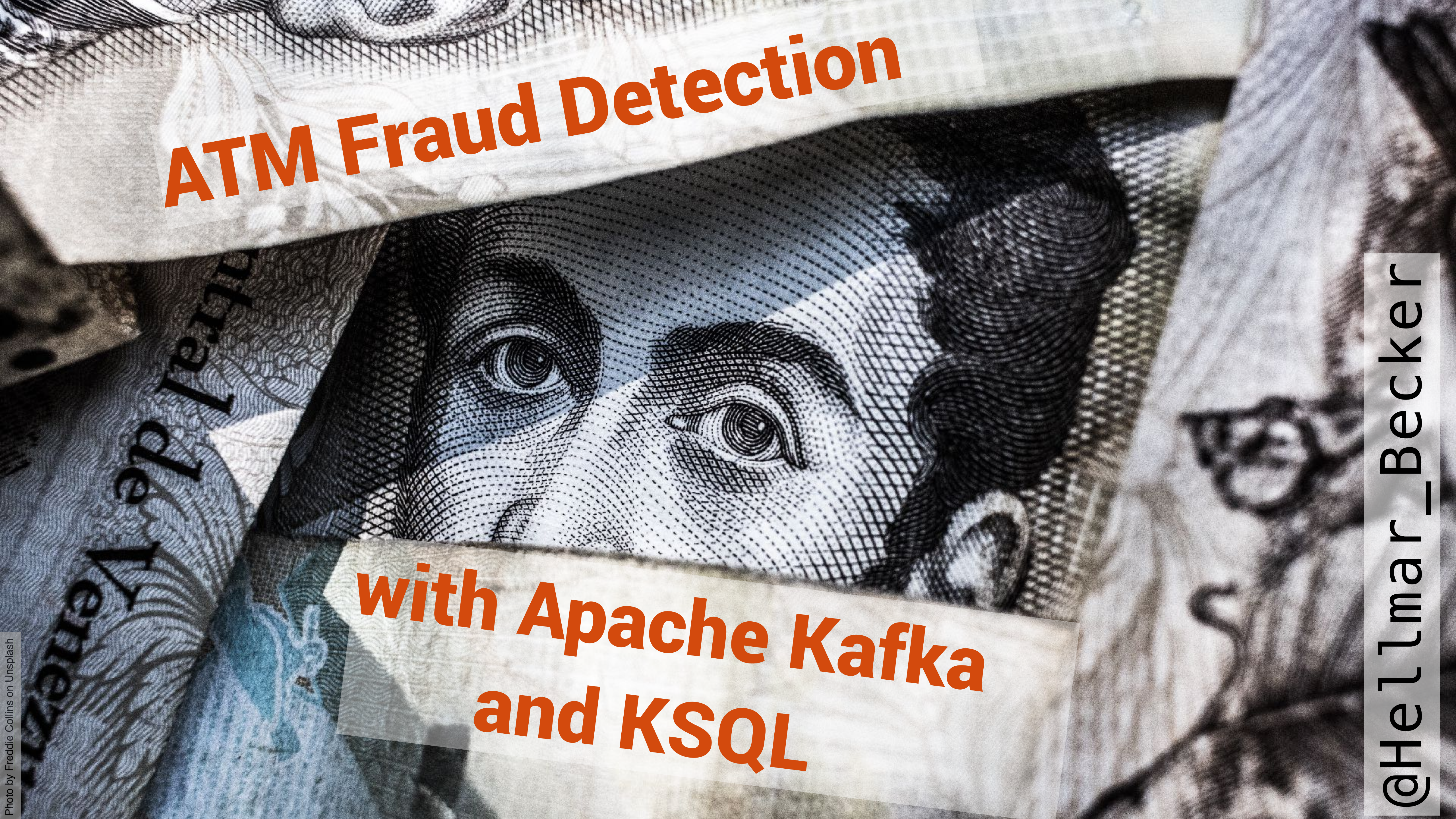




ATM Fraud Detection

**with Apache Kafka
and KSQL**

@Helmar_Becker

ATM






```
{ "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": {"lat": "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

Legit

Dodgy!

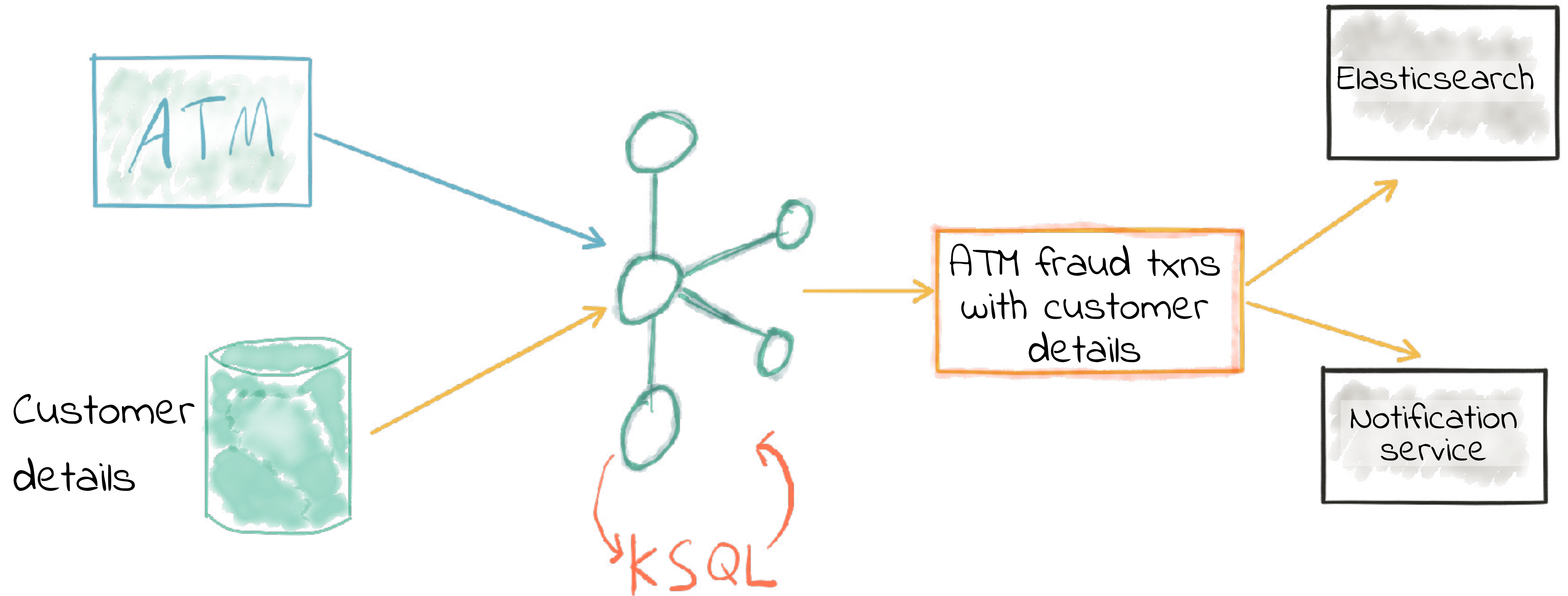
Legit


```
{ "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": {"lat": "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

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


	ACCOUNT_ID	CUSTOMER_NAME	CUSTOMER_PHONE	TX1_ATM	TX2_ATM	MINUTES_DIFFERENCE
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 Shilton	+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

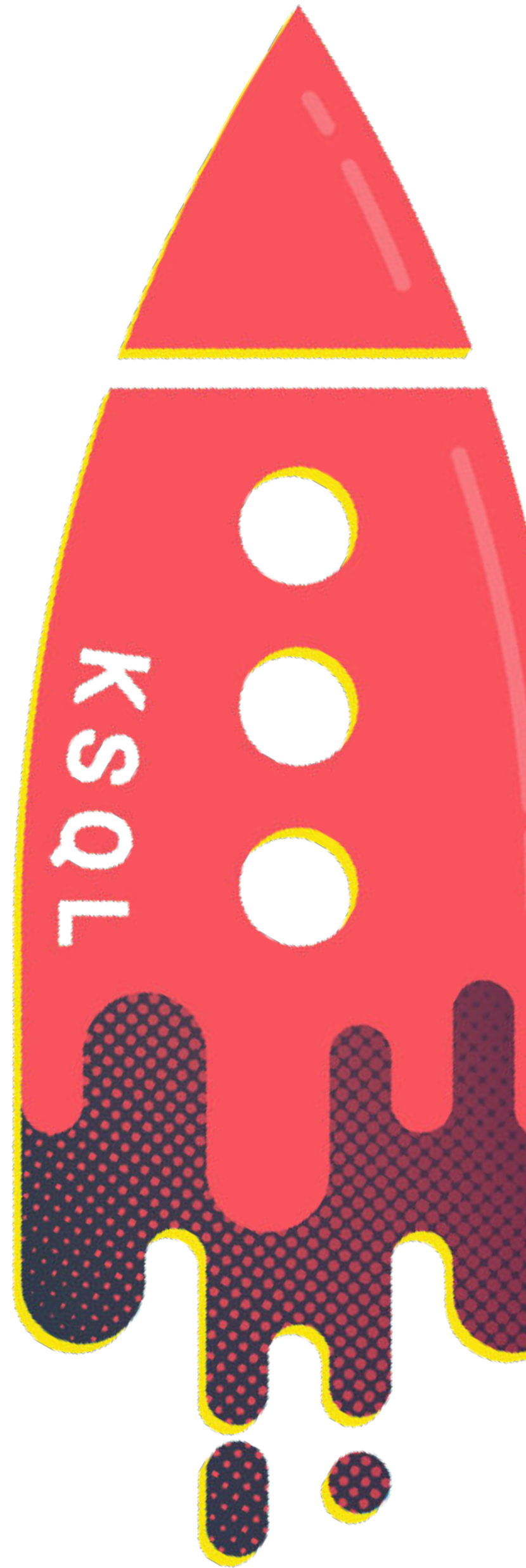
KSQL

is the

Streaming SQL Engine

for

Apache Kafka



KSQL for Real-Time Monitoring

- Log data monitoring, tracking and alerting
- syslog data
- Sensor / IoT data

```
CREATE STREAM SYSLOG_INVALID_USERS AS  
SELECT HOST, MESSAGE  
FROM SYSLOG  
WHERE MESSAGE LIKE '%Invalid user%';
```

<http://cnfl.io/syslogs-filtering> / <http://cnfl.io/syslog-alerting>



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 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

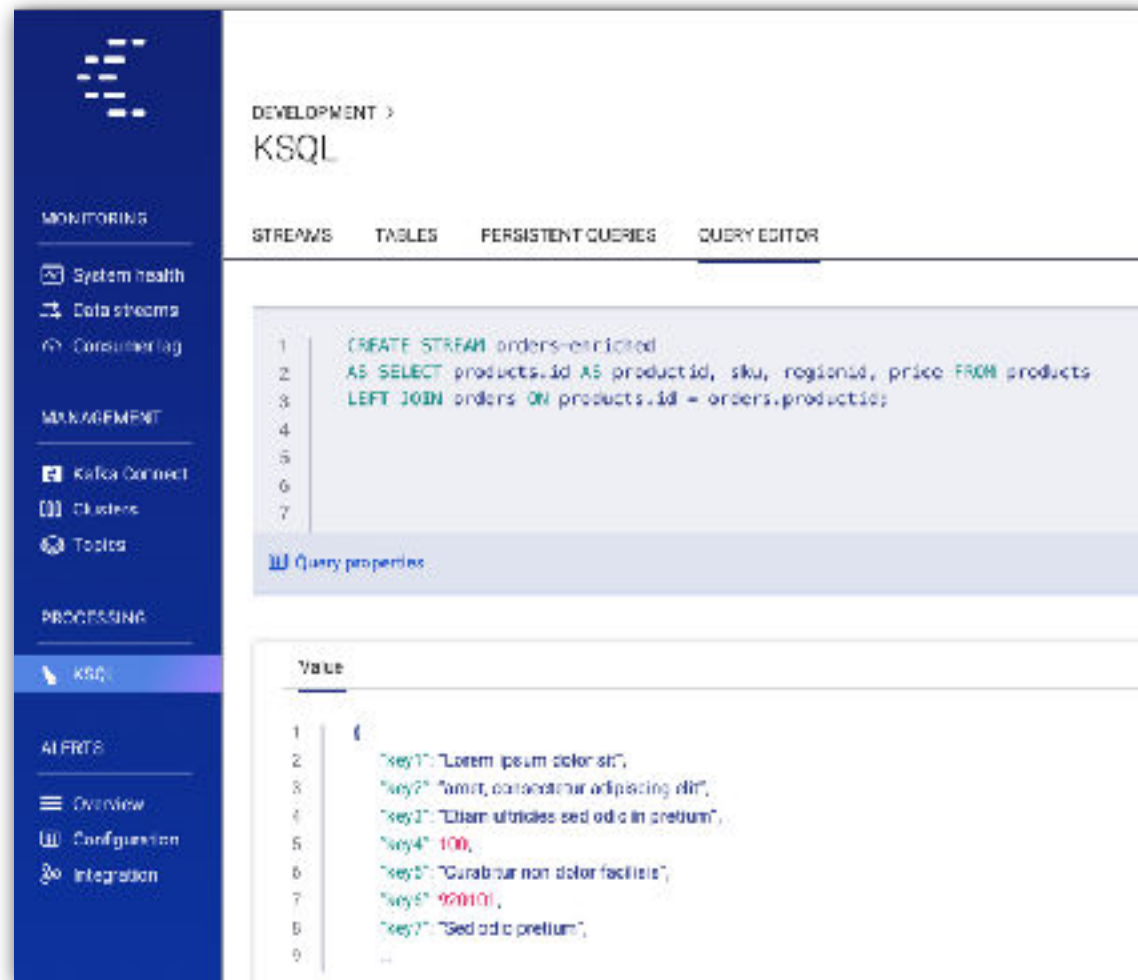
Make simple derivations of existing topics from the command line

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



KSQL in Development and Production

Interactive KSQL
for development and testing

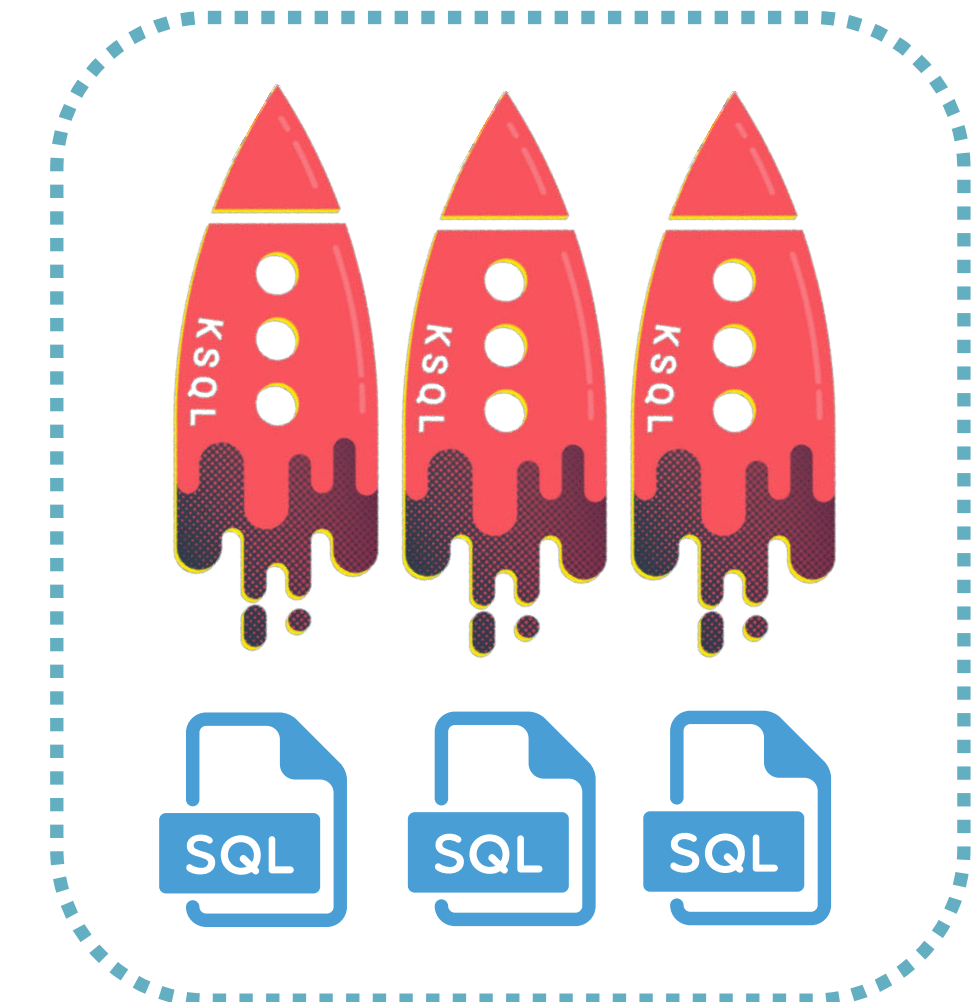


REST



Desired KSQL queries
have been identified

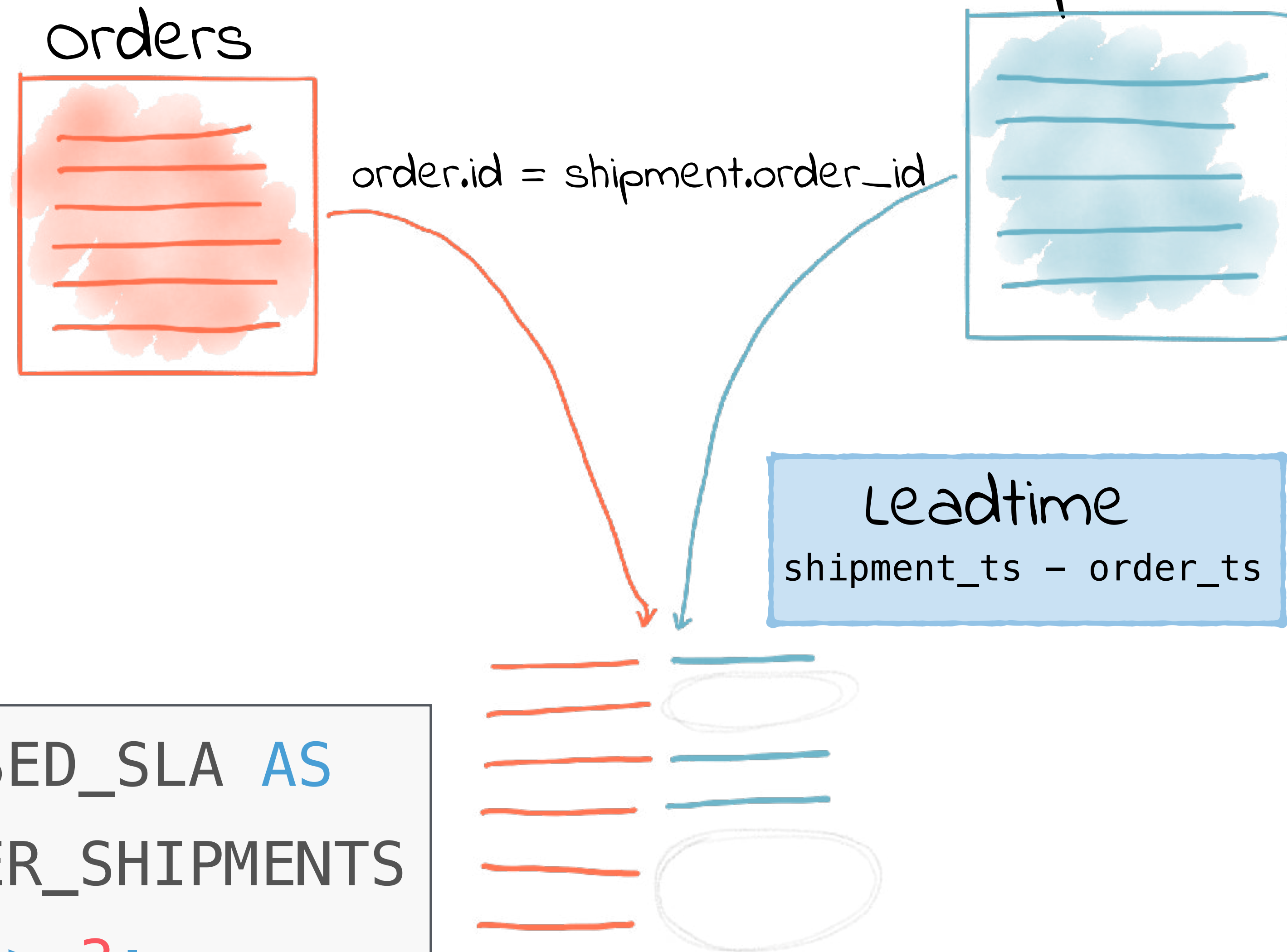
Headless KSQL
for Production



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



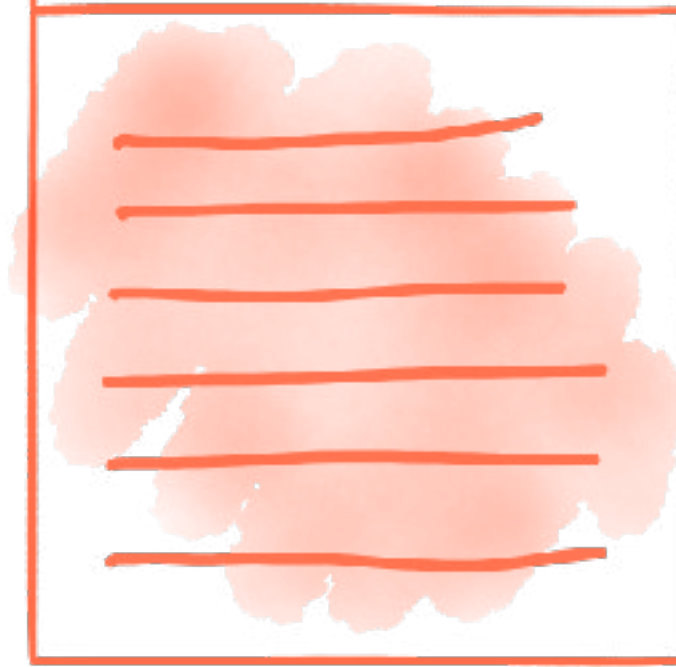
Stream Stream joins



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

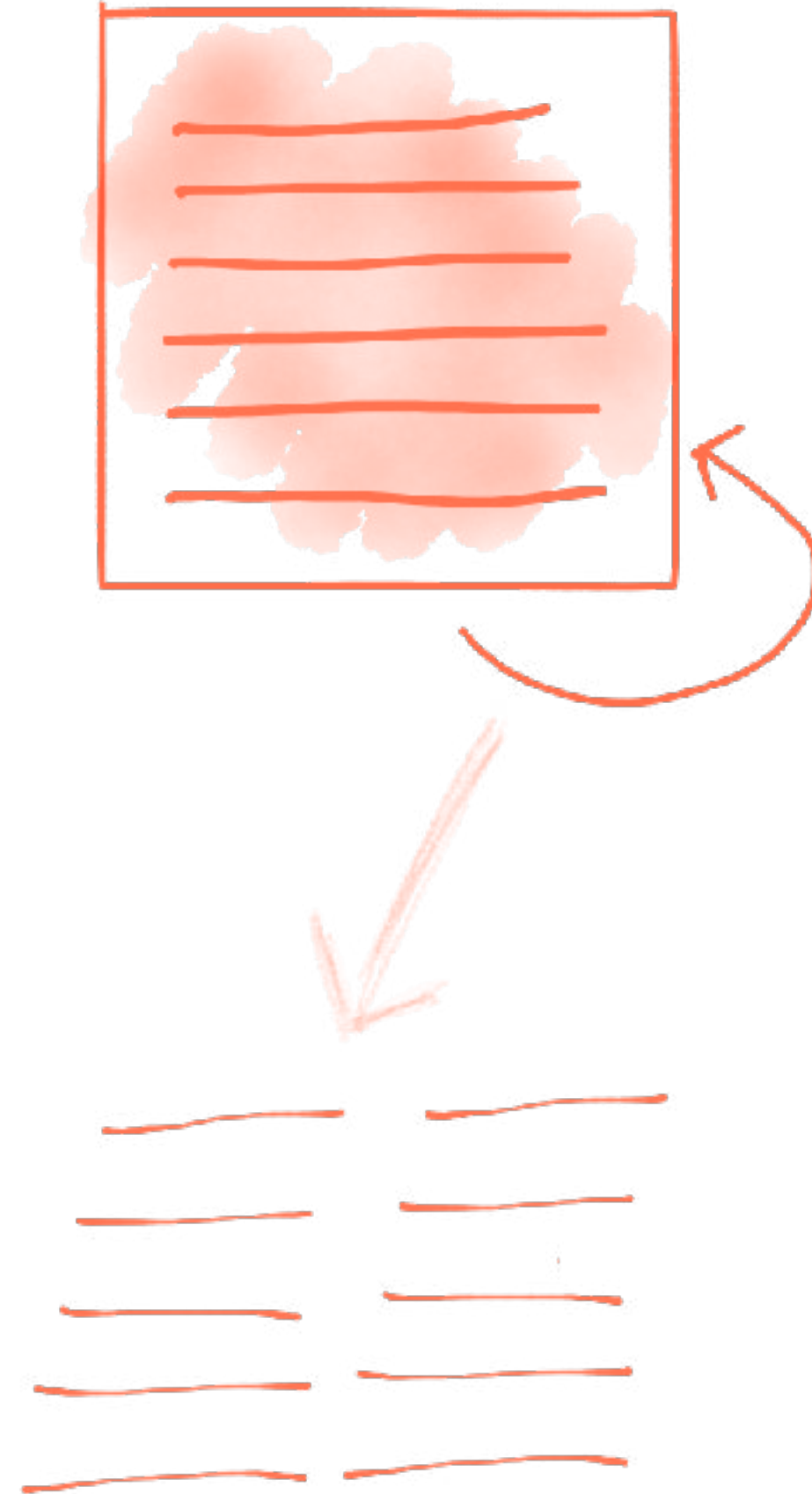

Stream Stream joins

ATM transactions



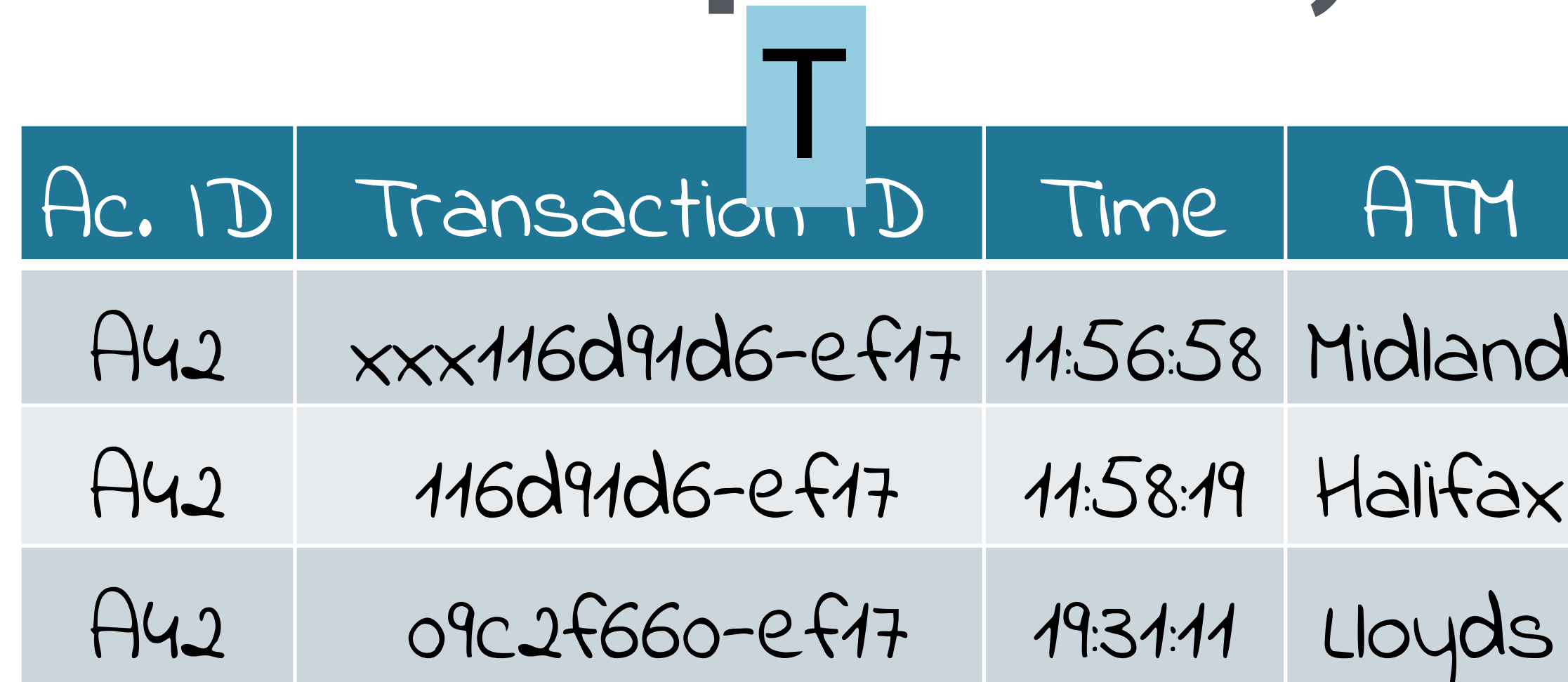
Stream Stream joins

ATM transactions



Demo!

Self-Join (Cartesian product)



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

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

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				T2			
Ac. ID	Transaction ID	Time	ATM	Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland	A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax	A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds	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				T2			
Ac. ID	Transaction ID	Time	ATM	Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland	A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax	A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds	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				T2			
Ac. ID	Transaction ID	Time	ATM	Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland	A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax	A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds	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				T2			
Ac. ID	Transaction ID	Time	ATM	Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland	A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax	A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds	A42	09c2f660-ef17	19:31:11	Lloyds

WITHIN 10 MINUTES

WHERE

T1.TRANSACTION_ID !=

T2.TRANSACTION_ID

Only join *forward*

T1				T2			
Ac. ID	Transaction ID	Time	ATM	Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland	A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax	A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds	A42	09c2f660-ef17	19:31:11	Lloyds

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

Only join *forward*

T1				T2			
Ac. ID	Transaction ID	Time	ATM	Ac. ID	Transaction ID	Time	ATM
A42	xxx116d91d6-ef17	11:56:58	Midland	A42	xxx116d91d6-ef17	11:56:58	Midland
A42	116d91d6-ef17	11:58:19	Halifax	A42	116d91d6-ef17	11:58:19	Halifax
A42	09c2f660-ef17	19:31:11	Lloyds	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

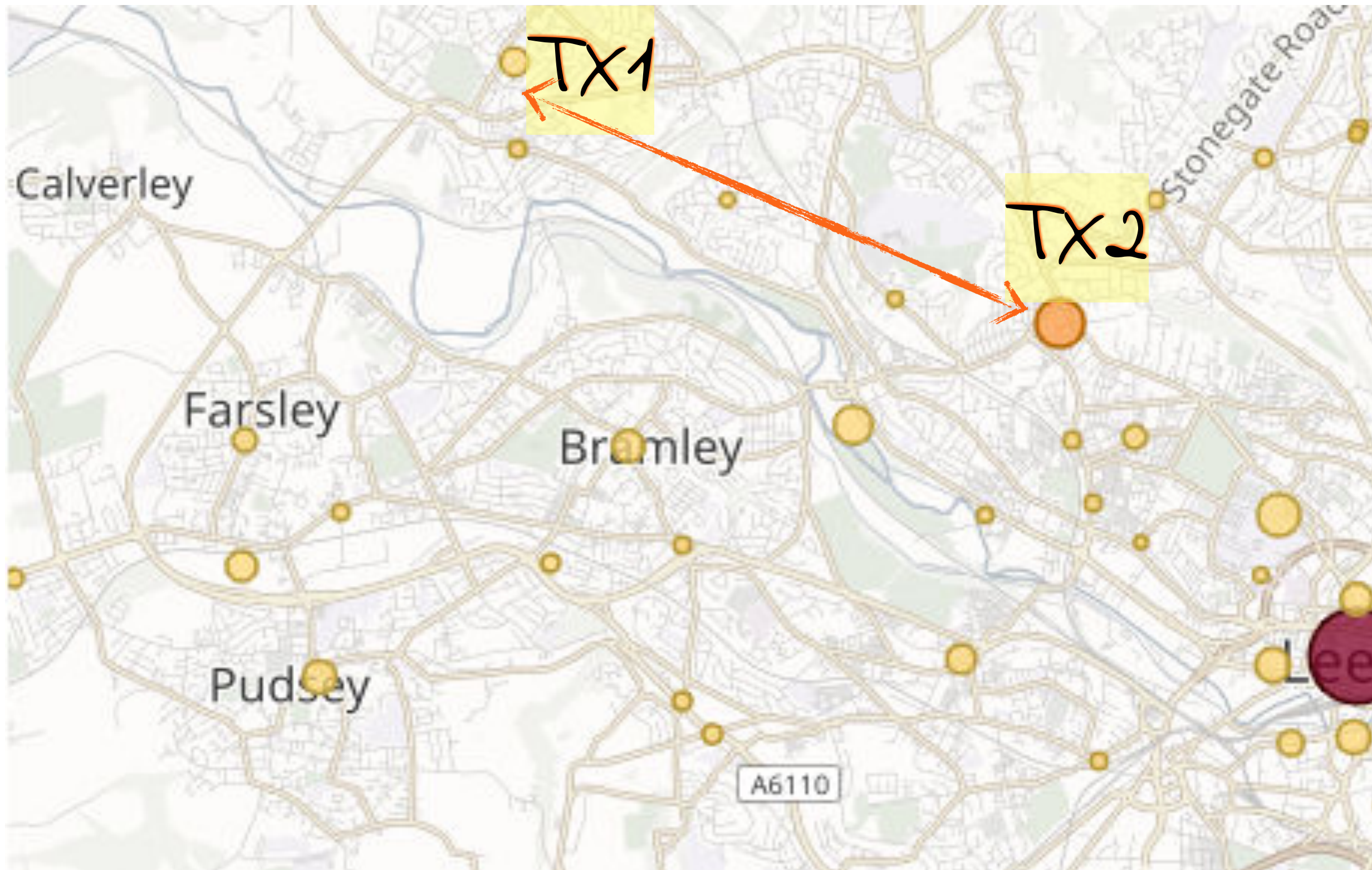
Legit

Dodgy!





Calculate distance between ATMs



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


Calculate time between transactions

TX2.ROWTIME – TX1.ROWTIME AS
MILLISECONDS_DIFFERENCE

(TX2.ROWTIME – TX1.ROWTIME)
/ 1000 / 60 / 60 AS HOURS_DIFFERENCE

$\text{GEO_DISTANCE}(\dots) / \text{HOURS_DIFFERENCE}$
AS KMH_REQUIRED



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",
```

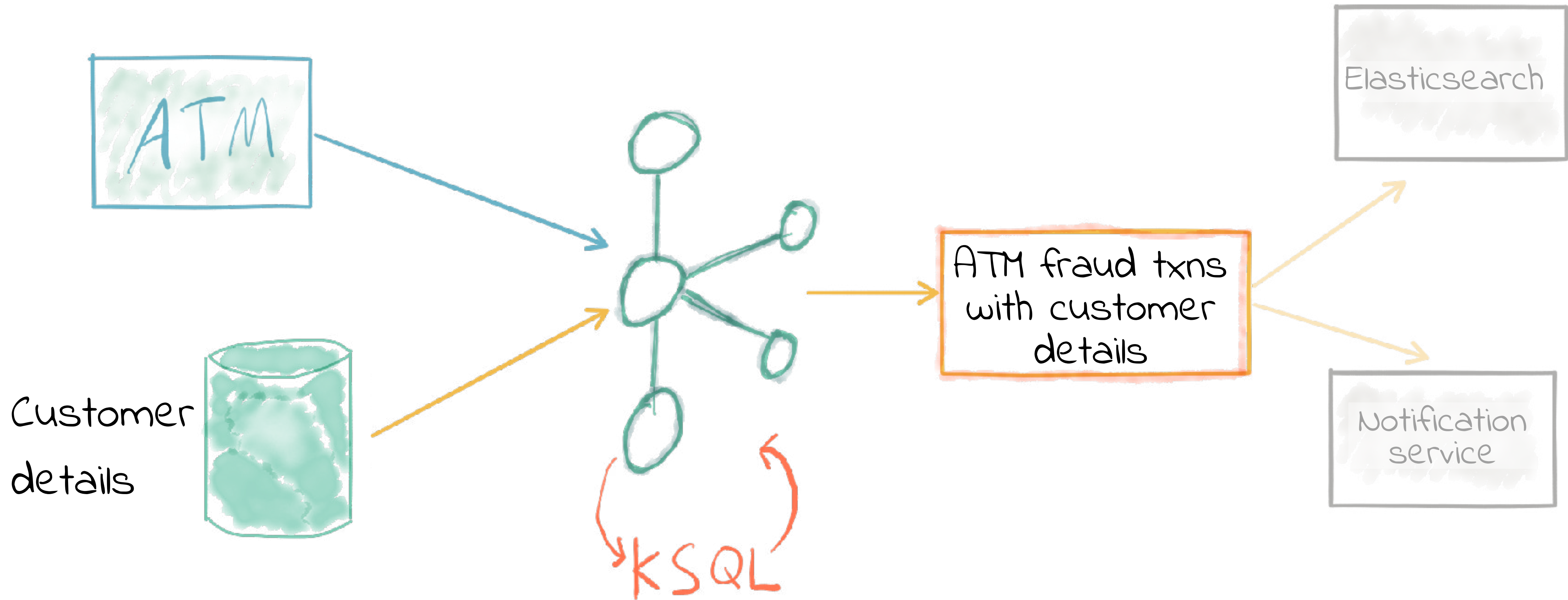
```
"offset": 111111111
```


But what about the account holder?



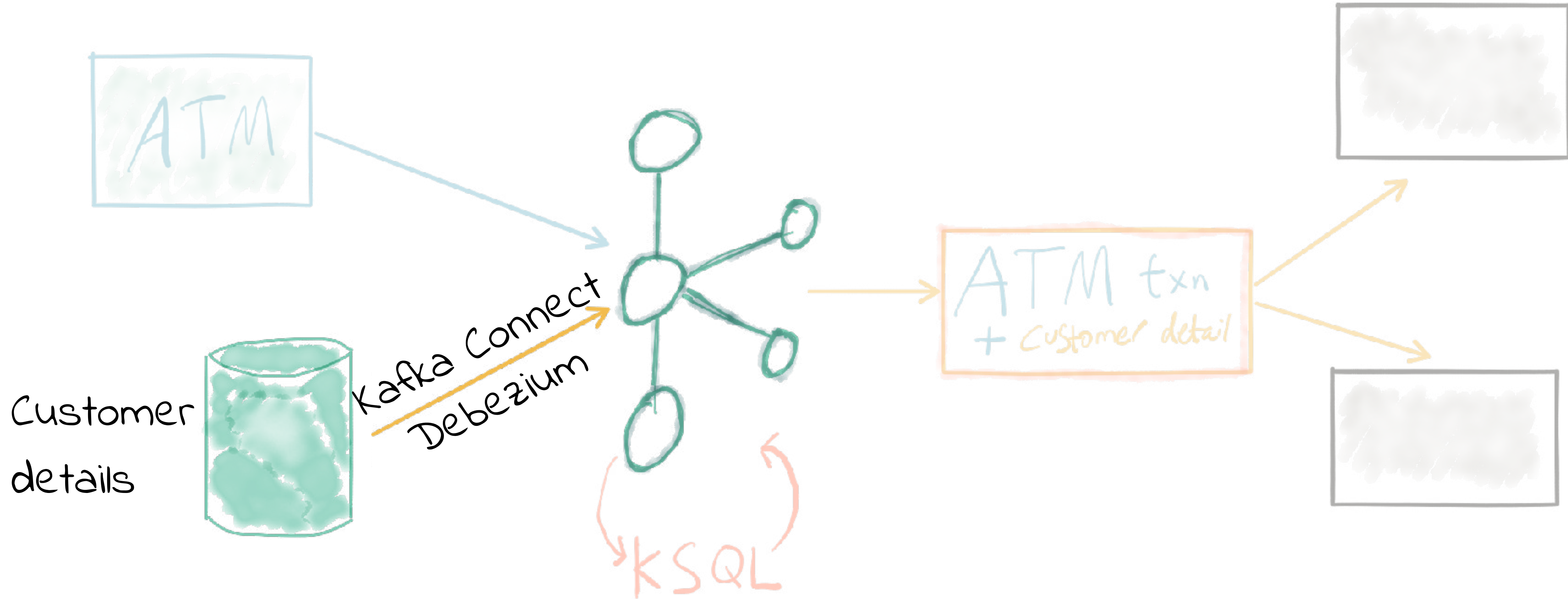
mysql> SELECT ACCOUNT_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE FROM accounts LIMIT 5;

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



1. Enrich transaction events with customer data

Demo Time!



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

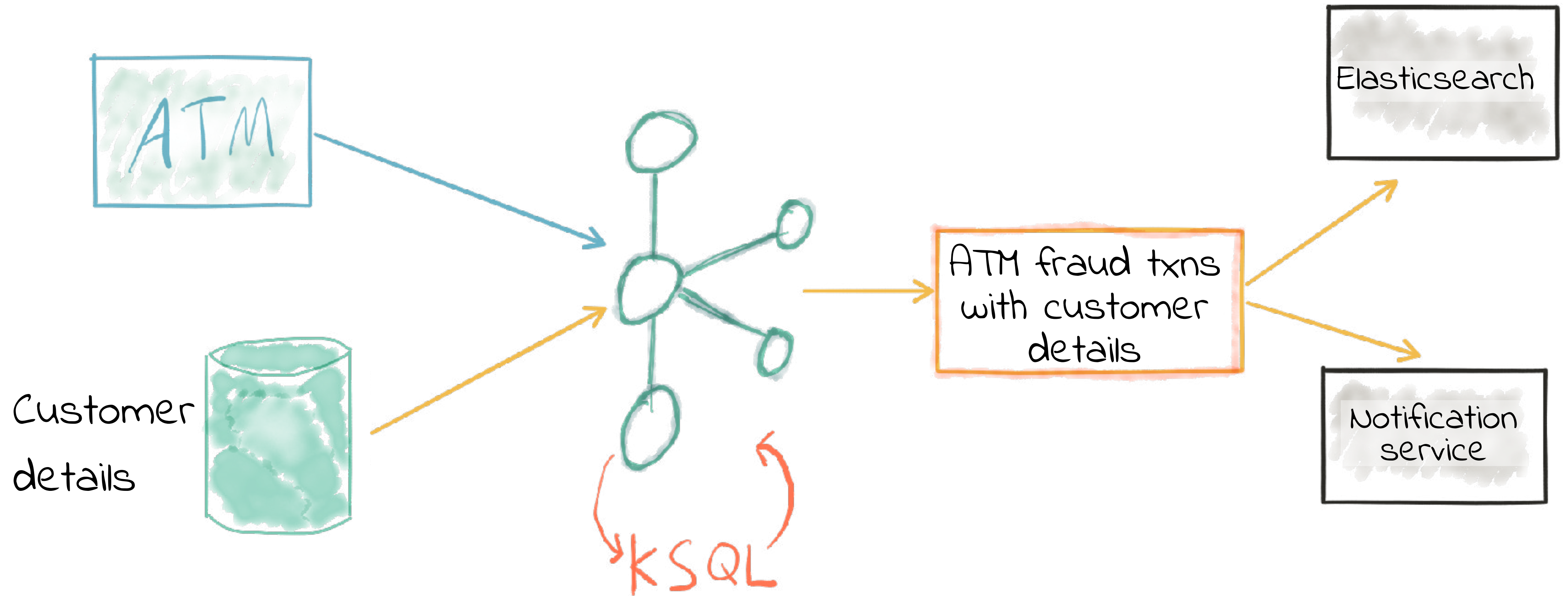
Suspect Transactions

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

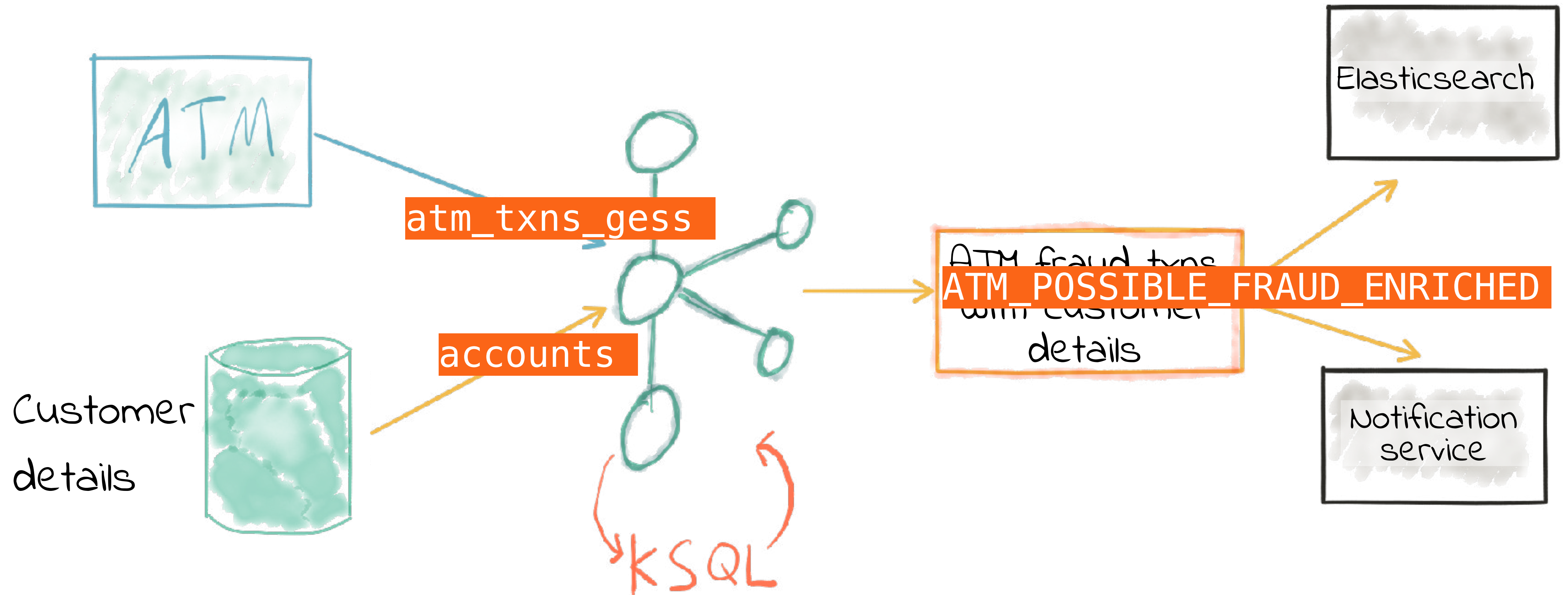
→ Dodgy!

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



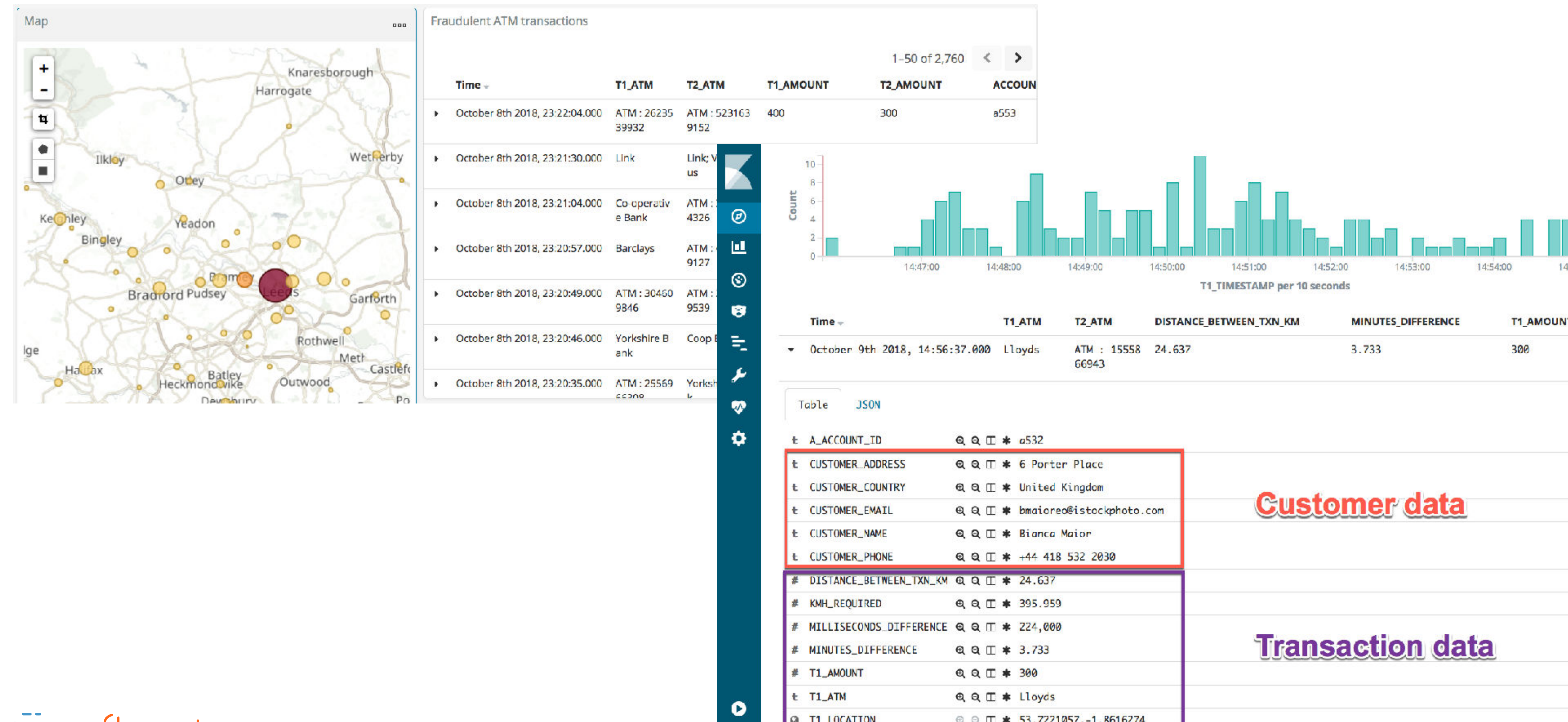
1. Spot fraud in stream of transactions
2. Enrich transaction events with customer data



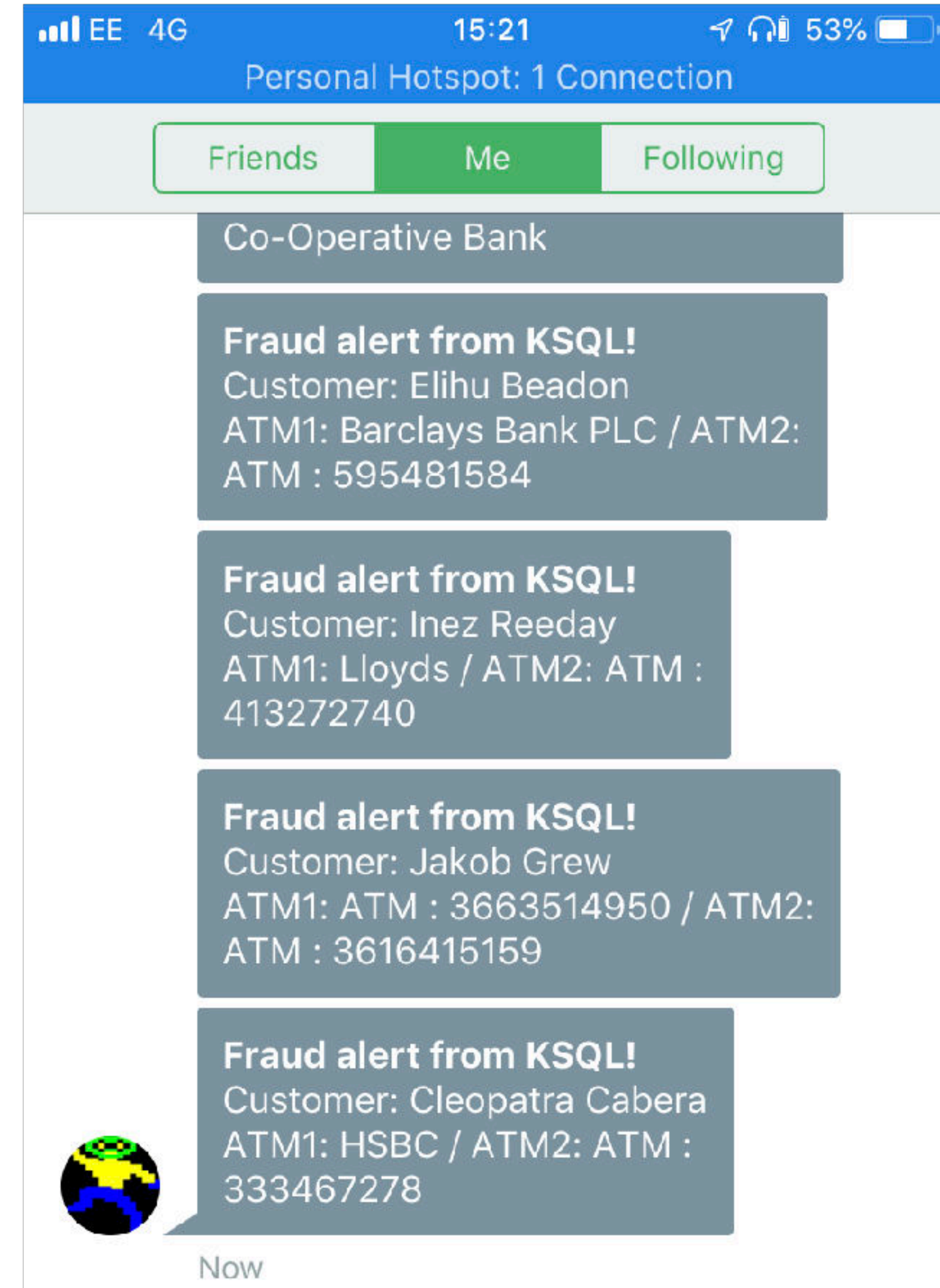
1. Spot fraud in stream of transactions
2. Enrich transaction events with customer data



Realtime Operations View & Analysis

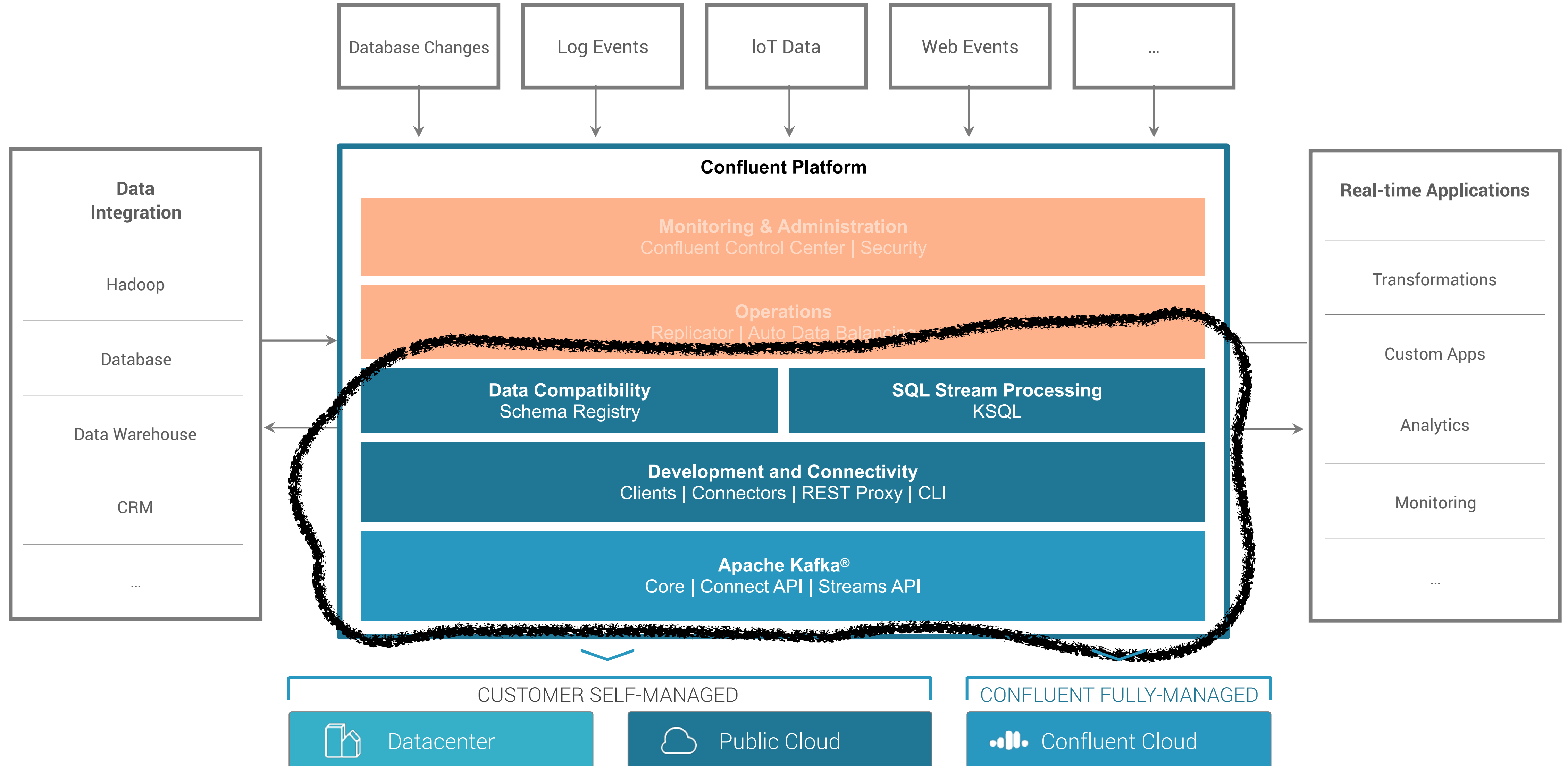


Push notification to the customer



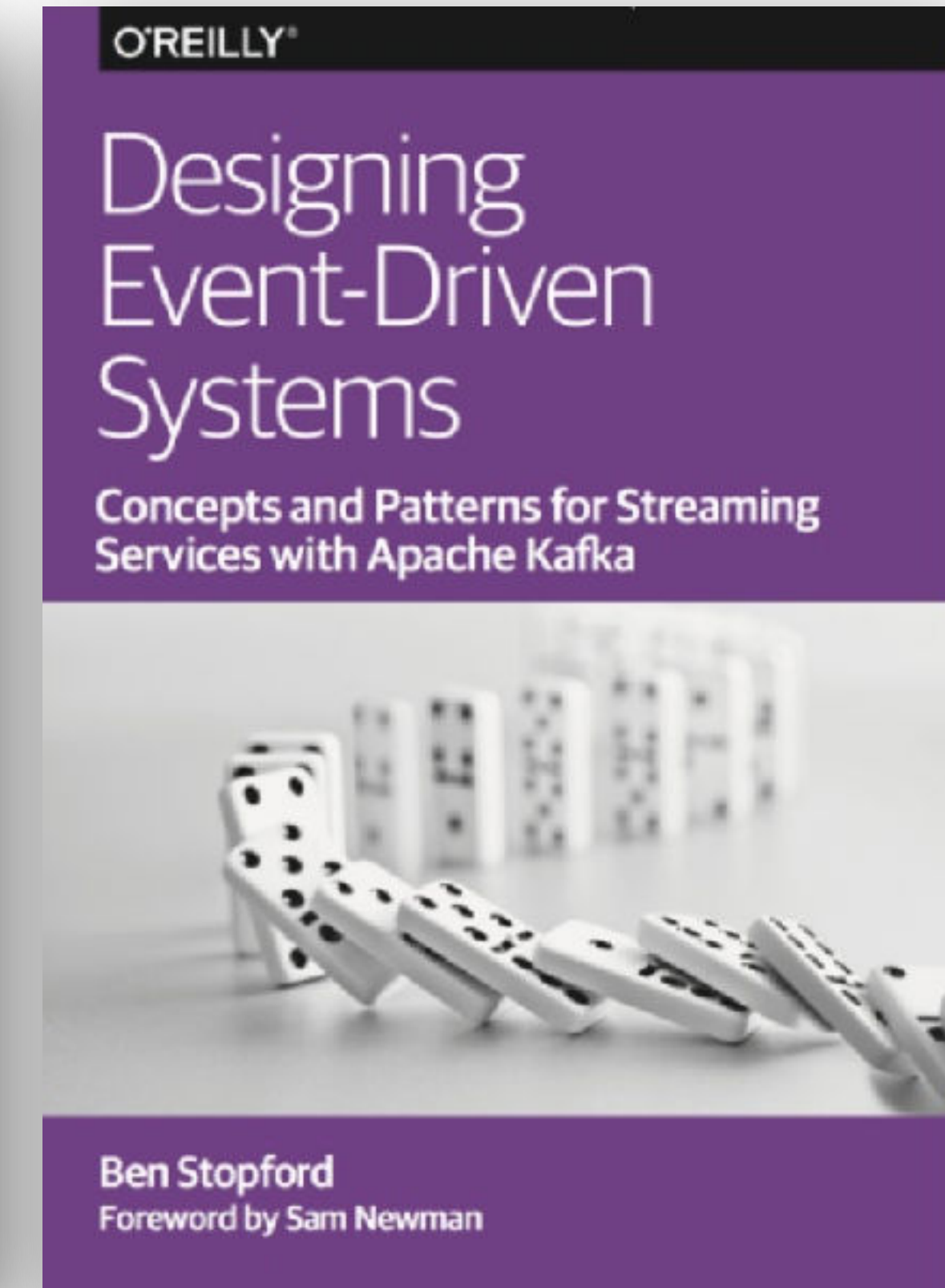
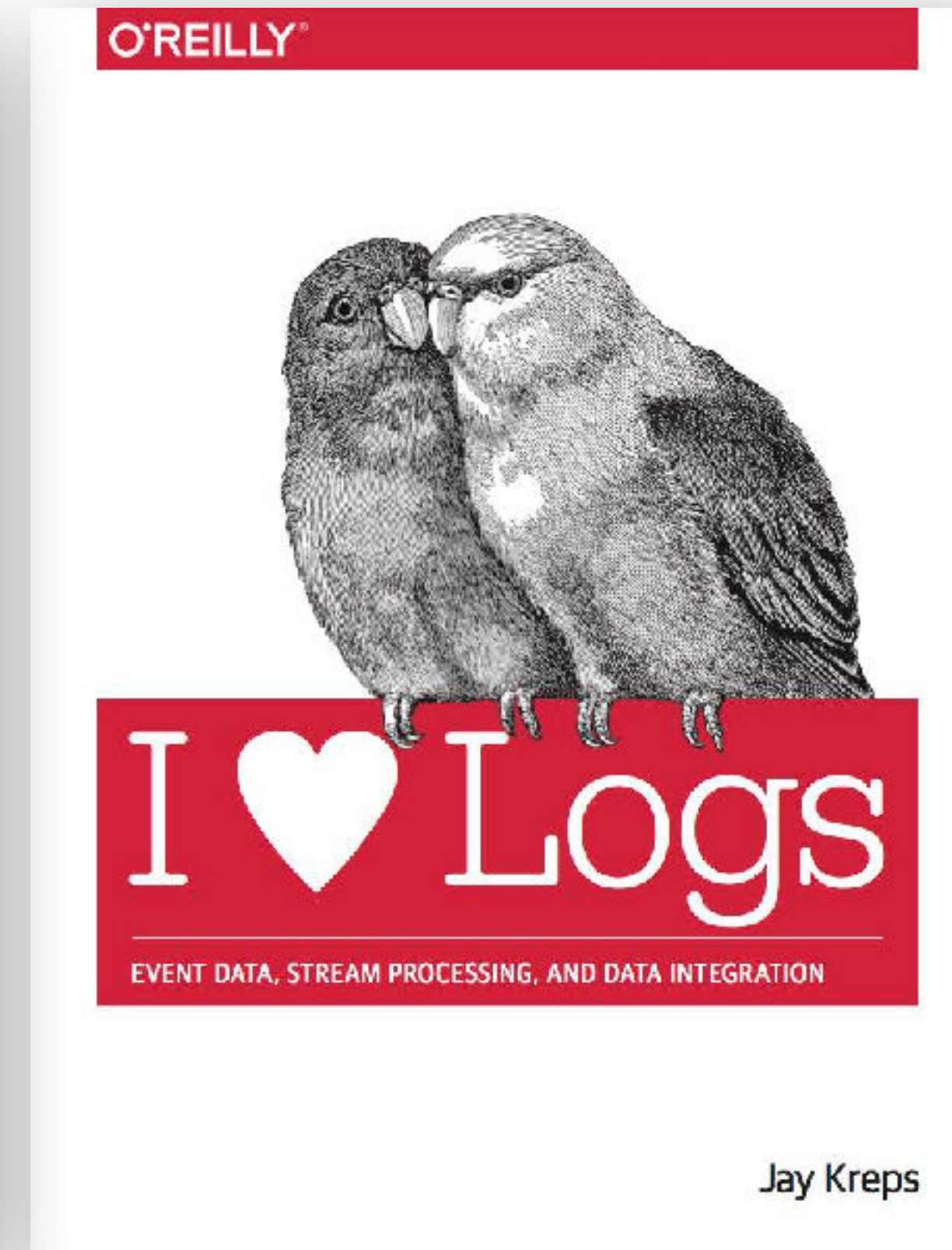
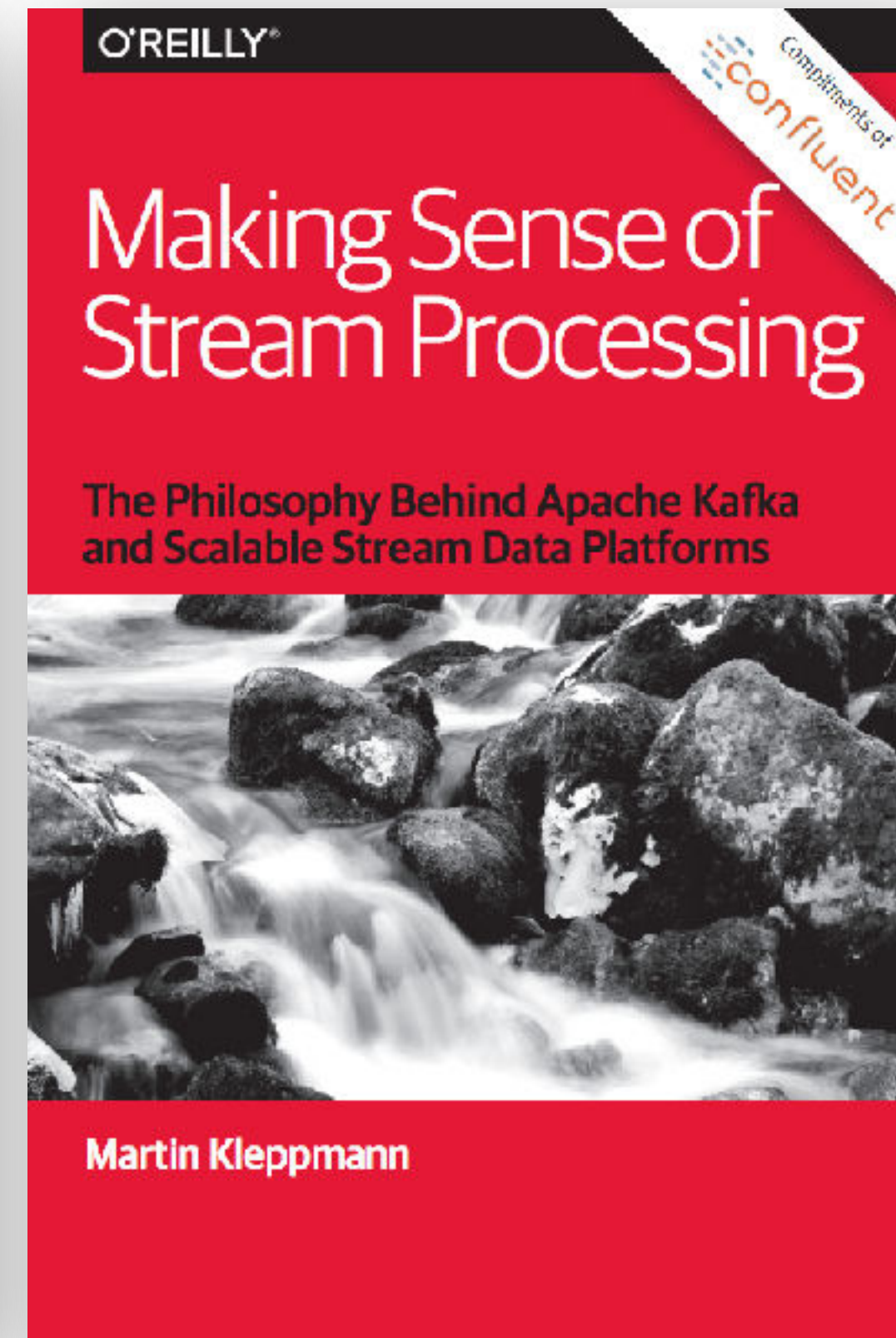
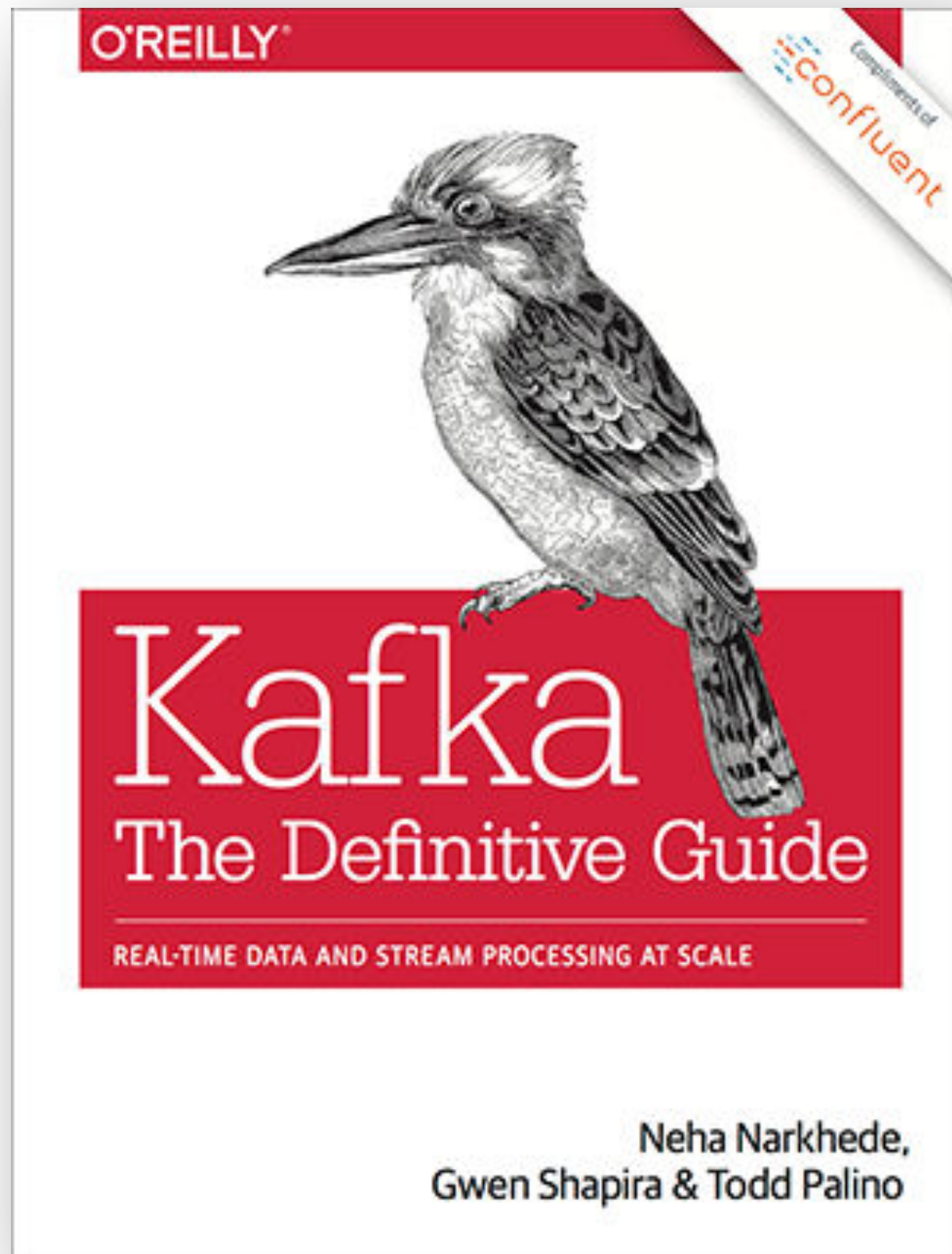
Confluent Community Components

Apache Kafka with a bunch of cool stuff! For free!



Free Books!

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



**[https://www.confluent.io/
ksql](https://www.confluent.io/ksql)**

<http://cnfl.io/slack>



<https://cnfl.io/demo-scene>

**@Hellmar_Becker
hellmar@confluent.io**

#EOF