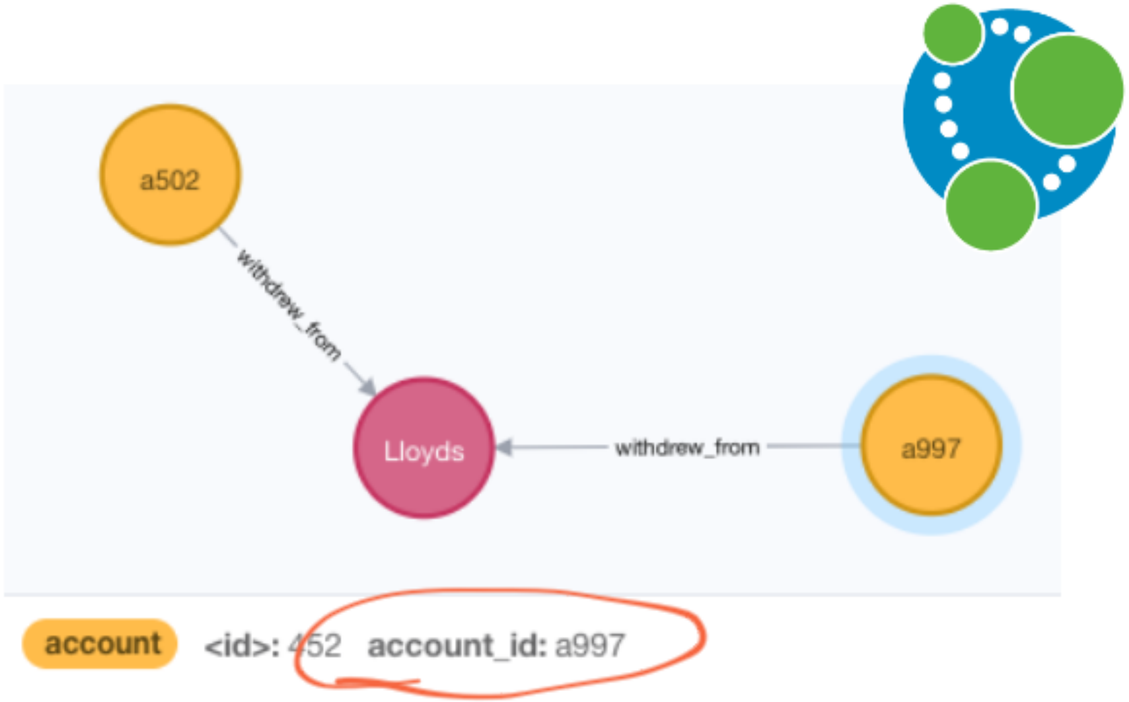
1. Spot fraud in stream of transactions
2. Enrich transaction events with customer data



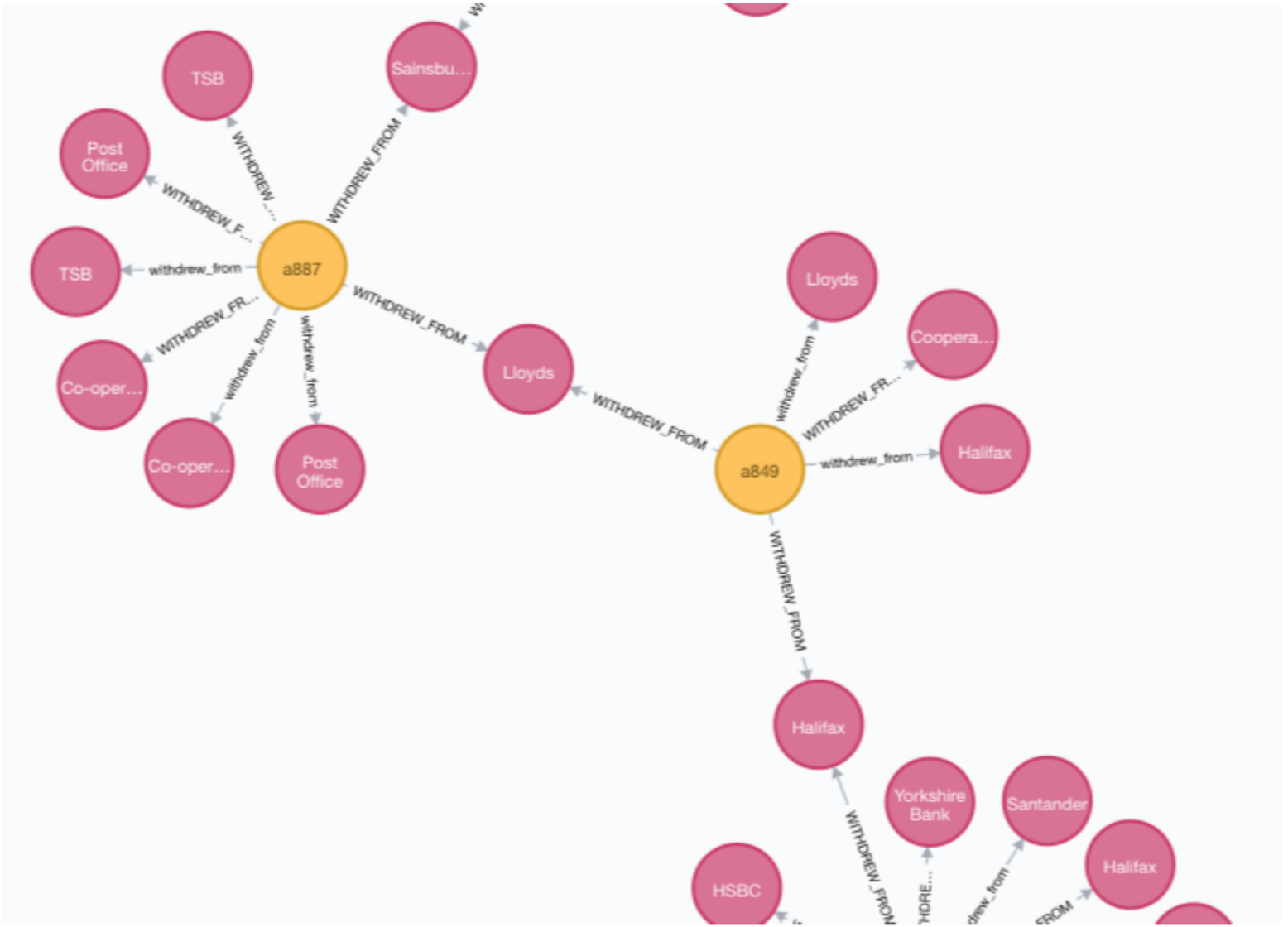
1. Spot fraud in stream of transactions
2. Enrich transaction events with customer data

Kafka Connect + Neo4j



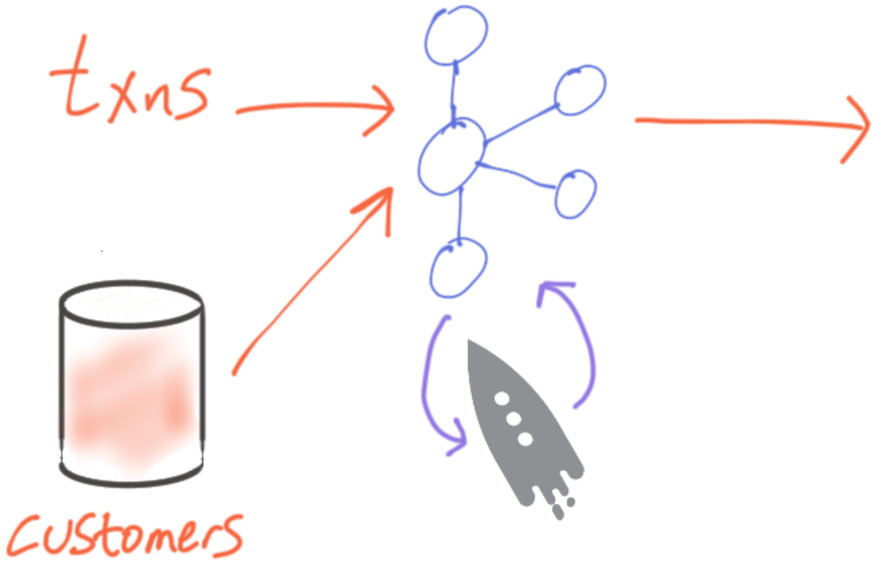


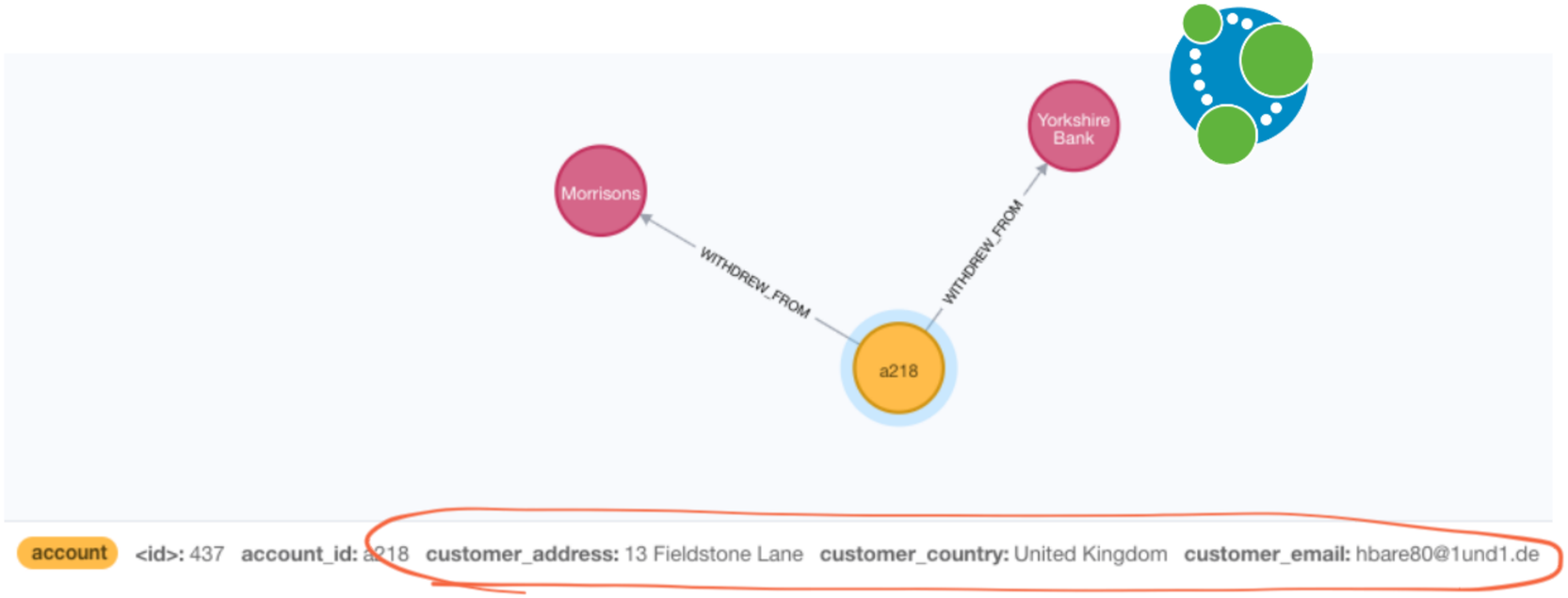
Kafka Connect + Neo4j



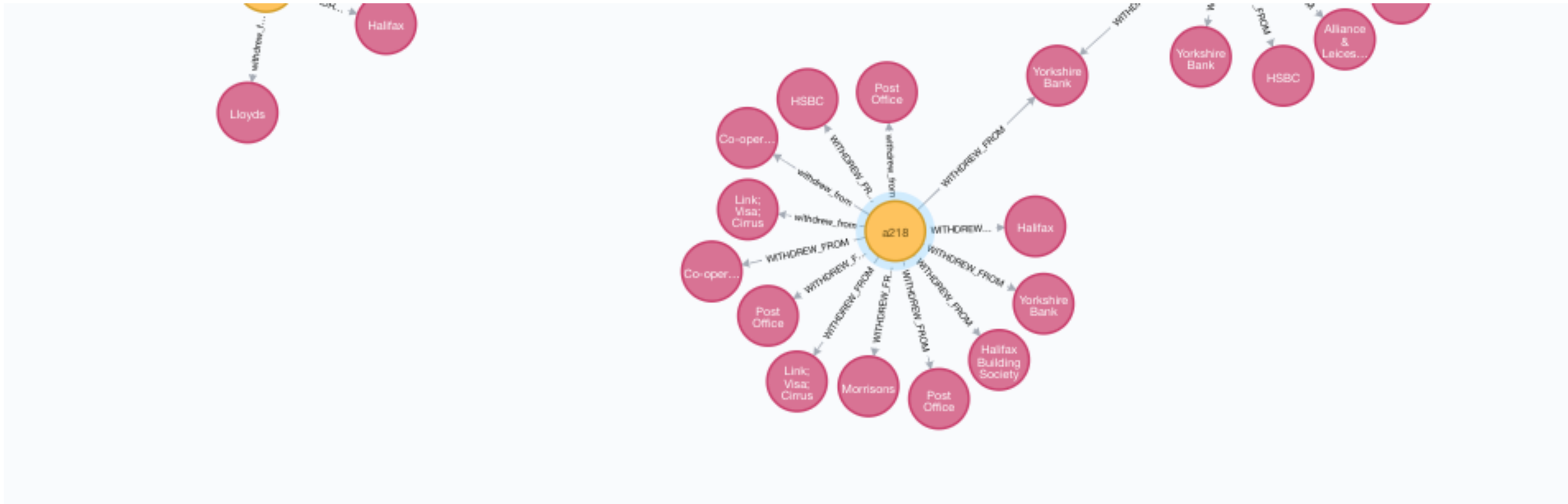
Kafka Connect + KSQL + Neo4j

@rmoff





Kafka Connect + KSQL + Neo4j

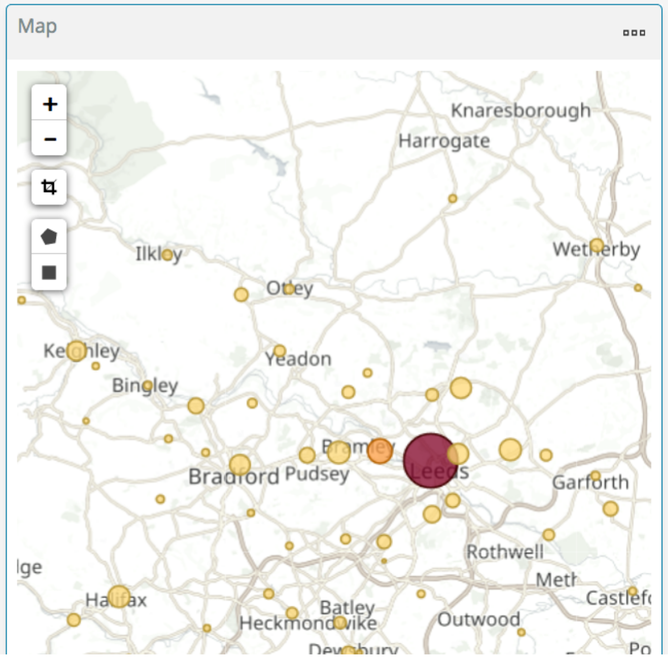


account <id>: 437 account_id: a218 customer_address: 13 Fieldstone Lane customer_country: United Kingdom customer_email: hbare80@1und1.de

Photo by Joshua Rodriguez on Unsplash

@rmoff

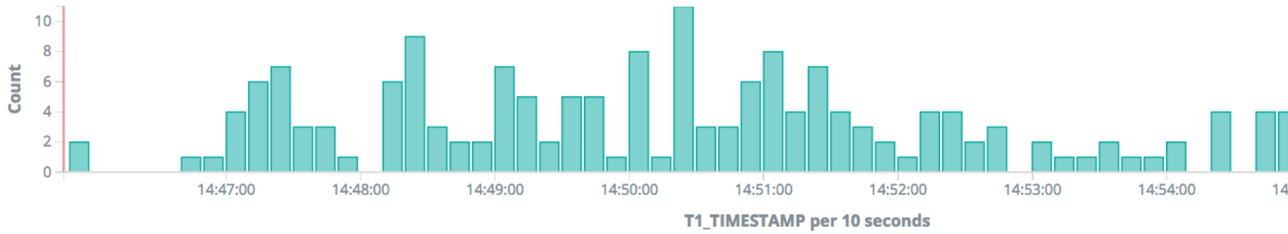
Realtime Operations View & Analysis



Fraudulent ATM transactions

1–50 of 2,760

Time	T1_ATM	T2_ATM	T1_AMOUNT	T2_AMOUNT	ACCOUN
October 8th 2018, 23:22:04.000	ATM : 26235 39932	ATM : 523163 9152	400	300	a553
October 8th 2018, 23:21:30.000	Link	Link; V us			
October 8th 2018, 23:21:04.000	Co-operativ e Bank	ATM : 4326			
October 8th 2018, 23:20:57.000	Barclays	ATM : 9127			
October 8th 2018, 23:20:49.000	ATM : 30460 9846	ATM : 9539			
October 8th 2018, 23:20:46.000	Yorkshire B ank	Coop B			
October 8th 2018, 23:20:35.000	ATM : 25569	Yorksh			



Time	T1_ATM	T2_ATM	DISTANCE_BETWEEN_TXN_KM	MINUTES_DIFFERENCE	T1_AMOUNT
October 9th 2018, 14:56:37.000	Lloyds	ATM : 15558 66943	24.637	3.733	300

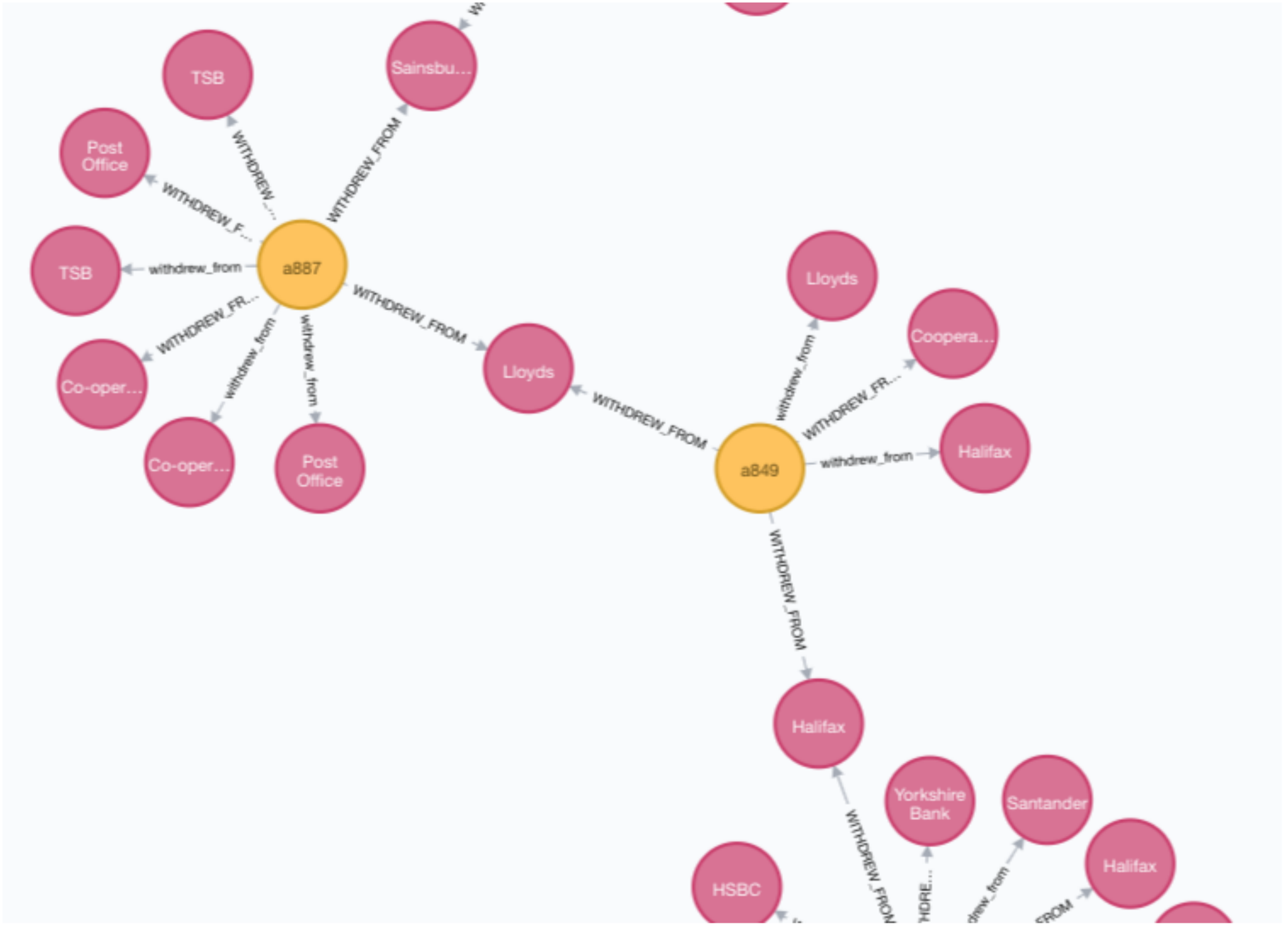
Table | JSON

t	A_ACCOUNT_ID	a532
t	CUSTOMER_ADDRESS	6 Porter Place
t	CUSTOMER_COUNTRY	United Kingdom
t	CUSTOMER_EMAIL	bmaioreo@istockphoto.com
t	CUSTOMER_NAME	Bianca Maior
t	CUSTOMER_PHONE	+44 418 532 2030
#	DISTANCE_BETWEEN_TXN_KM	24.637
#	KMH_REQUIRED	395.959
#	MILLISECONDS_DIFFERENCE	224,000
#	MINUTES_DIFFERENCE	3.733
#	T1_AMOUNT	300
t	T1_ATM	Lloyds
t	T1_LOCATION	53.7221057,-1.8616274

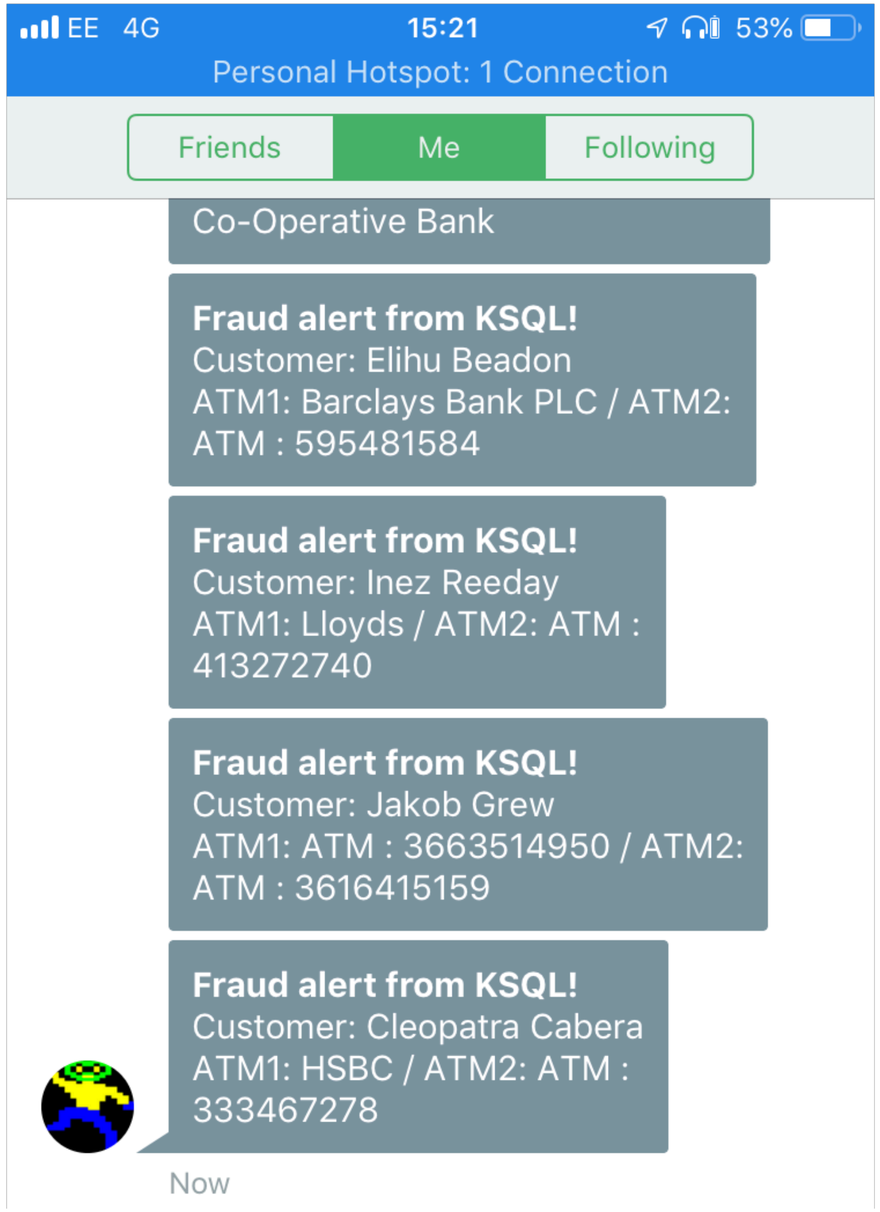
Customer data

Transaction data

Graph analysis

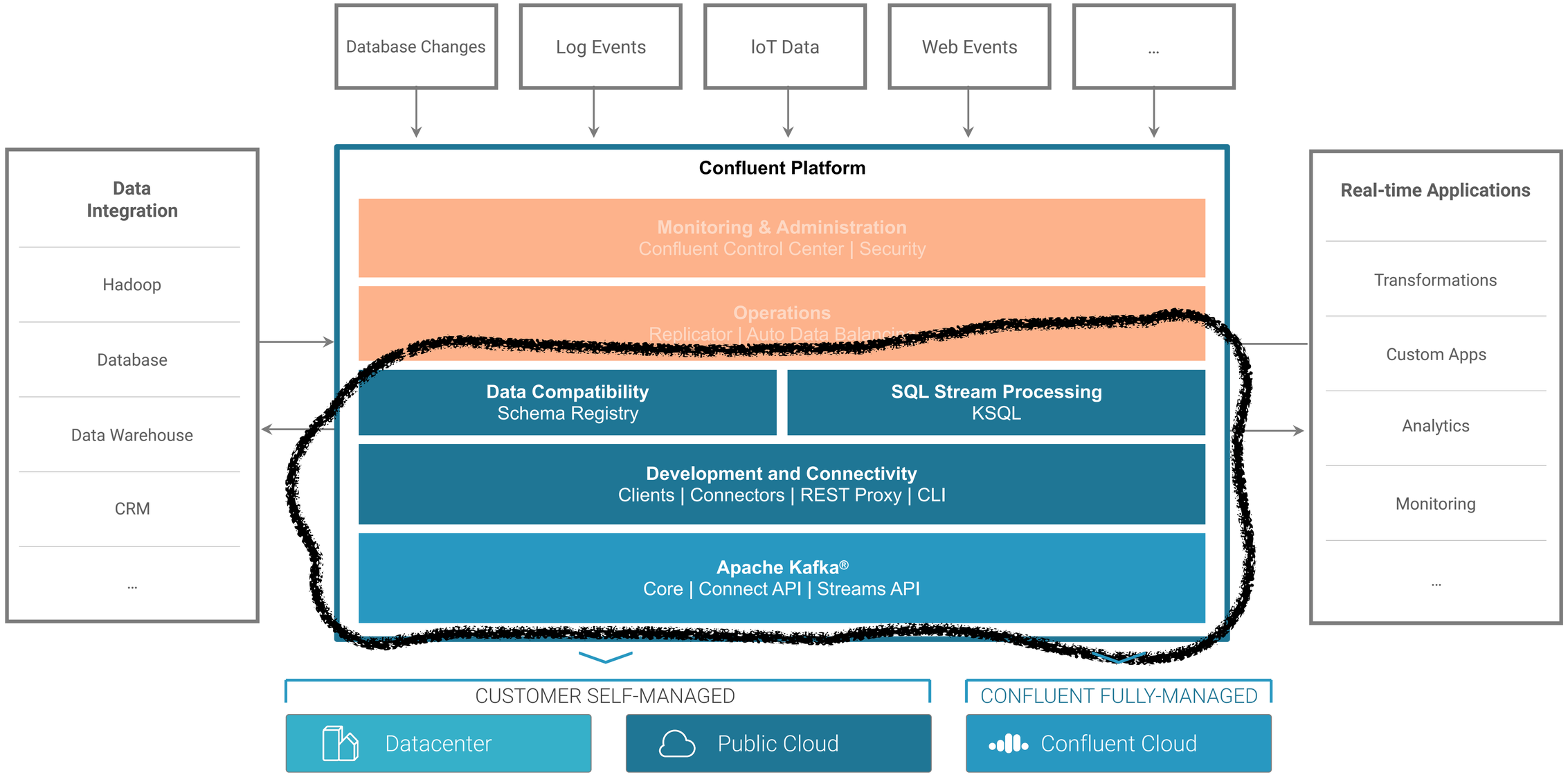


Push notification to the customer



@rmoff

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



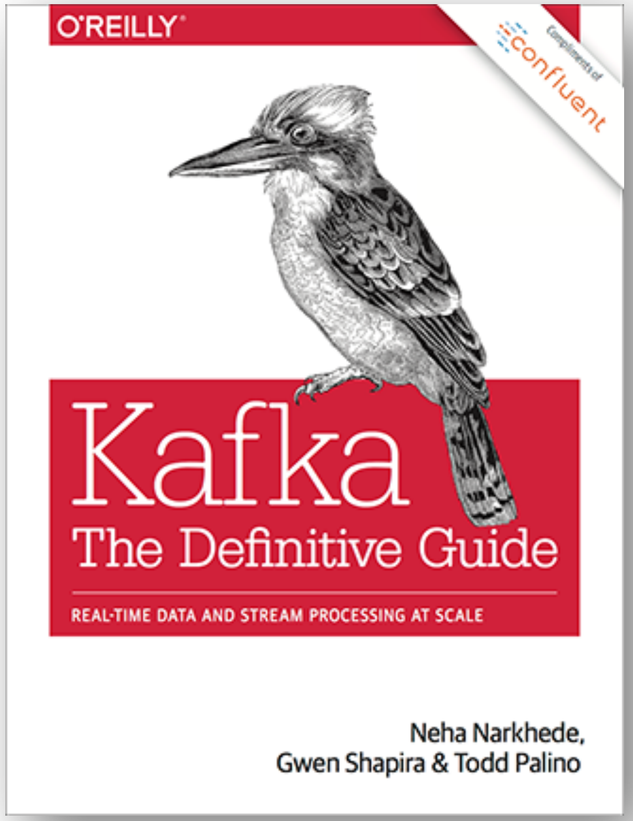
kafka summit

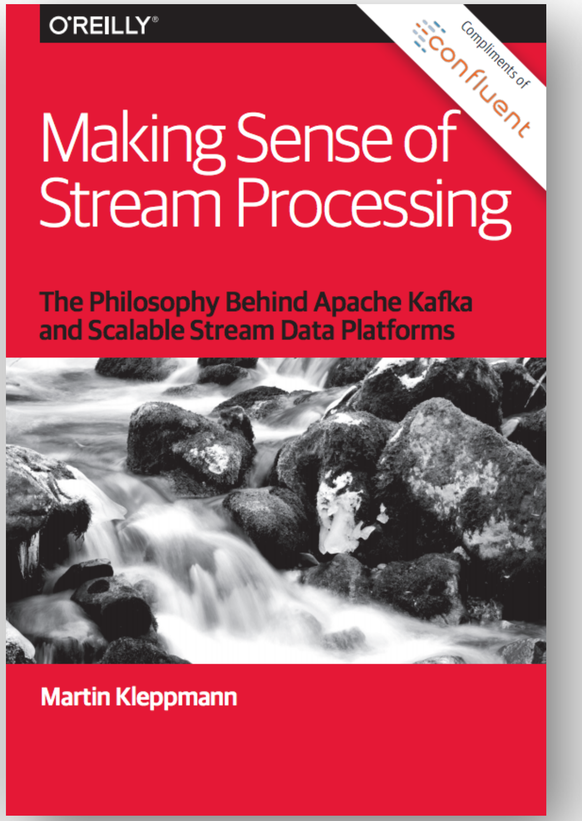
ORGANIZED BY confluent

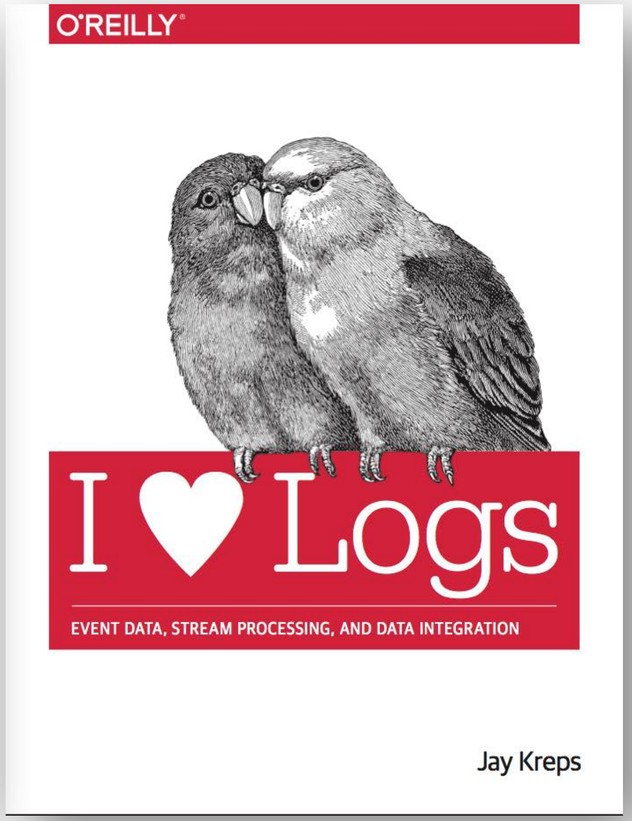
Discount code!
KS19Comm25

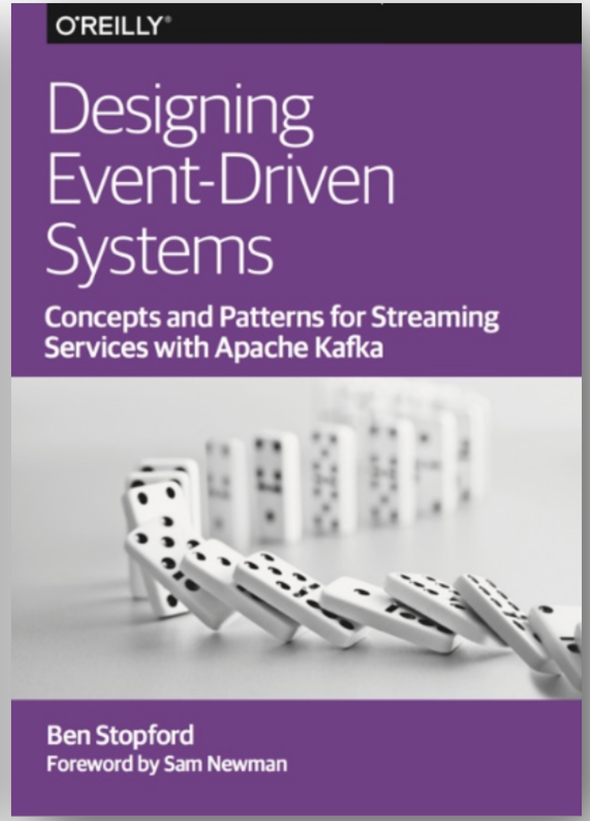
kafka-summit.org

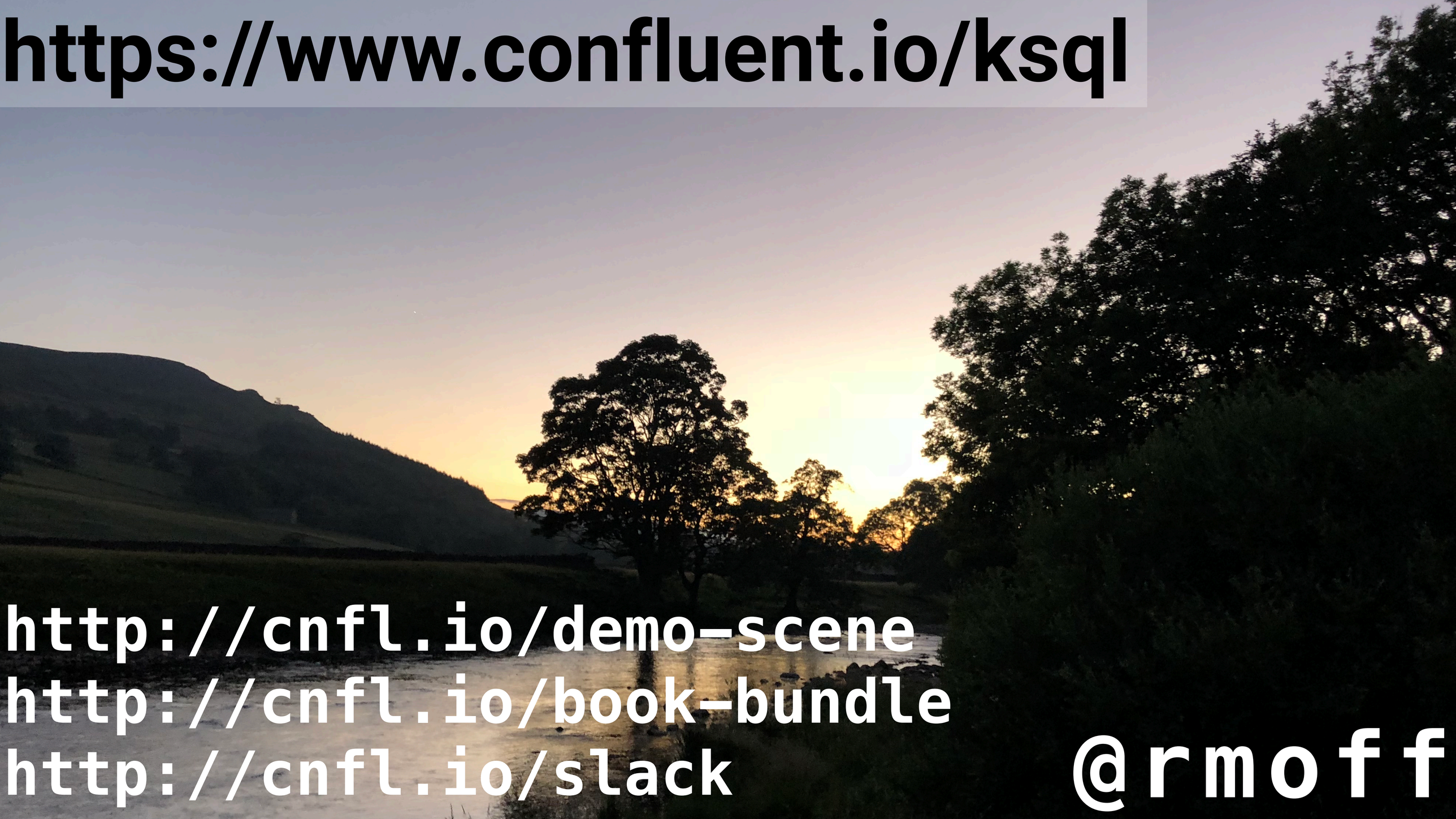
http://cnfl.io/book-bundle










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

#EOF