

KSQL

Open-source streaming for Apache Kafka

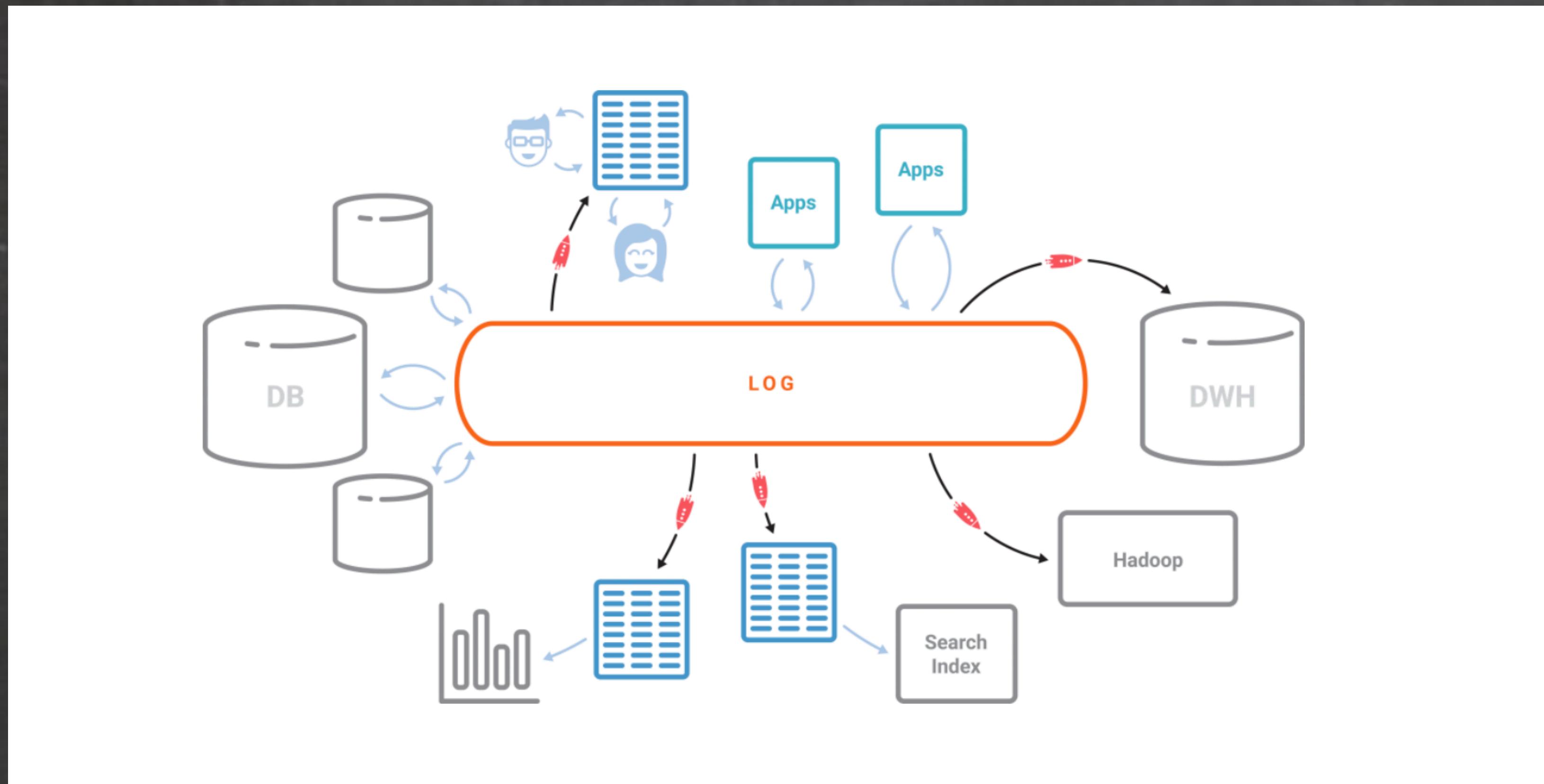
By Vishwa Teja Vangari



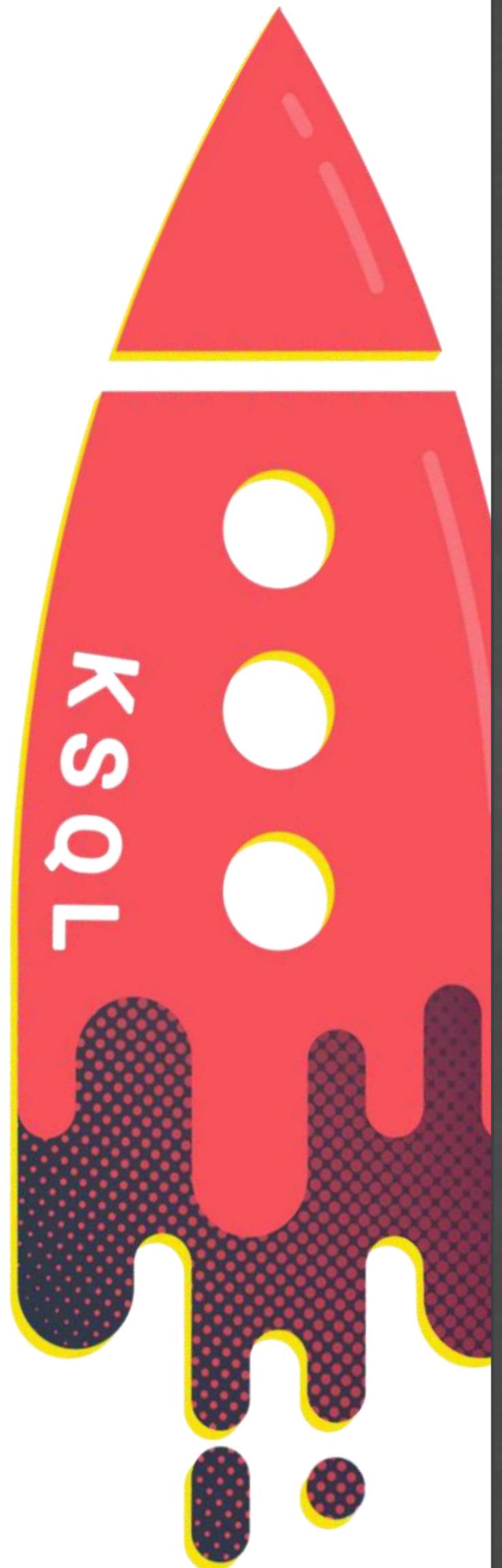
Agenda:

- KSQL
- Benefits
- KSQL Syntaxes & Semantics
- KSQL Examples
- DEMO

Recap of Apache KAFKA_{OG}



KSQL
is the
Streaming
SQL Engine
for
Apache Kafka



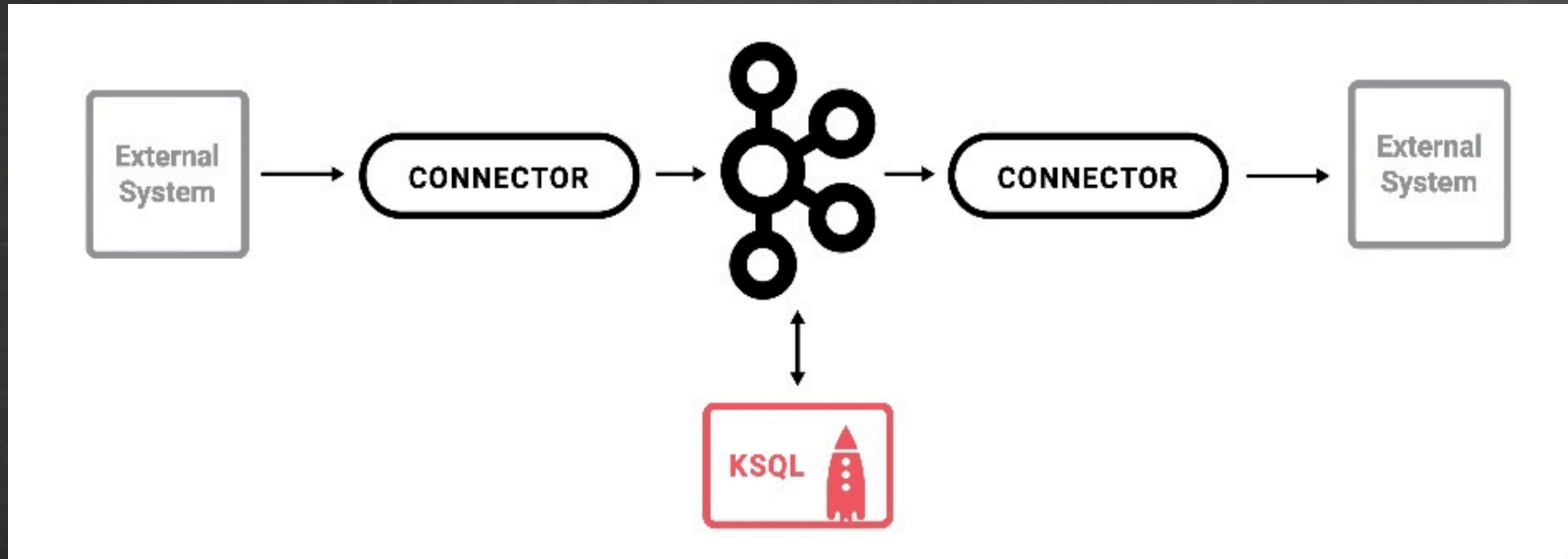
KSQL

is a

Declarative Stream Processing Language



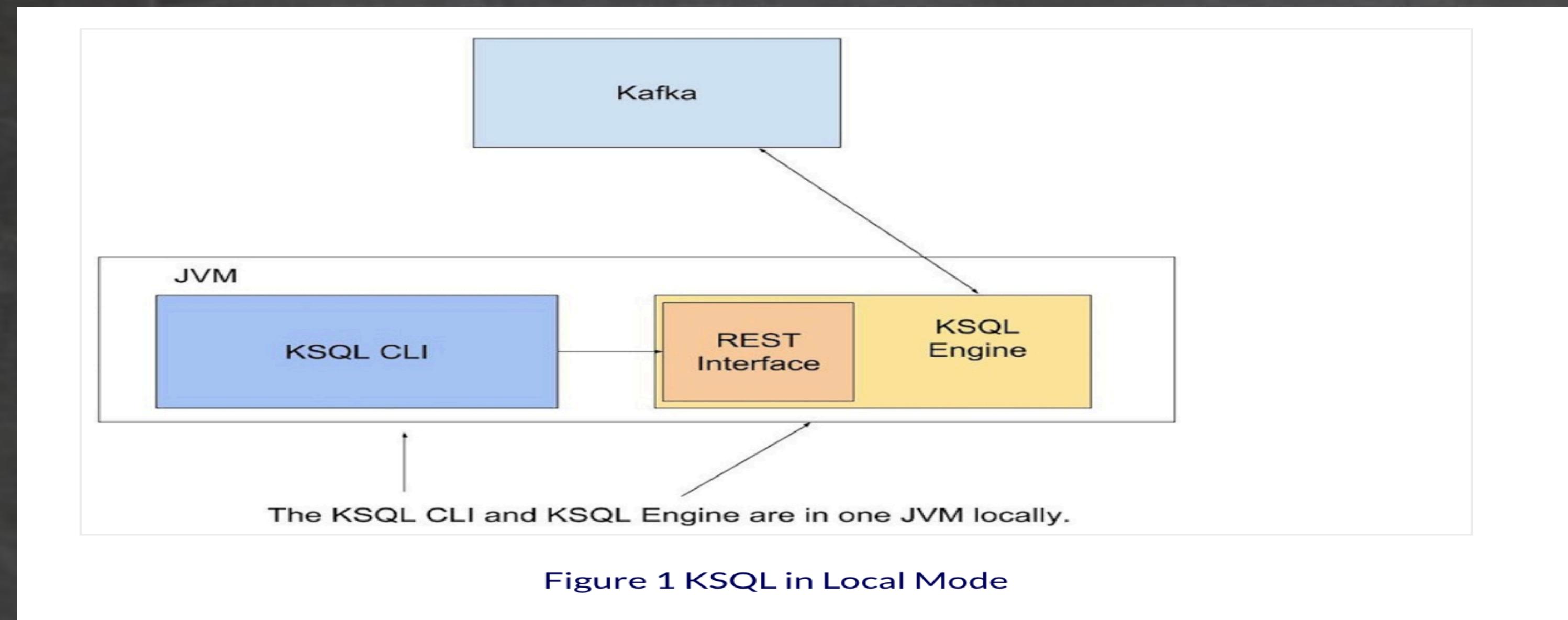
Where it is used?



Why it is used?

- Real-time monitoring meets real-time analytics
- Security and anomaly detection
- Online data integration
- Application Development

KSQL CLI and KSQL Server



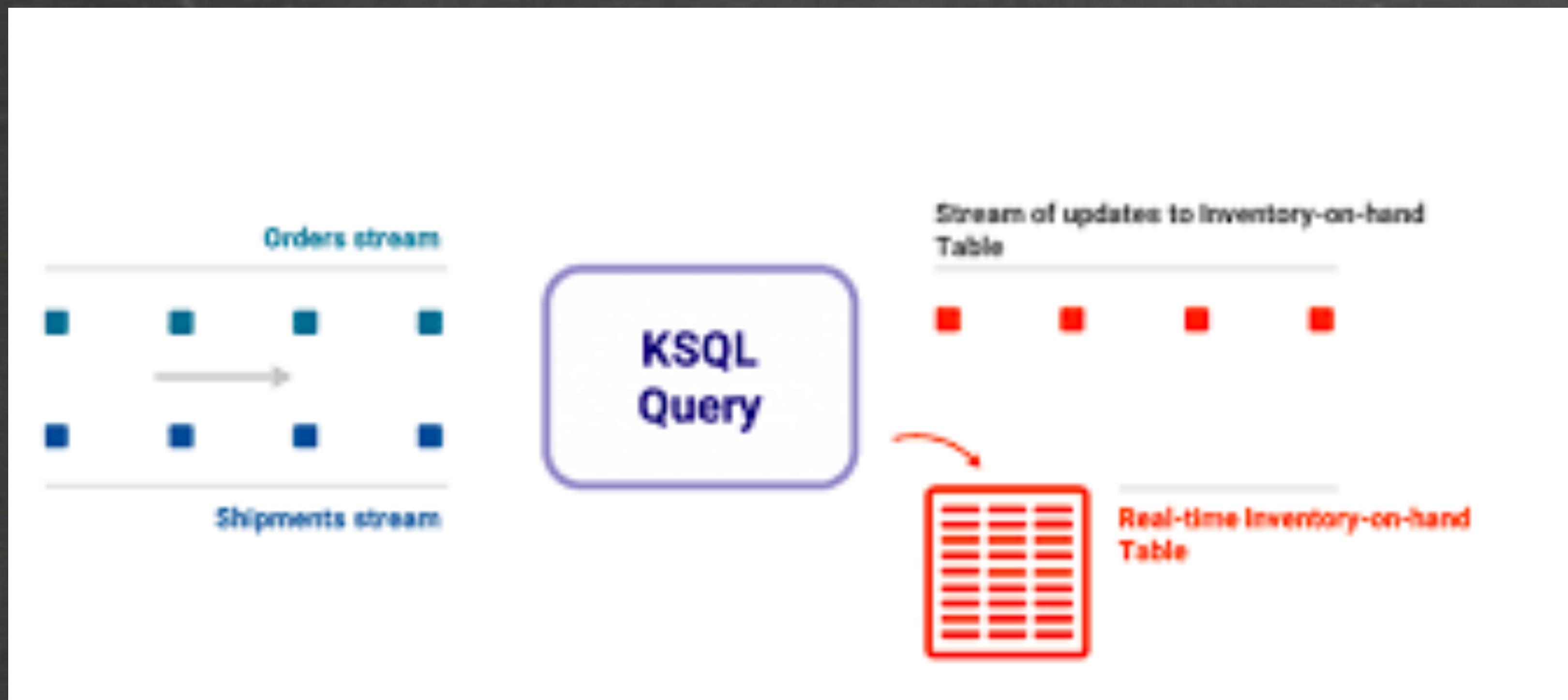
KSQL Statements

- CREATE STREAM
- CREATE TABLE
- CREATE STREAM AS SELECT
- CREATE TABLE AS SELECT
- DESCRIBE
- EXPLAIN
- DROP STREAM
- DROP TABLE
- PRINT
- SELECT
- SHOW TOPICS
- SHOW STREAMS
- SHOW TABLES
- SHOW QUERIES
- SHOW PROPERTIES
- TERMINATE

KSQL SUPPORTED DATA TYPES

- BOOLEAN
- INTEGER
- BIGINT
- DOUBLE
- VARCHAR (or STRING)
- ARRAY<ArrayType> (JSON and AVRO only)
- MAP<VARCHAR, ValueType> (JSON and AVRO only)

Stream and Table:



Creating a stream

> Stream with no key

```
CREATE STREAM pageviews \
(viewtime BIGINT, \
userid VARCHAR, \
pageid VARCHAR) \
WITH (KAFKA_TOPIC='pageviews-topic', \
VALUE_FORMAT='DELIMITED');
```

Creating a stream

- > Stream with specified key

```
CREATE STREAM pageviews \
  (viewtime BIGINT, \
  userid VARCHAR, \
  pageid VARCHAR) \
WITH (KAFKA_TOPIC='pageviews-topic', \
      VALUE_FORMAT='DELIMITED', \
      KEY='pageid');
```

Creating a stream

> Stream with specified key and timestamp

```
CREATE STREAM pageviews \
  (viewtime BIGINT, \
  userid VARCHAR, \
  pageid VARCHAR) \
WITH (KAFKA_TOPIC='pageviews-topic', \
      VALUE_FORMAT='DELIMITED', \
      KEY='pageid', \
      TIMESTAMP='viewtime');
```

Creating a Table

```
CREATE TABLE users \
  (registertime BIGINT, \
   gender VARCHAR, \
   regionid VARCHAR, \
   userid VARCHAR, \
   interests array<VARCHAR>, \
   contact_info map<VARCHAR, VARCHAR>) \
WITH (KAFKA_TOPIC='users-topic', \
      VALUE_FORMAT='JSON', \
      KEY = 'userid');
```

NOTE: Specifying Key is required in Table creation

Transforming

```
CREATE STREAM pageviews_transformed \
    WITH (TIMESTAMP='viewtime', \
          PARTITIONS=5, \
          VALUE_FORMAT='JSON') AS \
SELECT viewtime, \
      userid, \
      pageid, \
      TIMESTAMPTOSTRING(viewtime, 'yyyy-MM-dd HH:mm:ss.SSS') AS timestamp \
FROM pageviews \
PARTITION BY userid;
```

Joining

```
CREATE STREAM pageviews_enriched AS \
  SELECT pv.viewtime, \
    pv.userid AS userid, \
    pv.pageid, \
    pv.timestring, \
    u.gender, \
    u.regionid, \
    u.interests, \
    u.contact_info \
  FROM pageviews_transformed pv \
  LEFT JOIN users u ON pv.userid = users.userid;
```

Aggregating

```
CREATE TABLE pageviews_per_region AS \
    SELECT regionid, \
        count(*) \
    FROM pageviews_enriched \
    GROUP BY regionid;
```

Windowing

```
CREATE TABLE pageviews_per_region_per_minute AS \
    SELECT regionid, \
        count(*) \
    FROM pageviews_enriched \
    WINDOW TUMBLING (SIZE 1 MINUTE) \
    GROUP BY regionid;
```

Windowing: Hopping and Advance

```
CREATE TABLE pageviews_per_region_per_30secs10secs AS SELECT  
regionid, count(*) FROM pageviews_enriched WINDOW HOPPING  
(SIZE 30 SECONDS, ADVANCE BY 10 SECONDS) WHERE  
UCASE(gender)='FEMALE' AND LCASE (regionid) LIKE '%_6' GROUP BY  
regionid;
```

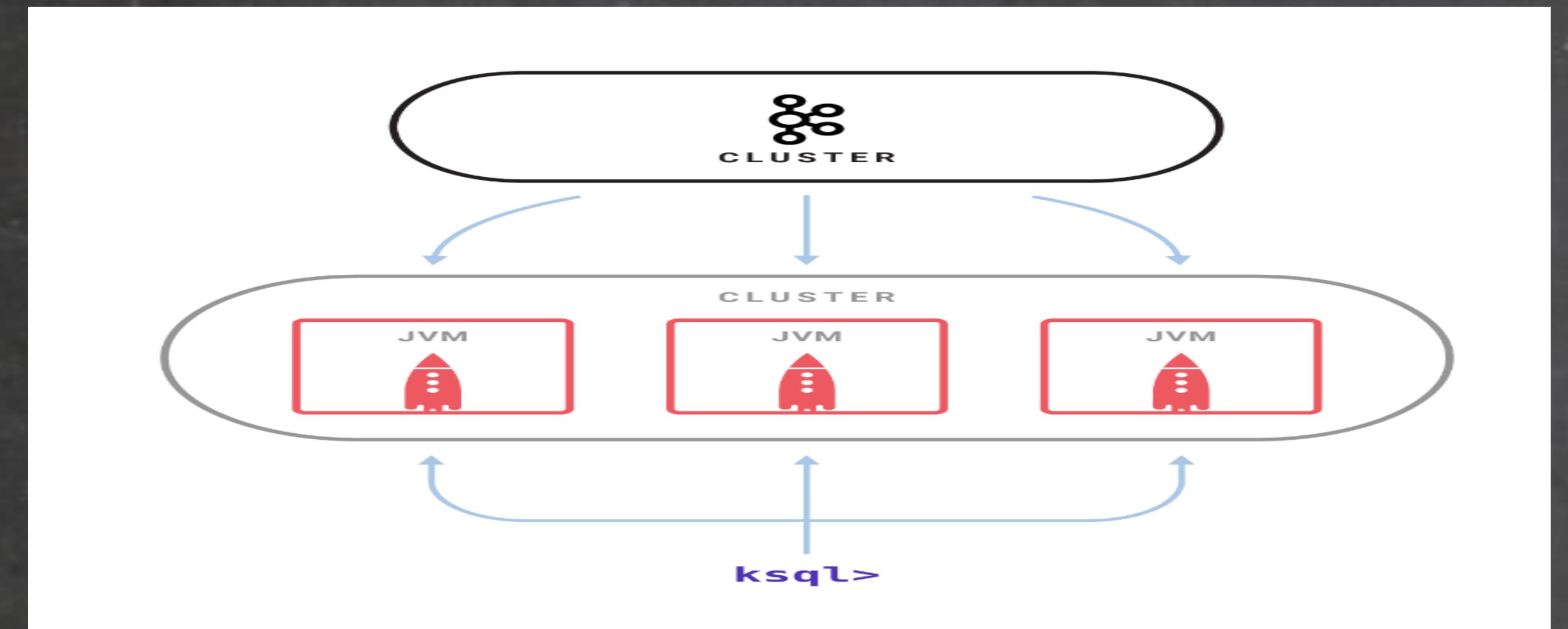
Sessionization

```
CREATE TABLE pageviews_per_region_per_session AS \
    SELECT regionid, \
        count(*) \
    FROM pageviews_enriched \
    WINDOW SESSION (60 SECONDS) \
    GROUP BY regionid;
```

Arrays and maps

```
CREATE STREAM pageviews_interest_contact AS \
    SELECT interests[0] AS first_interest, \
        contact_info['zipcode'] AS zipcode, \
        contact_info['city'] AS city, \
        viewtime, \
        userid, \
        pageid, \
        timestamp, \
        gender, \
        regionid \
    FROM pageviews_enriched;
```

KSQL CLUSTER



KSQLREST API

<https://docs.confluent.io/current/ksql/docs/api.html>

KSQL USE CASE:



References:

- <https://www.confluent.io/product/ksql/>
- <https://docs.confluent.io/current/ksql/docs/tutorials/index.html>

Thanks!

