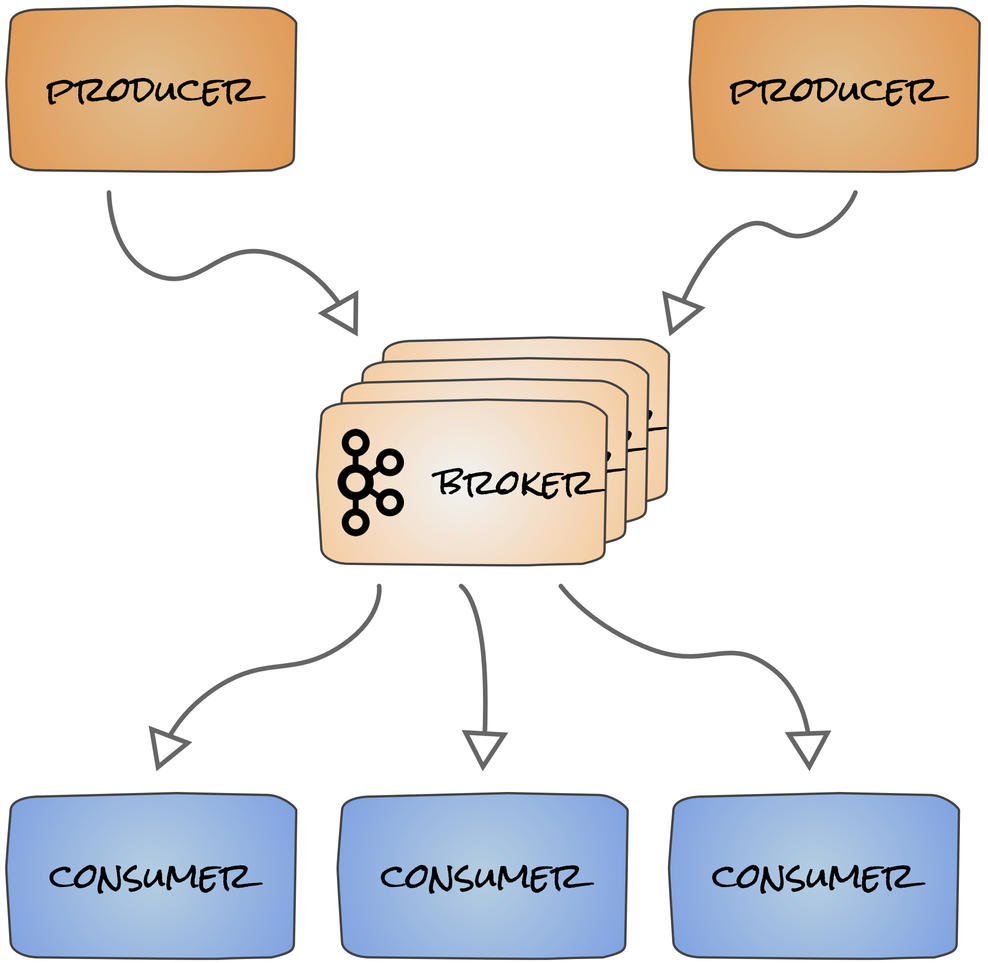
KSQL

Open-source streaming for Apache Kafka

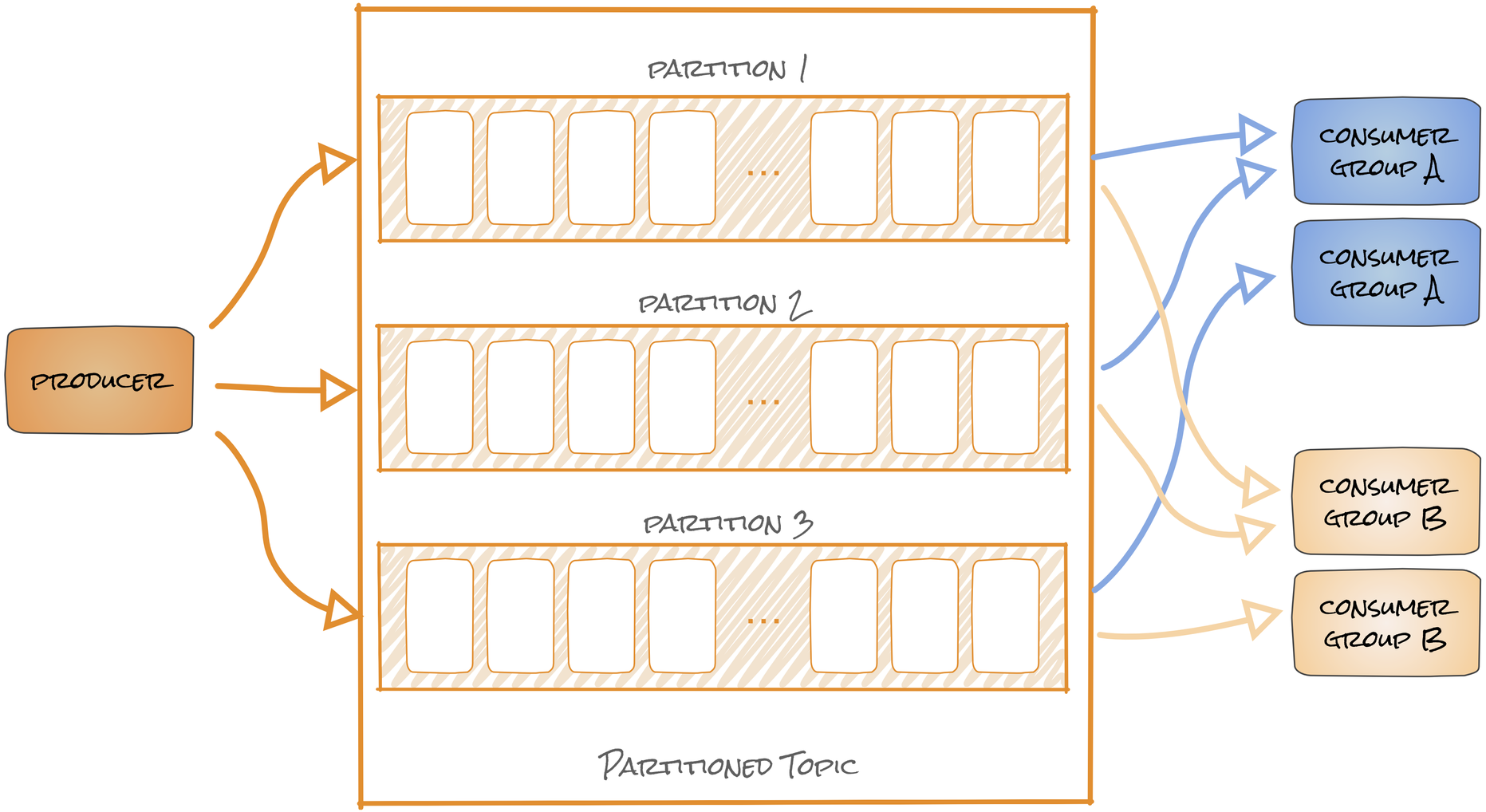
@tlberglund

Kafka Architecture



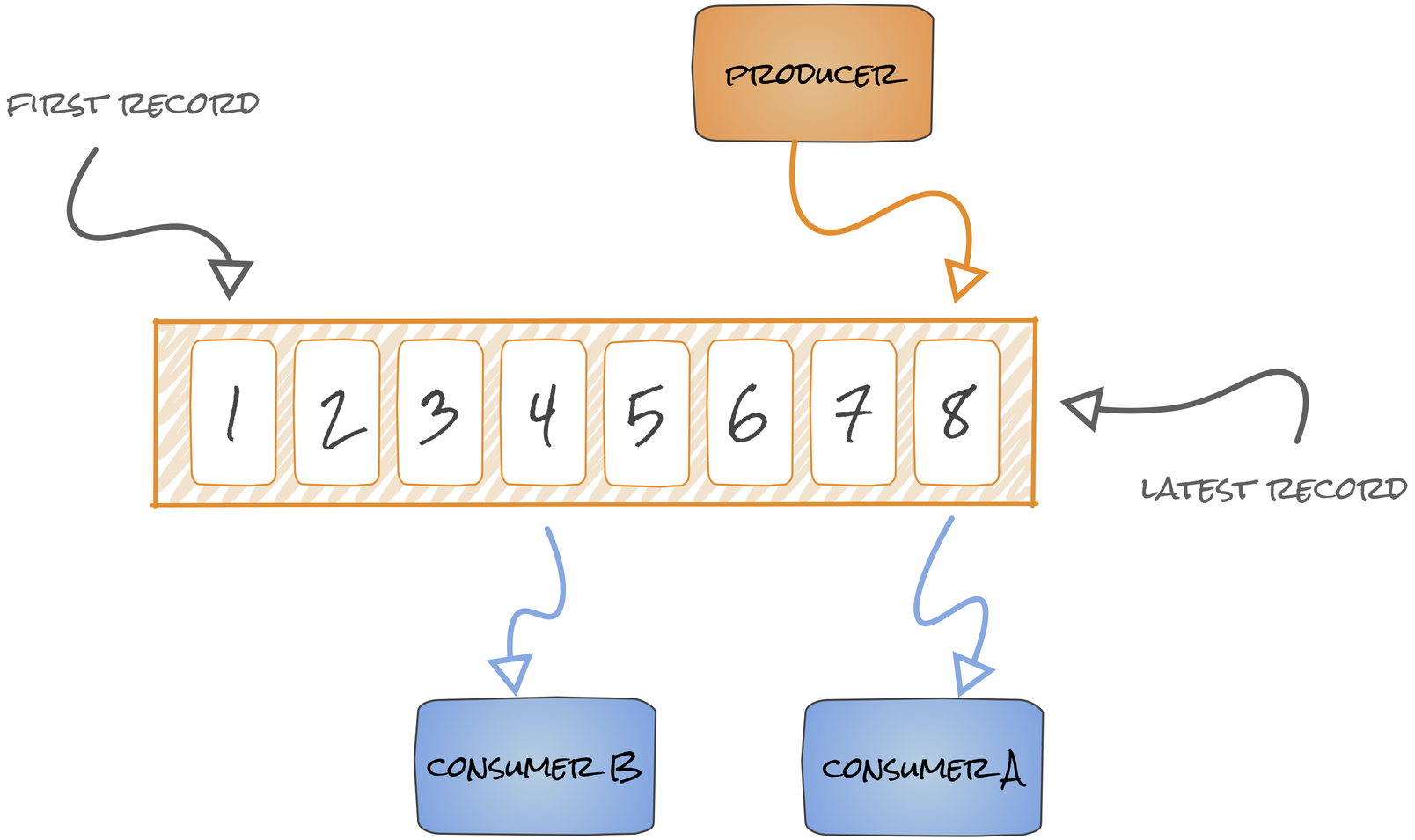
confluent

Scalable Consumption

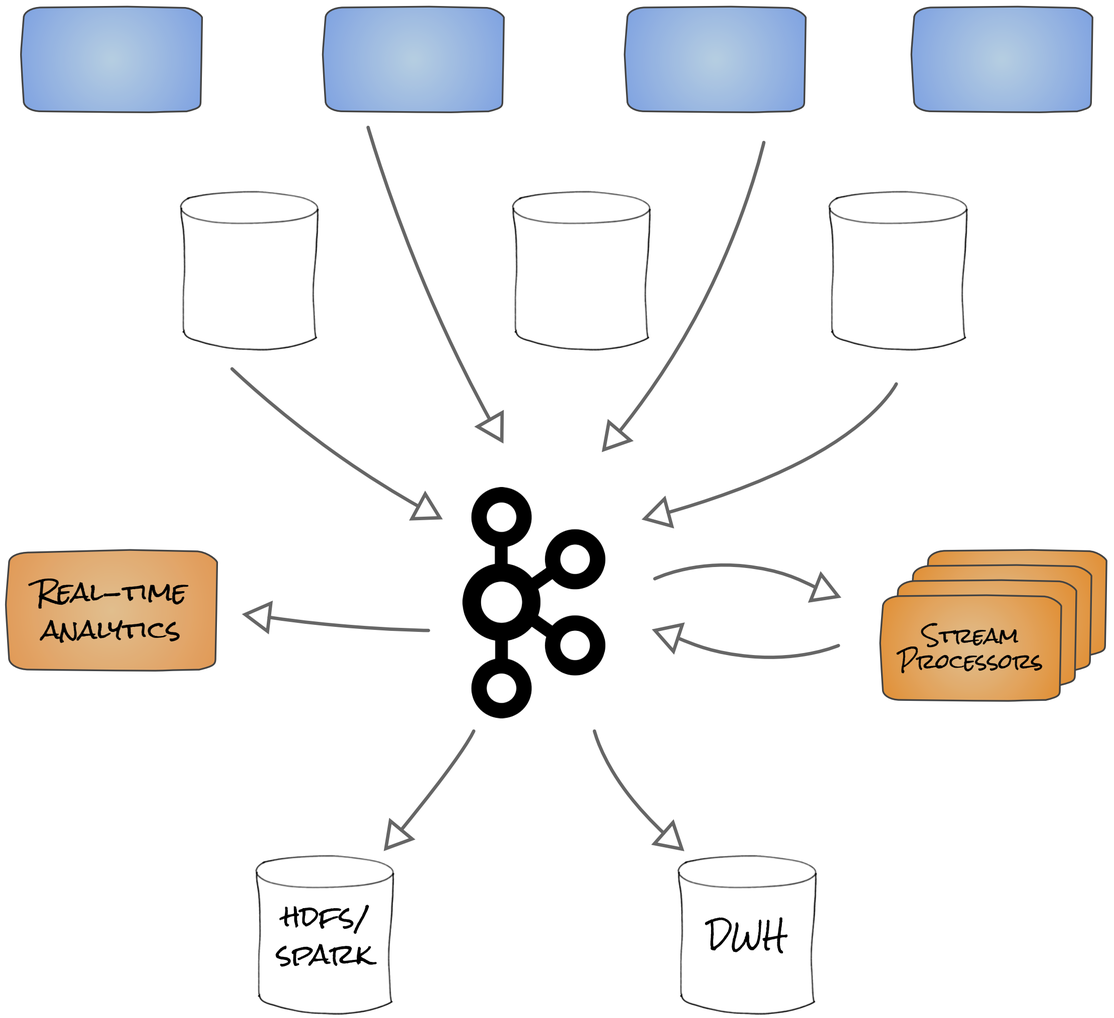


confluent

Logs and Pub/Sub



Streaming Platform



Applications

Databases

Streaming Platform

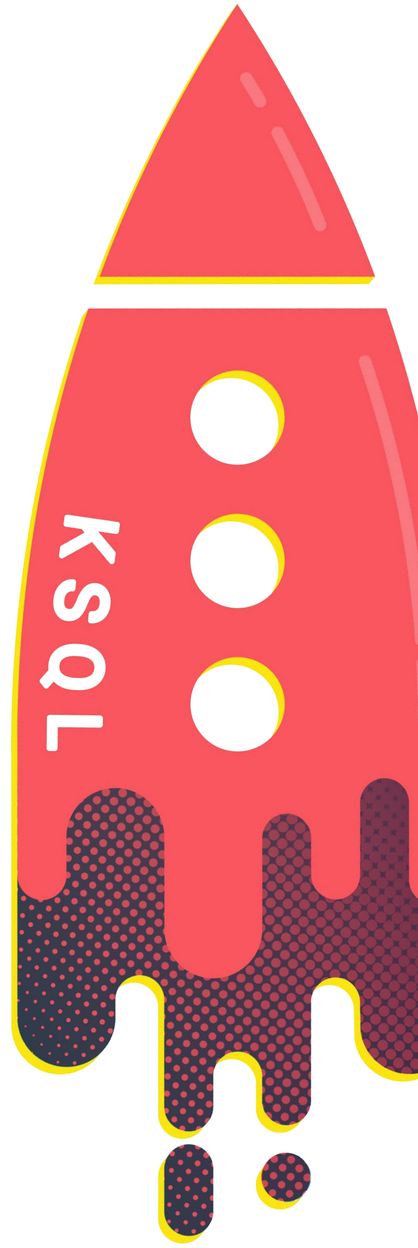
Offline Systems

confluent

KSQL

is a

Declarative Stream Processing Language



confluent

KSQL

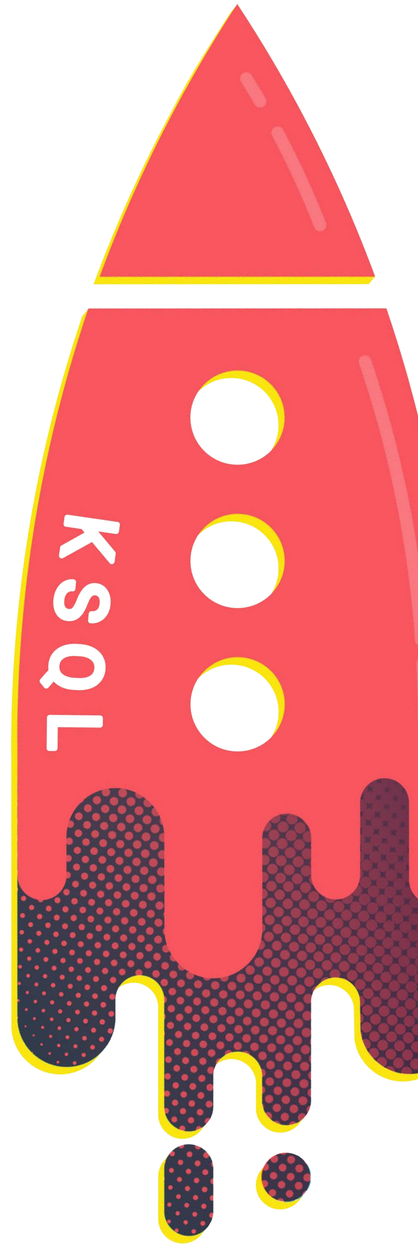
is the

Streaming

SQL Engine

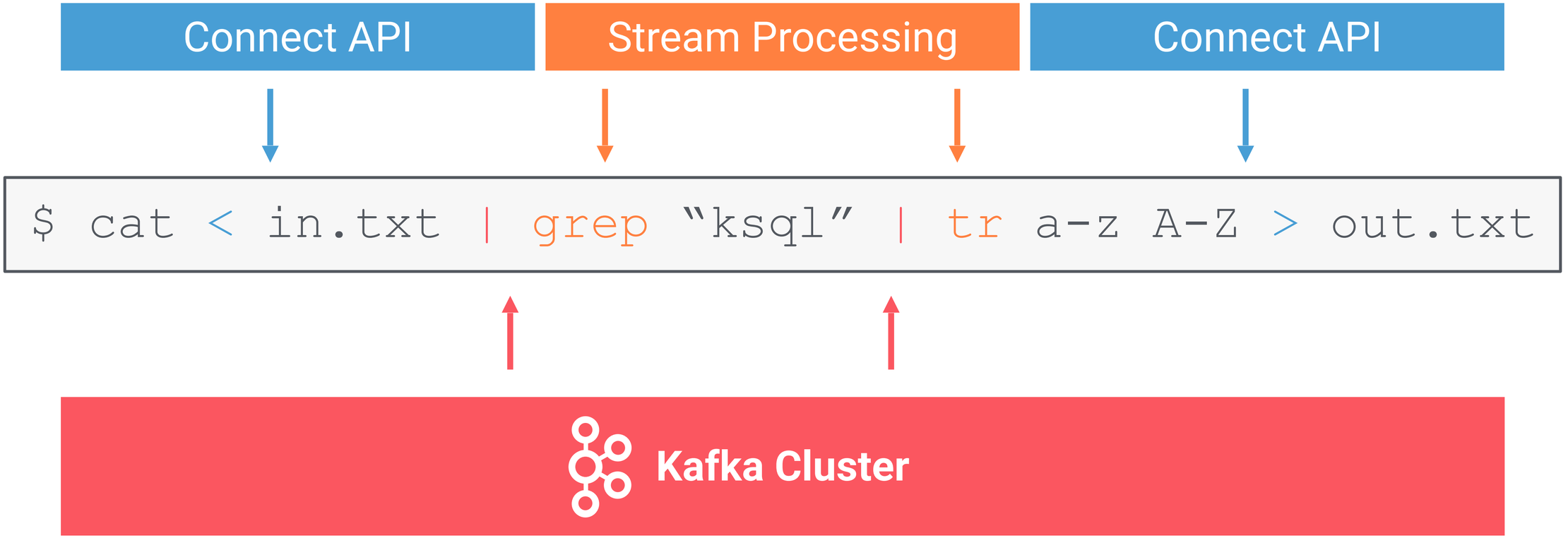
for

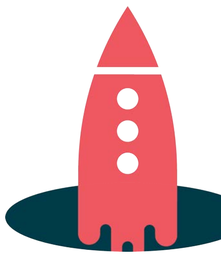
Apache Kafka



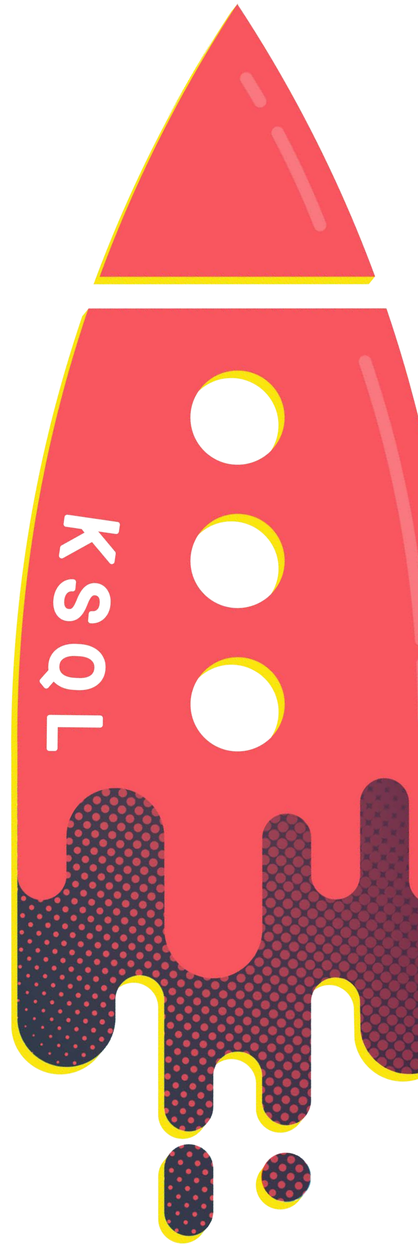
confluent

Stream Processing by Analogy





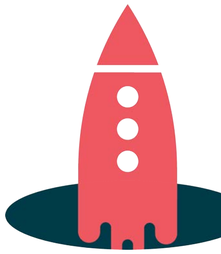
what are some KSQL use cases?



KSQL for Data Exploration

An easy way to inspect data in a running cluster

```
SELECT status, bytes
  FROM clickstream
  WHERE user_agent =
    'Mozilla/5.0 (compatible; MSIE 6.0)';
```

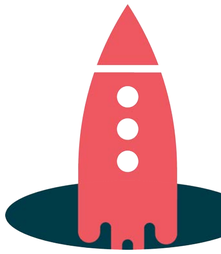


confluent

KSQL for Streaming ETL

- Kafka is popular for data pipelines.
- KSQL enables easy transformations of data within the pipe.
- Transforming data while moving from Kafka to another system.

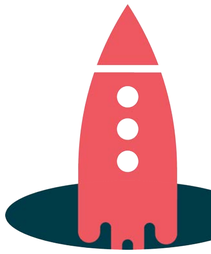
```
CREATE STREAM vip_actions AS
  SELECT userid, page, action FROM clickstream c
    LEFT JOIN users u ON c.userid = u.user_id
    WHERE u.level = 'Platinum';
```



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;
```

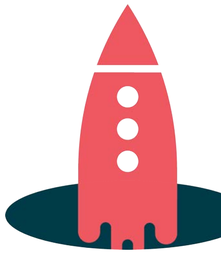


confluent

KSQL for Real-Time Monitoring

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

```
CREATE TABLE error_counts AS
  SELECT error_code, count(*)
    FROM monitoring_stream
    WINDOW TUMBLING (SIZE 1 MINUTE)
    WHERE type = 'ERROR'
    GROUP BY error_code;
```

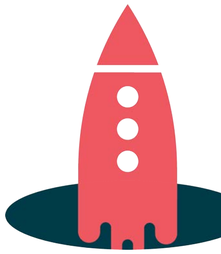


confluent

KSQL for Data Transformation

Make simple derivations of existing topics from the command line

```
CREATE STREAM views_by_userid
  WITH (PARTITIONS=6,
        VALUE_FORMAT='JSON',
        TIMESTAMP='view_time') AS
  SELECT * FROM clickstream PARTITION BY user_id;
```



confluent

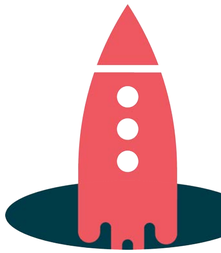
Where is KSQL not such a great fit?

Ad-hoc queries

- Limited span of time usually retained in Kafka
- No indexes

BI reports (Tableau etc.)

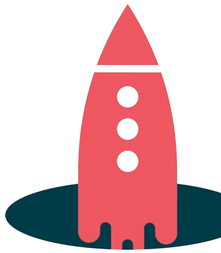
- No indexes
- No JDBC (most BI tools are not good with continuous results!)



Creating a Stream

```
CREATE STREAM clickstream (
  time    BIGINT,
  url     VARCHAR,
  status  INTEGER,
  bytes   INTEGER,
  userid  VARCHAR,
  agent   VARCHAR)
WITH (
  value_format = 'JSON',
  kafka_topic='my_clickstream_topic'
);
```

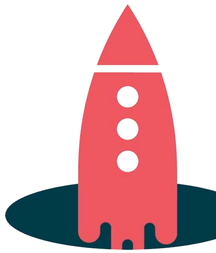
confluent



Creating a Table

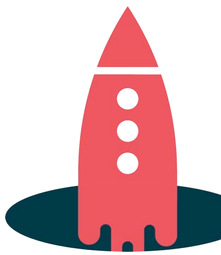
```
CREATE TABLE users (
  user_id         INTEGER,
  registered_at LONG,
  username        VARCHAR,
  name            VARCHAR,
  city            VARCHAR,
  level           VARCHAR)
WITH (
  key = 'user_id',
  kafka_topic='clickstream_users',
  value_format='JSON');
```

confluent



Joins for Enrichment

```
CREATE STREAM vip_actions AS
  SELECT userid, fullname, url, status
  FROM clickstream c
    LEFT JOIN users u ON c.userid = u.user_id
  WHERE u.level = 'Platinum';
```

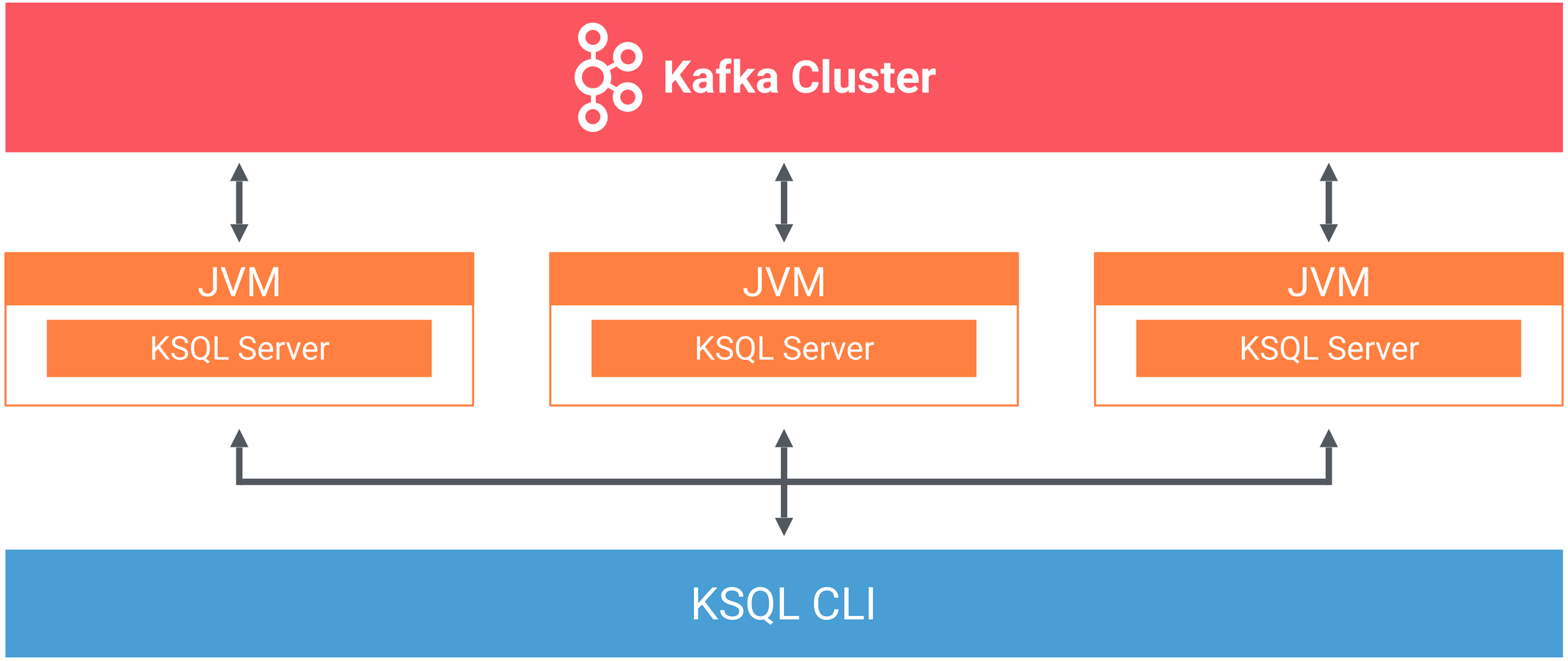


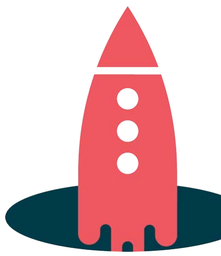
confluent

Demo

How to run KSQL

#2 CLIENT-SERVER





How to run KSQL

#2 CLIENT-SERVER

- **Start any number of server nodes**

```
bin/ksql-server-start
```

- **Start one or more CLIs and point them to a server**

```
bin/ksql-cli remote https://myksqlserver:8090
```

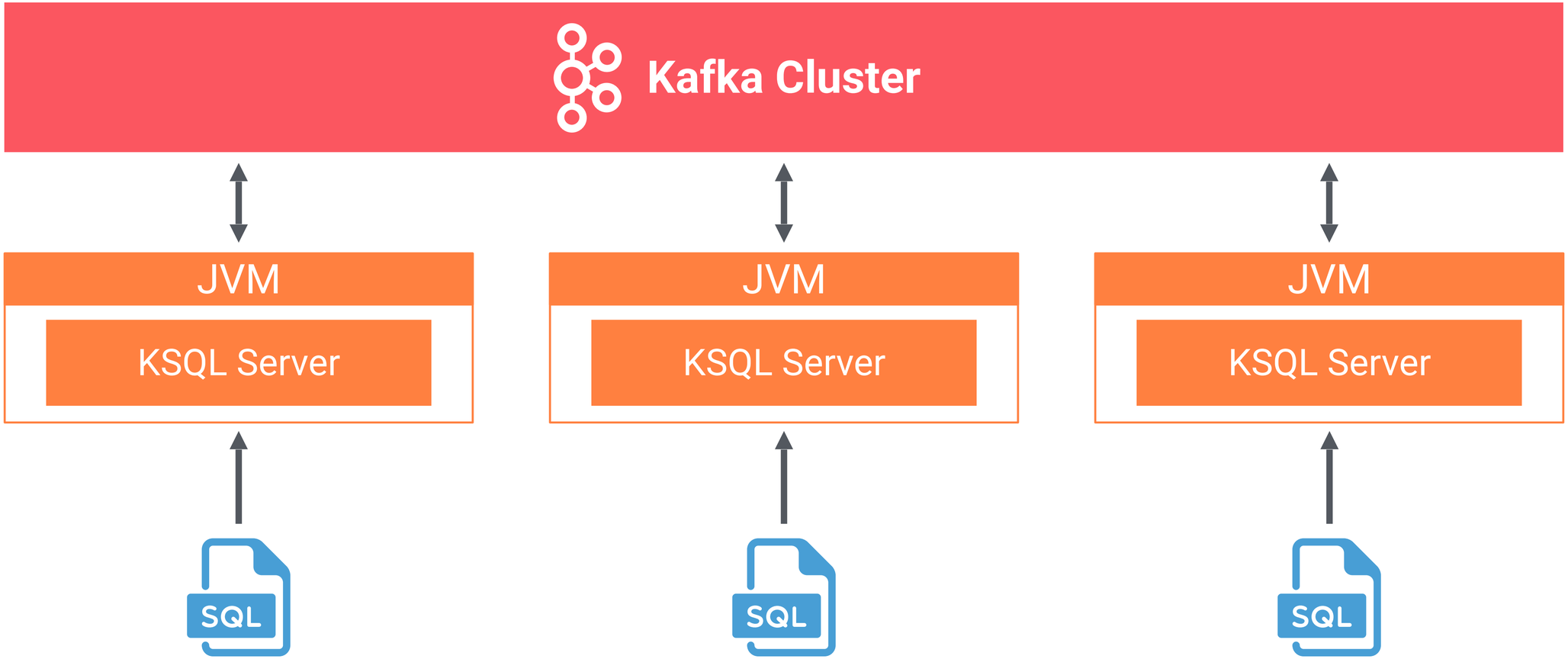
- **All servers share the processing load**

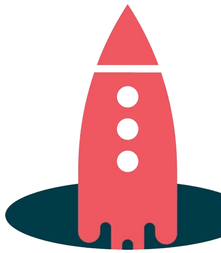
  Technically, instances of the same Kafka Streams Applications

  Scale up/down without restart

How to run KSQL

#3 AS A STANDALONE APPLICATION





How to run KSQL

#3 AS A STANDALONE APPLICATION

- **Start any number of server nodes**
  - **Pass a file of KSQL statement to execute**

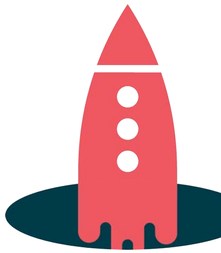
```
bin/ksql-node query-file=foo/bar.sql
```

- **Ideal for streaming ETL application deployment**
  - **Version-control your queries and transformations as code**
- **All running engines share the processing load**
  - **Technically, instances of the same Kafka Streams Applications**
  - **Scale up/down without restart**

Resources and Next Steps

http://confluent.io/ksql

https://slackpass.io/confluentcommunity #ksql



confluent

Thank you!

confluent

@tlberglund

tim@confluent.io