

BUILDING STREAM PROCESSING APPLICATIONS WITH APACHE KAFKA AND KSQL



I BUILD HIGHLY SCALABLE

Hello World

APPS

A middle-aged man with a friendly expression stands in a vast, open field. He is wearing a grey baseball cap, a blue and white plaid button-down shirt, and blue denim overalls. The background features rolling green hills under a clear, bright sky. The overall tone is positive and appreciative of rural life.

IT AIN'T MUCH

BUT IT'S HONEST WORK



I GUESS YOU FOLKS ARE NOT READY FOR THIS YET

BUT YOUR KIDS ARE GONNA LOVE IT

STREAM PROCESSING

PROCESSING

STREAM

PROCESSING

A

STREAM

OF EVENTS

**STREAMS
ARE
EVERYWHERE**

A CUSTOMER EXPERIENCE



A SALE



A SENSOR READING





AN APPLICATION LOG AGGREGATION



DATABASES



STREAMS OF EVENTS



TIME



STREAM PROCESSING WITH KSQL



STREAM: widgets



STREAM: widgets_red

STREAM PROCESSING WITH KSQL



STREAM:
widgets

KSQL
↓

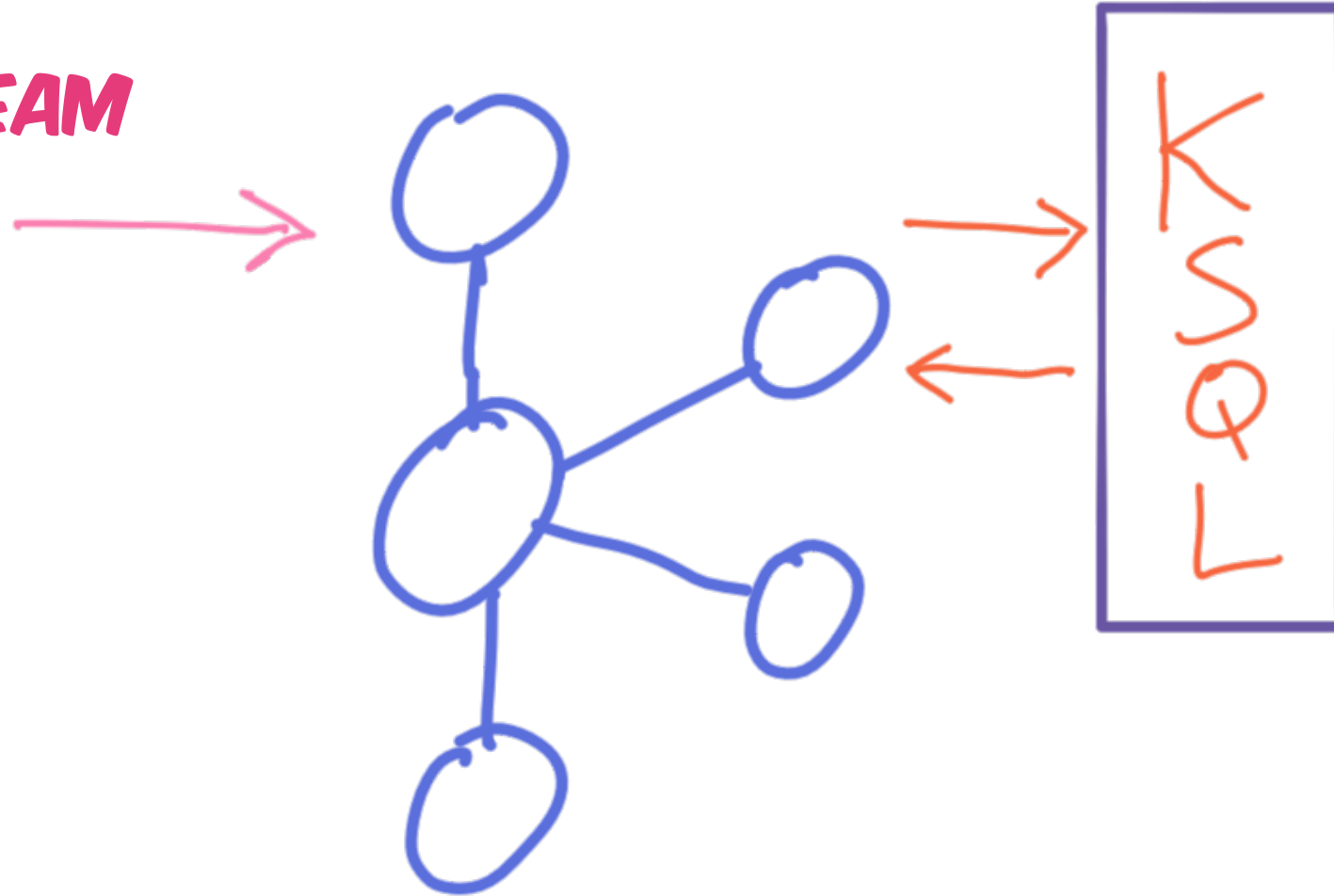
```
CREATE STREAM widgets_red AS  
SELECT * FROM widgets  
WHERE colour='RED';
```



STREAM:
widgets_red

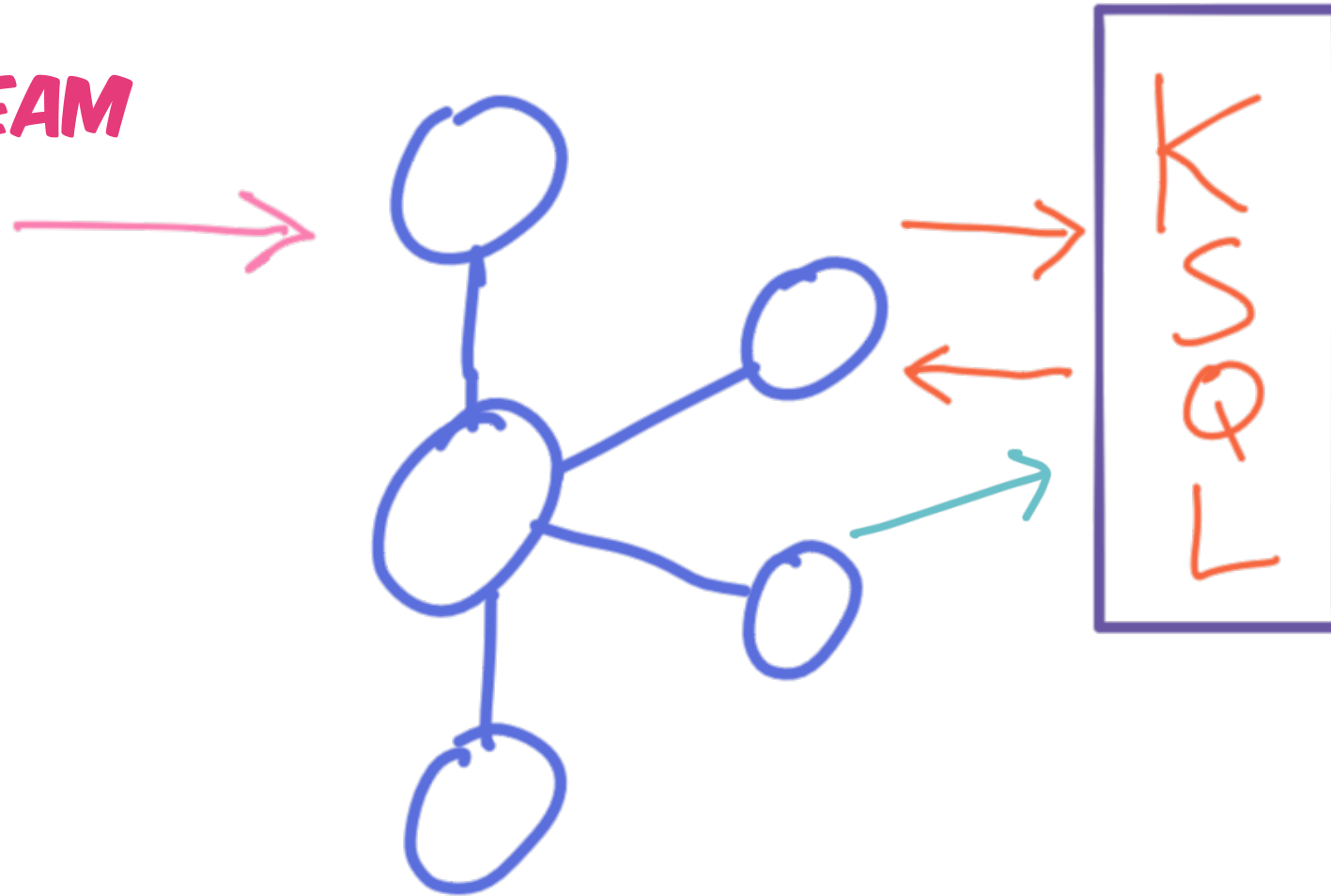
STREAM PROCESSING WITH KSQL

SOURCE STREAM



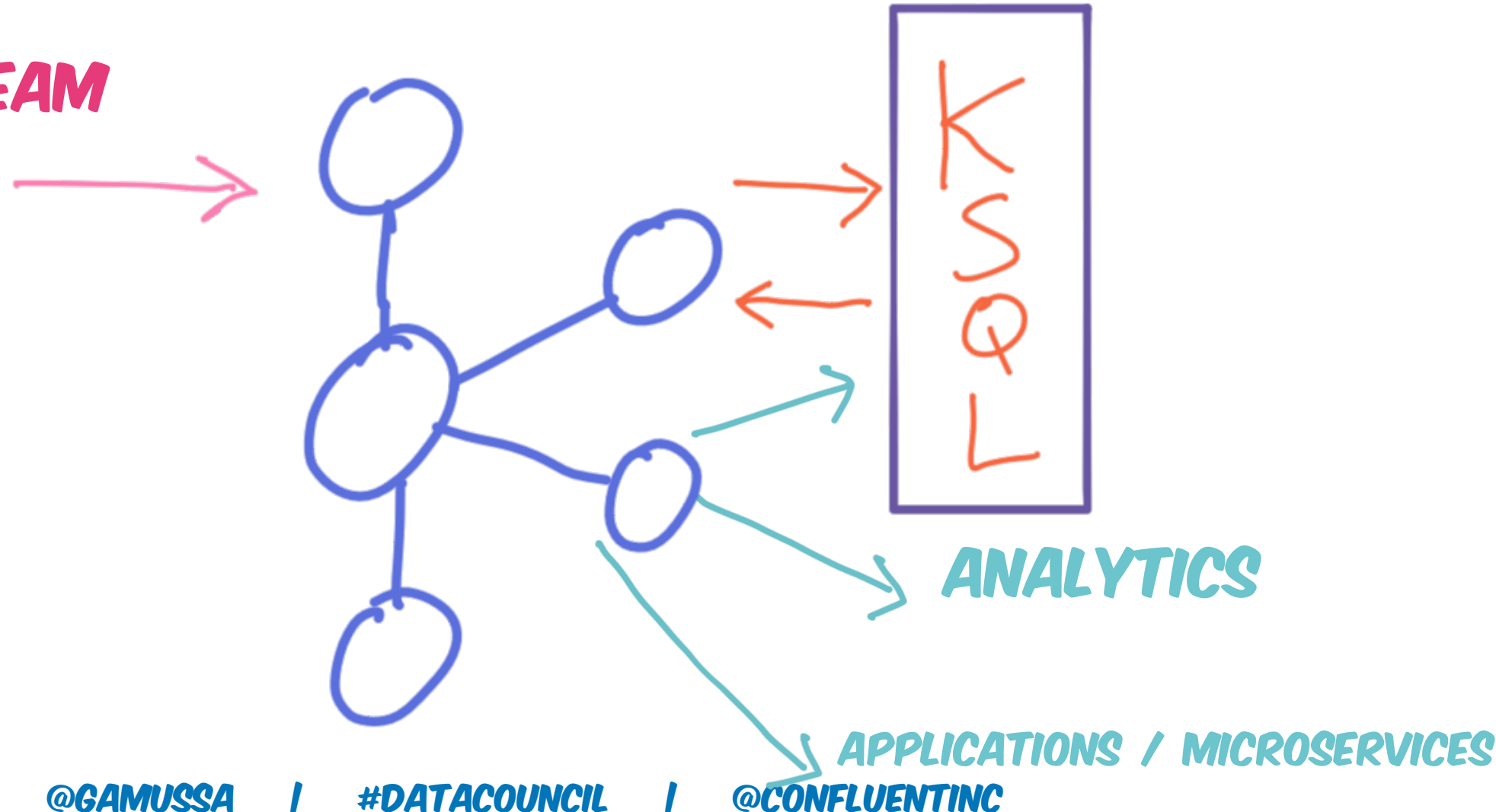
STREAM PROCESSING WITH KSQL

SOURCE STREAM



STREAM PROCESSING WITH KSQL

SOURCE STREAM



INTERACTING WITH KSQL

KSQL - CONFLUENT CONTROL CENTER

The screenshot displays the Confluent Control Center KSQL Editor interface. At the top, there are four tabs: KSQL EDITOR, STREAMS, TABLES, and RUNNING QUERIES. The KSQL EDITOR tab is active, showing a KSQL query in a text editor with line numbers 1 through 4. The query is: `SELECT MAKE, COUNT(*) AS ORDER_COUNT FROM ORDERS_ENRICHED GROUP BY MAKE;`. Below the editor, there is a section for 'Query properties' with a green dot icon. The 'Query Results' section is visible, showing a table with two columns: MAKE and ORDER_COUNT. The table contains two rows of data: 'Fahey and Sons' with an order count of 113, and 'Harris, Poulos and Kreiger' with an order count of 845. To the left of the table, there is a summary of query statistics.

MAKE	ORDER_COUNT
Fahey and Sons	113
Harris, Poulos and Kreiger	845

Query Results

Data structure	STREAM
Total messages	16580
Messages/sec	85.86
Total message bytes	--

KSQL - CLI

```
CLI v5.2.2, Server v5.2.2 located at http://ksql-server:8088
```

```
Having trouble? Type 'help' (case-insensitive) for a rundown of how things work!
```

```
ksql>
```

```
ksql> CREATE STREAM ORDERS_NO_ADDRESS_DATA AS
```

```
> SELECT TIMESTAMPTOSTRING(ROWTIME, 'yyyy-MM-dd HH:mm:ss') AS ORDER_TIMESTAMP,  
>         ORDERID,  
>         ITEMID,  
>         ORDERUNITS  
> FROM ORDERS;
```

```
Message
```

```
-----  
Stream created and running  
-----
```

```
ksql> select * from ORDERS_NO_ADDRESS_DATA;
```

```
1562059702636 | 0 | 2019-07-02 09:28:22 | 0 | Item_48 | 16  
1562059703535 | 0 | 2019-07-02 09:28:23 | 0 | Item_45 | 16  
1562059703638 | 1 | 2019-07-02 09:28:23 | 1 | Item_18 | 15  
1562059703804 | 2 | 2019-07-02 09:28:23 | 2 | Item_11 | 2  
1562059703826 | 2 | 2019-07-02 09:28:23 | 2 | Item_0 | 6
```

KSQL - REST API

```
$ echo '{"ksql":"SELECT ORDERID, ITEMID, ADDRESS FROM ORDERS LIMIT 5;", "streamsProperties":  
      "ksql.streams.auto.offset.reset": "earliest"}}' | \  
http http://localhost:8088/query  
HTTP/1.1 200 OK  
Content-Encoding: gzip  
Content-Type: application/json  
Date: Tue, 02 Jul 2019 12:46:25 GMT  
Server: Jetty(9.4.14.v20181114)  
Transfer-Encoding: chunked  
Vary: Accept-Encoding, User-Agent  
  
{  
  "row": {  
    "columns": [0, "Item_0", {"STREET": "377 Maryland Place", "CITY": "Beaumont", "STATE": "Texas"},  
    "terminal": false  
  },  
  "row": {  
    "columns": [0, "Item_0", {"STREET": "072 Butternut Lane", "CITY": "Grand Junction", "STATE": "Colorado",  
    "message": null, "terminal": false  
  },  
  "row": {  
    "columns": [1, "Item_0", {"STREET": "703 Hoffman Place", "CITY": "Mountain View", "STATE": "California",  
    "message": null, "terminal": false  
  },  
  "row": {  
    "columns": [2, "Item_0", {"STREET": "0 Dorton Circle", "CITY": "Brooklyn", "STATE": "New York"},  
    "message": null, "terminal": false  
  },  
  "row": {  
    "columns": [3, "Item_0", {"STREET": "404 Mayer Park", "CITY": "Lubbock", "STATE": "Texas"}],  
    "message": null, "terminal": false  
  },  
  "row": null, "errorMessage": null, "finalMessage": "Limit Reached", "terminal": true  
}
```

KSQL IN ACTION

FILTERING WITH KSQL

 ***ORDERS***

FILTERING WITH KSQL

 **ORDERS**

KSQL ↓

```
CREATE STREAM ORDERS_NY AS
SELECT *
  FROM ORDERS
 WHERE ADDRESS->STATE='New York';
```

FILTERING WITH KSQL

 **ORDERS**

ksql



```
CREATE STREAM ORDERS_NY AS  
SELECT *  
  FROM ORDERS  
 WHERE ADDRESS->STATE='New York';
```



ORDERS_NY

SCHEMA MANIPULATION WITH KSQL



ORDERS

```
{ "ordertime": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5,  
  "address": {  
    "street": "243 Utah Way",  
    "city": "Orange",  
    "state": "California"  
  }  
}
```

SCHEMA MANIPULATION WITH KSQL



ORDERS

ksql



```
CREATE STREAM ORDERS_NO_ADDRESS_DATA AS  
SELECT ORDERTIME, ORDERID, ITEMID, ORDERUNITS  
FROM ORDERS;
```

```
{ "ordertime": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5,  
  "address": {  
    "street": "243 Utah Way",  
    "city": "Orange",  
    "state": "California"  
  }  
}
```

SCHEMA MANIPULATION WITH KSQL



ORDERS

```
{ "ordertime": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5,  
  "address": {  
    "street": "243 Utah Way",  
    "city": "Orange",  
    "state": "California"  
  }  
}
```

ksql



```
CREATE STREAM ORDERS_NO_ADDRESS_DATA AS  
  SELECT TIMESTAMPSTRING(ROWTIME, 'yyyy-MM-dd HH:mm:ss')  
        AS ORDER_TIMESTAMP,  
        ORDERID, ITEMID, ORDERUNITS  
  FROM ORDERS;
```



ORDERS_NO_ADDRESS_DATA

```
{ "order_ts": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5  
}
```

SCHEMA MANIPULATION WITH KONG



```
{  
  "ordertime": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5,  
  "address": {  
    "street": "243 Utah Way",  
    "city": "Orange",  
    "state": "California"  
  }  
}
```

SCHEMA MANIPULATION WITH KSQL



KSQL

```
CREATE STREAM ORDERS_FLAT AS
SELECT [...]
      ADDRESS->STREET AS ADDRESS_STREET,
      ADDRESS->CITY AS ADDRESS_CITY,
      ADDRESS->STATE AS ADDRESS_STATE
FROM ORDERS;
```

```
{
  "ordertime": 1560070133853,
  "orderid": 67,
  "itemid": "Item_9",
  "orderunits": 5,
  "address": {
    "street": "243 Utah Way",
    "city": "Orange",
    "state": "California"
  }
}
```


SCHEMA MANIPULATION WITH KSQL

KSQL

```
CREATE STREAM ORDERS_FLAT AS  
SELECT [...]
```

```
    ADDRESS->STREET AS ADDRESS_STREET,  
    ADDRESS->CITY AS ADDRESS_CITY,  
    ADDRESS->STATE AS ADDRESS_STATE
```

```
FROM ORDERS;
```

ORDERS_FLAT

```
{  
  "ordertime": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5,  
  "address": {  
    "street": "243 Utah Way",  
    "city": "Orange",  
    "state": "California"  
  }  
}
```

```
{  
  "ordertime": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5,  
  "address-street": "243 Utah Way",  
  "address-city": "Orange",  
  "address-state": "California"  
}
```

RESERIALISING DATA WITH KSQL



ORDERS

```
{"ordertime": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5,  
  "address-street": "243 Utah Way",  
  "address-city": "Orange",  
  "address-state": "California"}
```

RESERIALISING DATA WITH KSQL



ORDERS

```
{"ordertime": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5,  
  "address-street": "243 Utah Way",  
  "address-city": "Orange",  
  "address-state": "California"}
```

KSQL



```
CREATE STREAM ORDERS_CSV  
  WITH (VALUE_FORMAT='DELIMITED') AS  
  SELECT * FROM ORDERS_FLAT;
```

RESERIALISING DATA WITH KSQL



ORDERS

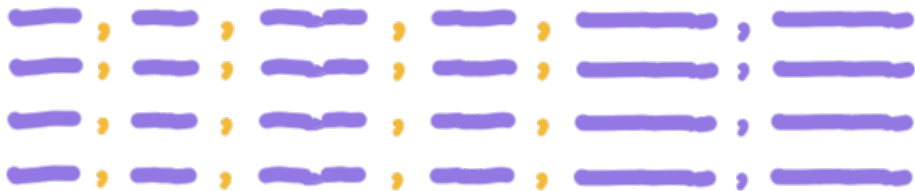
```
{ "ordertime": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5,  
  "address-street": "243 Utah Way",  
  "address-city": "Orange",  
  "address-state": "California" }
```

KSQL



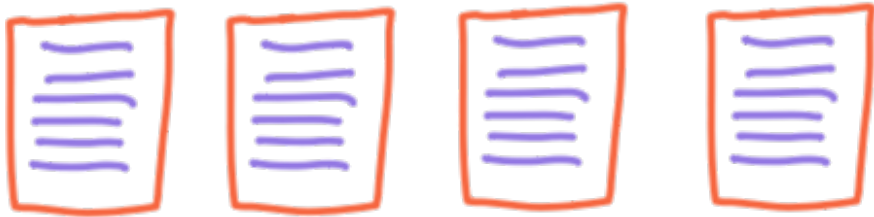
```
CREATE STREAM ORDERS_CSV  
  WITH (VALUE_FORMAT='DELIMITED') AS  
  SELECT * FROM ORDERS_FLAT;
```

ORDERS_CSV



```
1560045914101,24644,Item_0,1,43078 Dexte  
1560047305664,24643,Item_29,3,209 Monter  
1560057079799,24642,Item_38,18,3 Autum  
1560088652051,24647,Item_6,6,82893 Arkan  
1560105559145,24648,Item_0,12,45896 Warr  
1560108336441,24646,Item_33,4,272 Heffer  
1560123862235,24641,Item_15,16,0 Dorton  
1560124799053,24645,Item_12,1,71 Knutson
```

LOOKUPS AND JOINS WITH KSQL



ORDERS

```
{"ordertime": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5}
```

LOOKUPS AND JOINS WITH KSQL

ITEMS

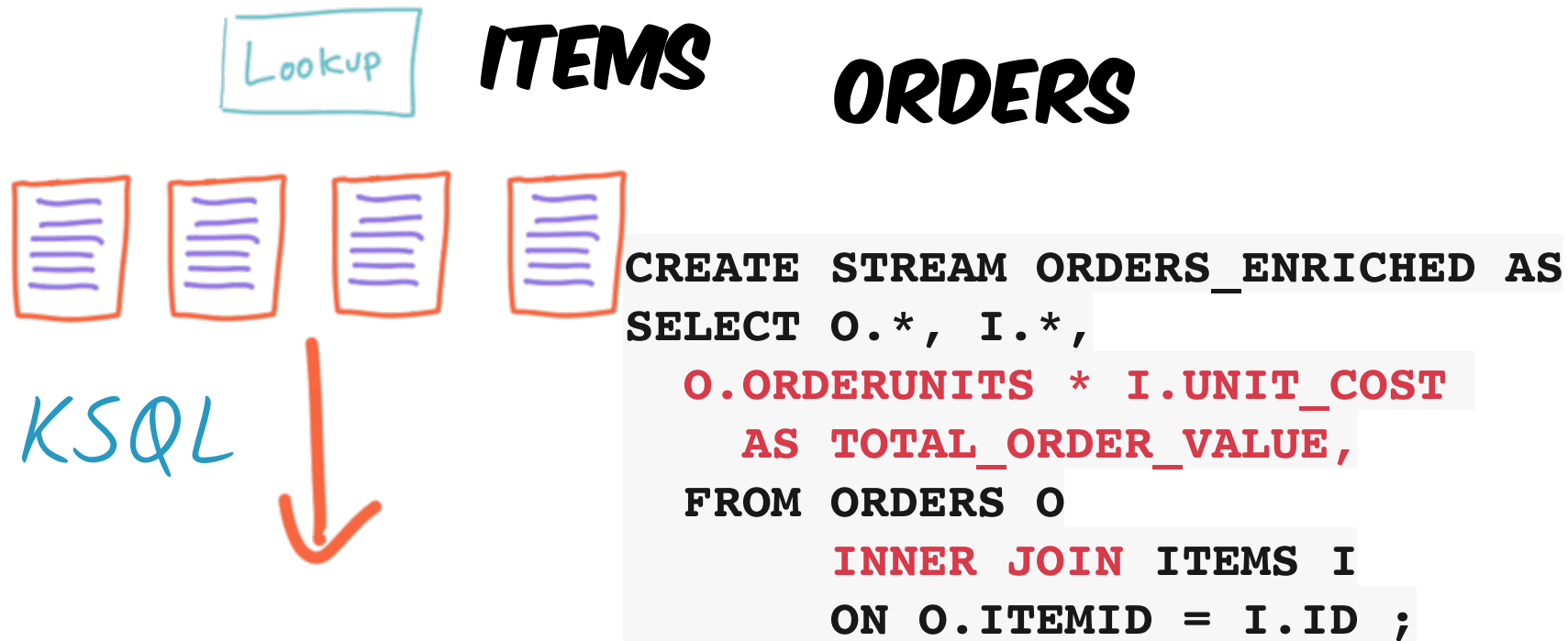
Lookup



ORDERS

```
{  
  "id": "Item_9",  
  "make": "Boyle-McDermott",  
  "model": "Apiaceae",  
  "unit_cost": 19.9  
}  
  
{"ordertime": 1560070133853,  
  "orderid": 67,  
  "itemid": "Item_9",  
  "orderunits": 5}
```

LOOKUPS AND JOINS WITH KSQL



```
{
  "id": "Item_9",
  "make": "Boyle-McDermott",
  "model": "Apiaceae",
  "unit_cost": 19.9
}

{"ordertime": 1560070133853,
 "orderid": 67,
 "itemid": "Item_9",
 "orderunits": 5}
```


LOOKUPS AND JOINS WITH KSQL

Lookup

ITEMS

ORDERS

```
CREATE STREAM ORDERS_ENRICHED AS
SELECT O.*, I.*,
       O.ORDERUNITS * I.UNIT_COST
       AS TOTAL_ORDER_VALUE,
FROM ORDERS O
      INNER JOIN ITEMS I
      ON O.ITEMID = I.ID ;
```

ORDERS_ENRICHED

ksql

```
{
  "id": "Item_9",
  "make": "Boyle-McDermott",
  "model": "Apiaceae",
  "unit_cost": 19.9
}

{"ordertime": 1560070133853,
 "orderid": 67,
 "itemid": "Item_9",
 "orderunits": 5}

{
  "ordertime": 1560070133853,
  "orderid": 67,
  "itemid": "Item_9",
  "orderunits": 5,
  "make": "Boyle-McDermott",
  "model": "Apiaceae",
  "unit_cost": 19.9,
  "total_order_value": 99.5
}
```

CONNECTING TO OTHER SYSTEMS WITH KAFKA CONNECT

Lookup



KSQL



Kafka Connect

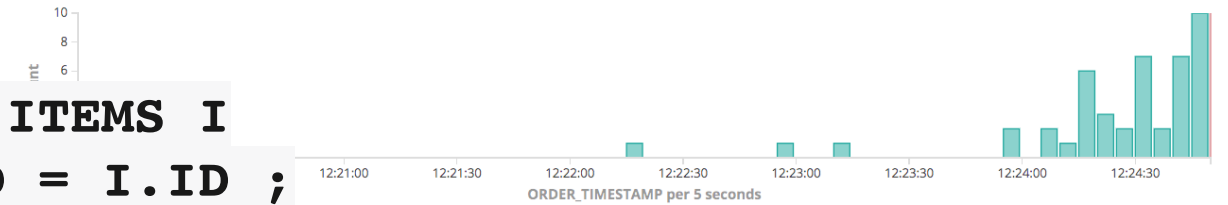


```
CREATE STREAM ORDERS_ENRICHED AS
SELECT [...]
FROM ORDERS O
INNER JOIN ITEMS I
ON O.ITEMID = I.ID ;
```

New Save Open Share Inspect Auto-refresh Last 5m Options Refresh

extension:PHP)

July 2nd 2019, 12:19:48.935 - July 2nd 2019, 12:24:48.935 — Auto



Available fields
t ADDRESS.STATE
Top 5 values in 45 / 45 records
Texas 35.6%
California 35.6%
New York 17.8%
Colorado 11.1%
Visualize
t ADDRESS.STREET
t COLOUR
t ITEMID

Time	ADDRESS.CITY	ORDERUNITS	TOTAL_ORDER_VALUE	UNIT_COST
July 2nd 2019, 12:24:48.440	Fresno	9	170.82	18.98
July 2nd 2019, 12:24:47.962	Houston	1	18.02	18.02
July 2nd 2019, 12:24:47.908	Glendale	19	211.66	11.14
July 2nd 2019, 12:24:47.888	Buffalo	7	33.32	4.76
July 2nd 2019, 12:24:47.356	Oakland	6	103.02	17.17
July 2nd 2019, 12:24:47.292	Stockton	16	280.96	17.56
July 2nd 2019, 12:24:47.164	Houston	19	282.53	14.87
July 2nd 2019, 12:24:45.214	Santa Cruz	7	18.08	0.40

STATEFUL AGGREGATION WITH KSQL



ORDERS

STATEFUL AGGREGATION WITH KSQL



```
SELECT MAKE, COUNT(*) AS ORDER_COUNT  
FROM ORDERS_ENRICHED  
GROUP BY MAKE;
```

STATEFUL AGGREGATION WITH KSQL

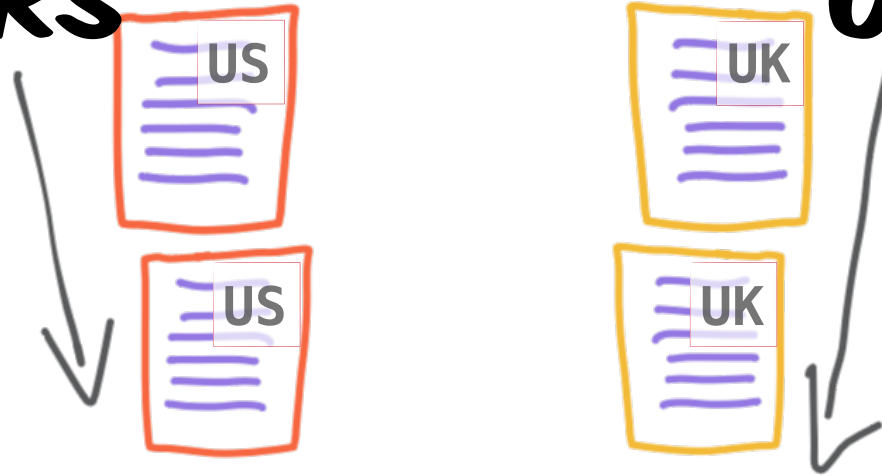


ORDERS

`SELECT MAKE, COUNT(*) AS ORDER_COUNT
FROM ORDERS_ENRICHED
GROUP BY MAKE;`

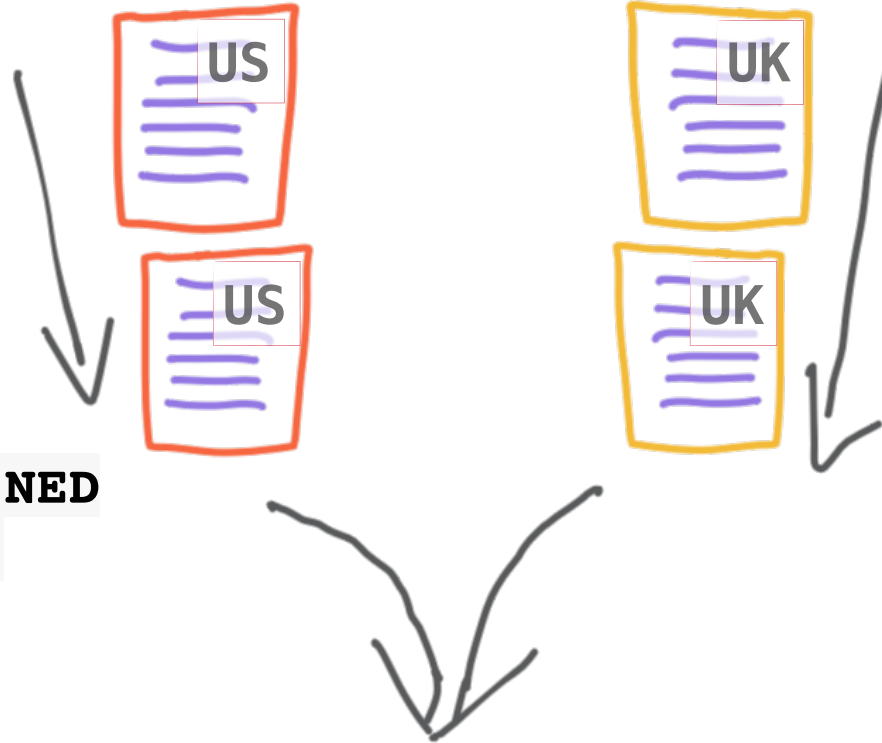
COUNT=4

TRANSFORM DATA WITH KSQL - MERGE STREAMS ORDERS ORDERS_UK



TRANSFORM DATA WITH KSQL - MERGE STREAMS

ORDERS



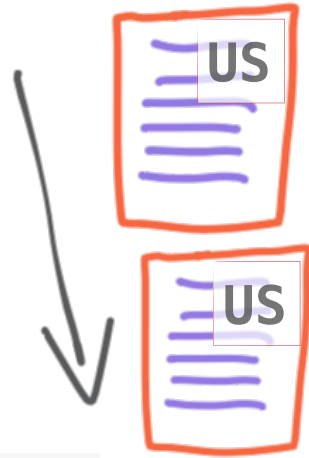
ORDERS_UK

```
INSERT INTO ORDERS_COMBINED  
SELECT 'US' AS SOURCE,  
ORDERTIME,  
ITEMID,  
ORDERUNITS,  
ADDRESS  
FROM ORDERS;
```

```
INSERT INTO ORDERS_COMBINED  
SELECT 'UK' AS SOURCE,  
ORDERTIME,  
ITEMID,  
ORDERUNITS,  
ADDRESS  
FROM ORDERS_UK;
```

TRANSFORM DATA WITH KSQL - MERGE STREAMS

ORDERS



ORDERS_UK



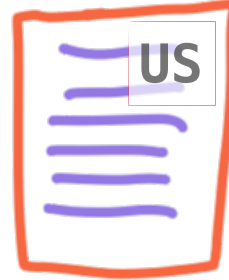
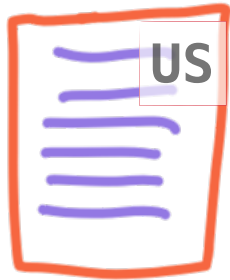
```
INSERT INTO ORDERS_COMBINED
SELECT 'US' AS SOURCE,
ORDERTIME,
ITEMID,
ORDERUNITS,
ADDRESS
FROM ORDERS;
```

```
INSERT INTO ORDERS_COMBINED
SELECT 'UK' AS SOURCE,
ORDERTIME,
ITEMID,
ORDERUNITS,
ADDRESS
FROM ORDERS_UK;
```



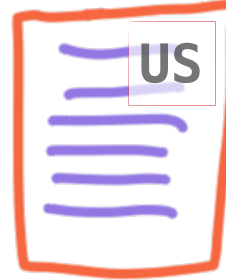
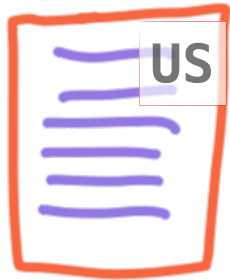
ORDERS_COMBINED

TRANSFORM DATA WITH KSQL - MERGE STREAMS



ORDERS_COMBINED

TRANSFORM DATA WITH KSQL - MERGE STREAMS



ORDERS_COMBINED

```
CREATE STREAM ORDERS_US AS  
  SELECT *  
    FROM ORDERS_COMBINED  
   WHERE SOURCE = 'US' ;
```



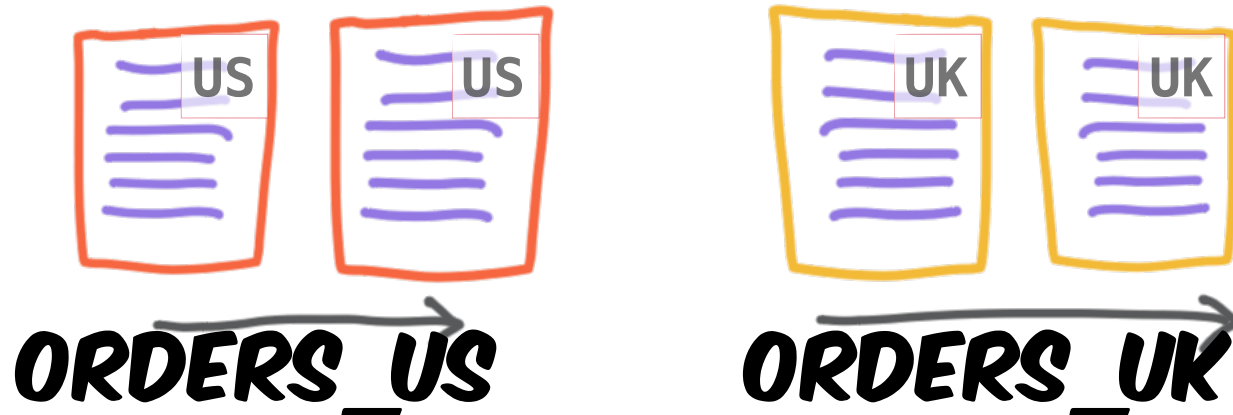
```
CREATE STREAM ORDERS_UK AS  
  SELECT *  
    FROM ORDERS_COMBINED  
   WHERE SOURCE = 'UK' ;
```

TRANSFORM DATA WITH KSQL - MERGE STREAMS



```
CREATE STREAM ORDERS_US AS  
SELECT *  
FROM ORDERS_COMBINED  
WHERE SOURCE = 'US';
```

```
CREATE STREAM ORDERS_UK AS  
SELECT *  
FROM ORDERS_COMBINED  
WHERE SOURCE = 'UK';
```



DO KSQL ON KUBERNETES DEMO



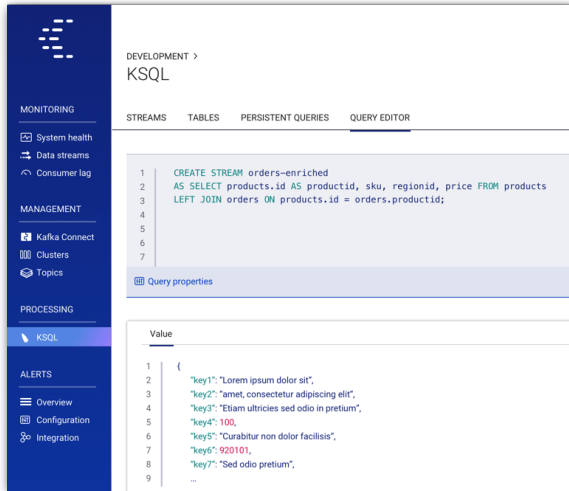
AND EVERYONE LOOSES THEIR MINDS

KSQL OPERATIONS AND DEPLOYMENT

KSQL IN DEVELOPMENT AND PRODUCTION

Interactive KSQL
for development and testing

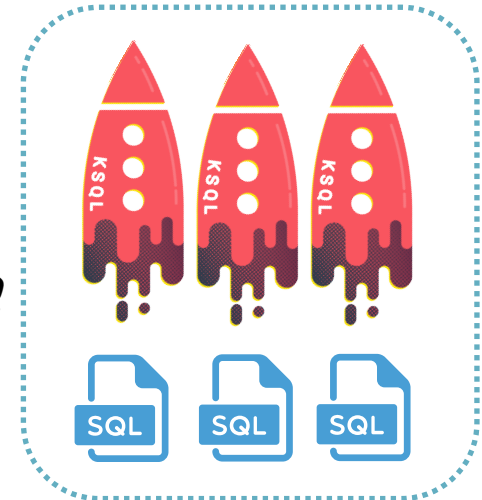
Headless KSQL
for Production



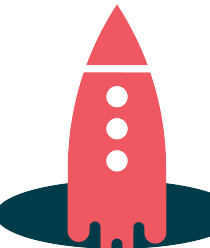
REST



**DESIRED KSQL QUERIES
HAVE BEEN IDENTIFIED**



**“HMM, LET ME TRY
OUT THIS IDEA...”**



HOW TO RUN KSQL



KSQL Server
(JVM process)

DEB, RPM, ZIP, TAR downloads
<http://confluent.io/ksql>

Docker images
[confluentinc/cp-ksql-server](#)
[confluentinc/cp-ksql-cli](#)



Physical



docker



kubernetes



openstack®

vmware®



Microsoft
Azure



Google Cloud Platform

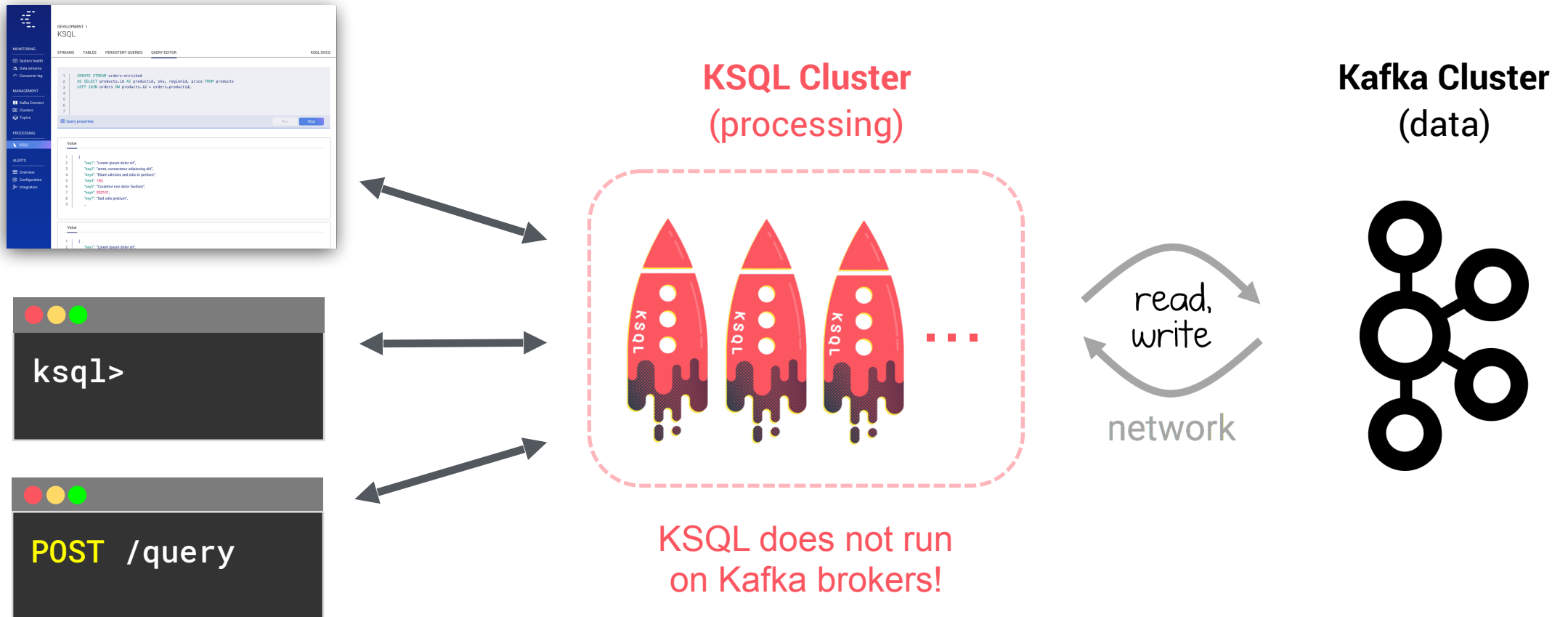


amazon
web services

...and many more...

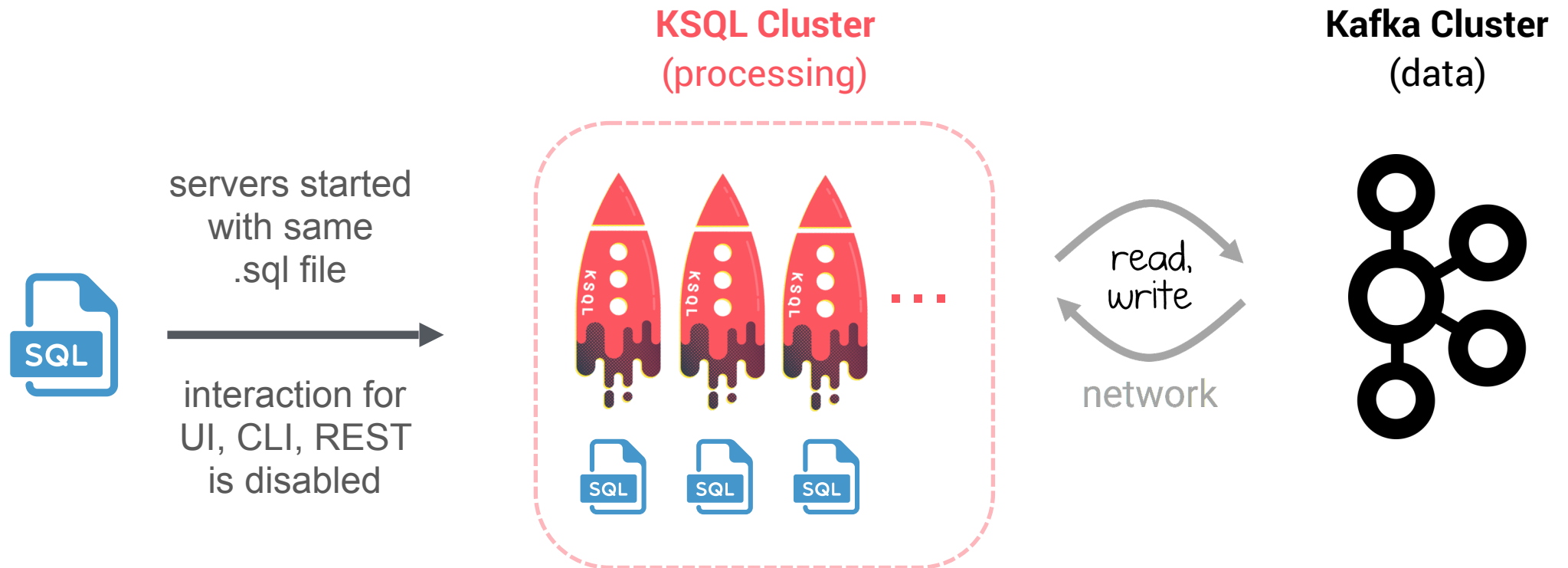
HOW TO RUN KSQL

#1 Interactive KSQL, for development & testing



HOW TO RUN KSQL

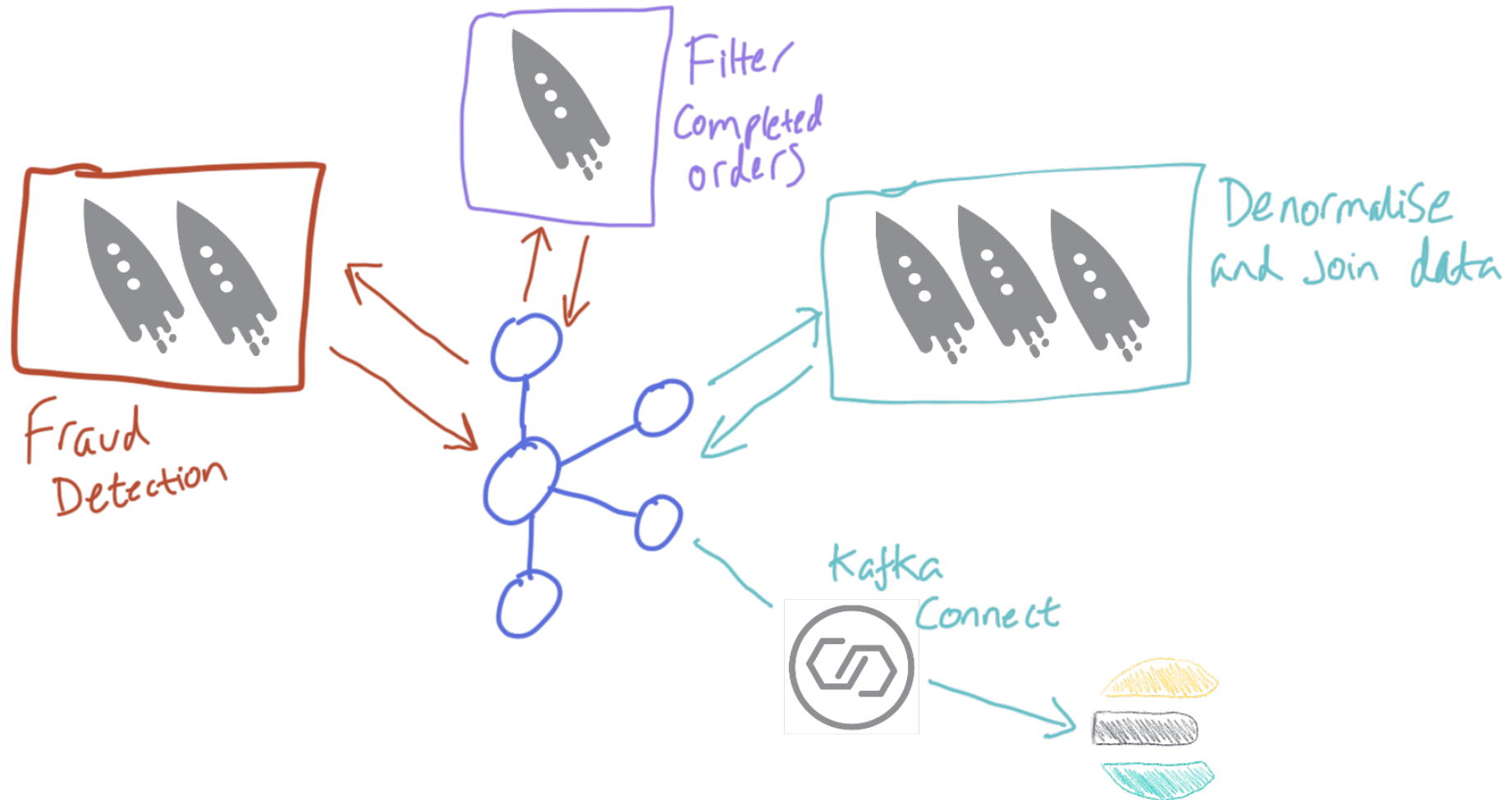
#2 Headless KSQL, for production



ONE DOES NOT SIMPLY RUN

A SINGLE KSQL CLUSTER

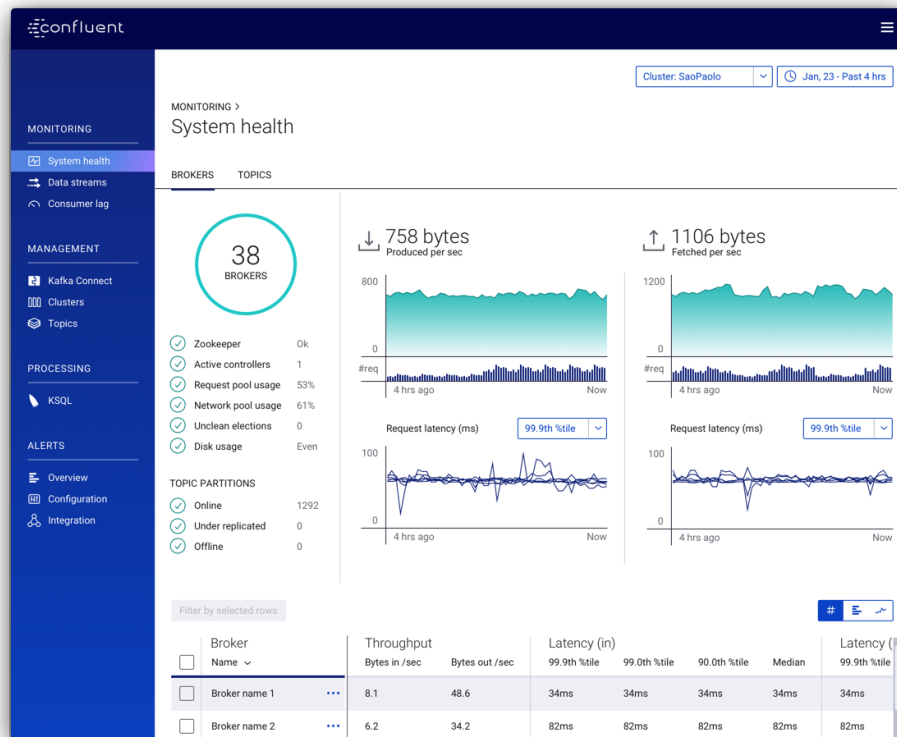
THINK **APPLICATIONS**, NOT DATABASE INSTANCES



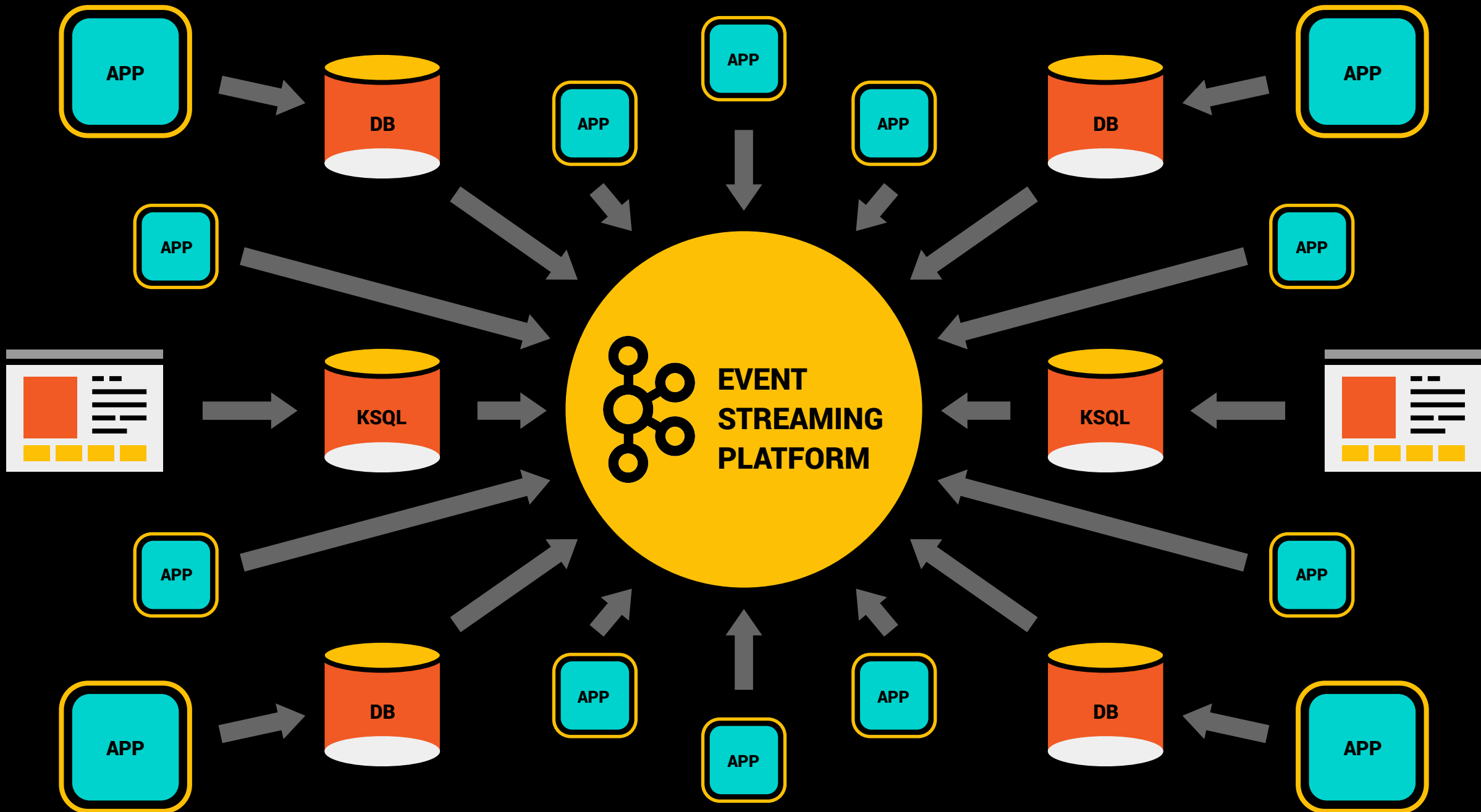
MONITORING KSQL

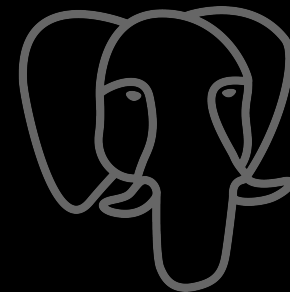
Confluent Control Center

JMX



<https://www.confluent.io/blog/troubleshooting-ksql-part-2>

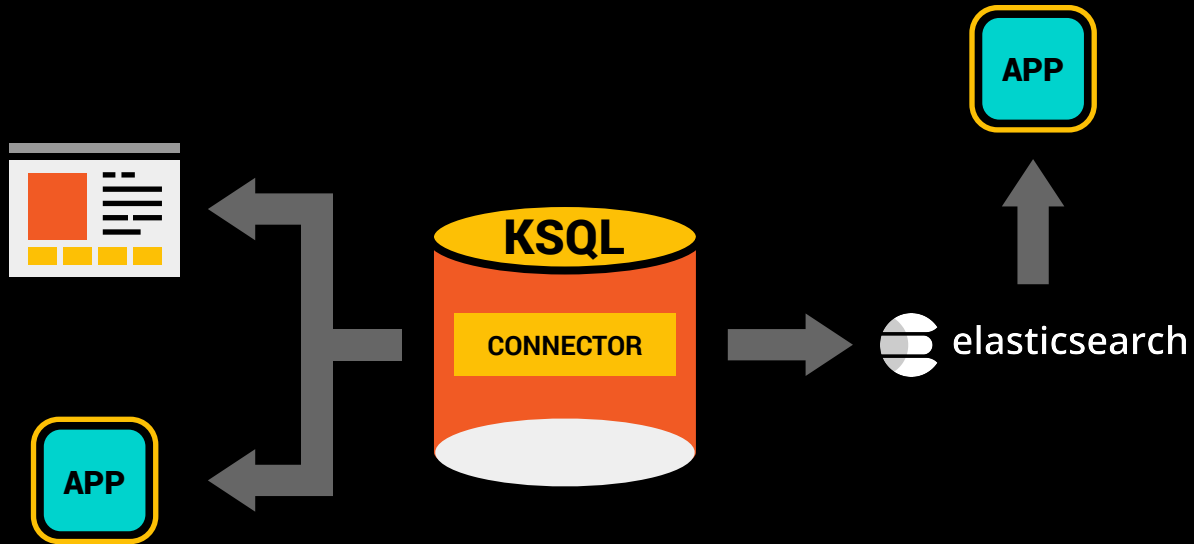




**We still need all
of these!**



In fact this makes it easier to use them...



```
CREATE SINK CONNECTOR elasticConnector WITH (  
  'connector.class' =  
  '...ElasticsearchSinkConnector',  
  'topics' = 'CREDIT_SCORES',  
  'connection.url' = 'http://localhost:9200',  
  'type.name' = 'kafka-connect',  
  ...  
);
```

KLIP 7: cnfl.io/ksql-klip-7
KLIP 8: cnfl.io/ksql-klip-8
github.com/confluentinc/ksql

TARGET FIRST RELEASE:
NOVEMBER



THANKS!

@gamussa
viktor@confluent.io



<https://slackpass.io/confluentcommunity>
#ksql

