



Crossing the Streams: Rethinking Stream Processing with Kafka Streams and KSQL

**TORONTO KAFKA MEETUP,
NOVEMBER 2018**



Viktor Gamov at #KafkaSummit

@gAmUssA

🙋 Folks, since I'm doing beginner's level talk at [#KafkaSummit](#), I would love to hear what were the hardest things for the beginner to understand about [@kafkstreams](#) and [#streamprocessing](#) so I can address those in my talk **!?**

⚡ RT and reply to this tweet!
[#askconfluent](#)



Confluent ✓ @confluentinc

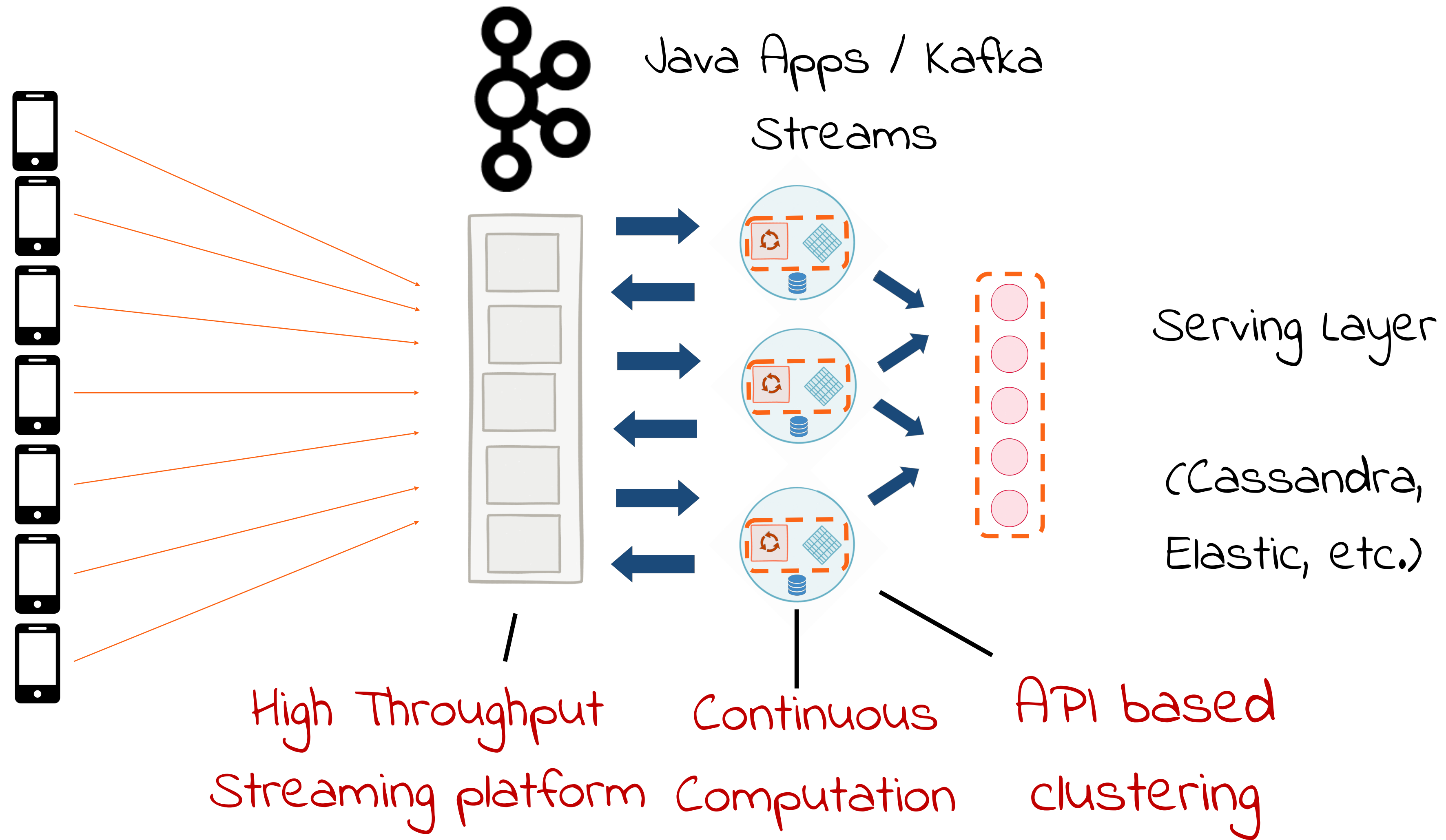
Get on board with streams at #KafkaSummit SF! During his session, @gAmUssA will introduce @apachekafka Streams and #KSQL and discuss how to deploy stream processing apps by examining actual working code....

1:09 PM - 5 Oct 2018

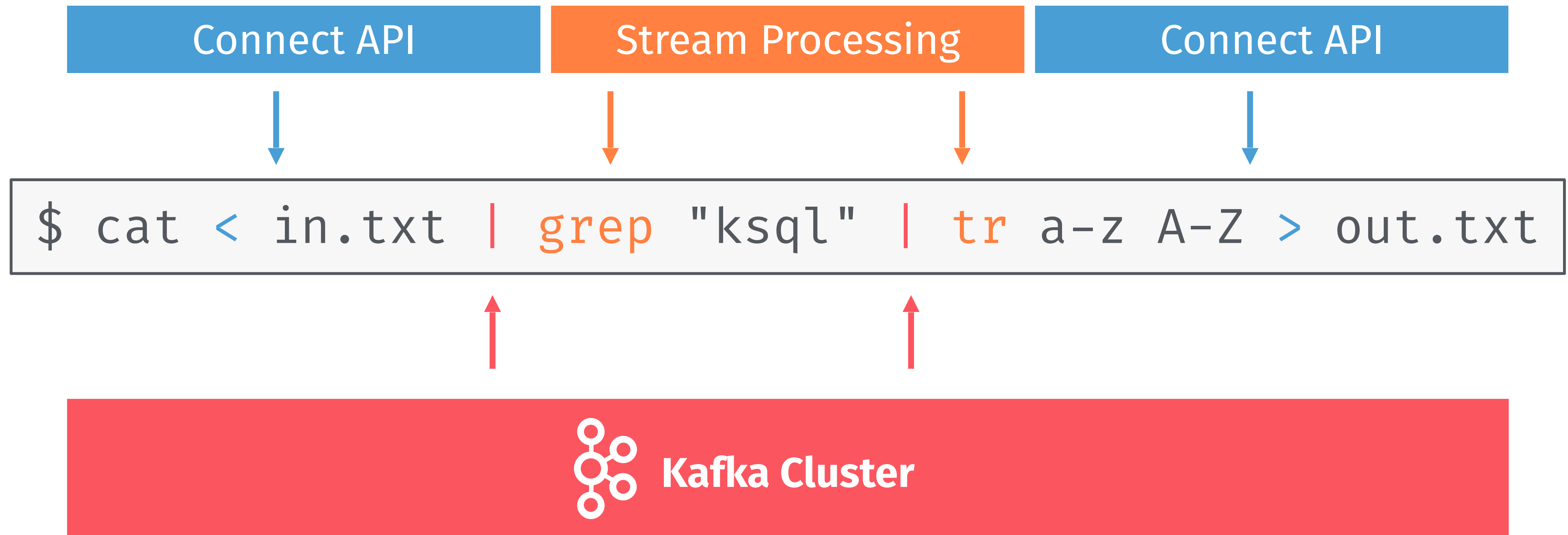
<https://twitter.com/gAmUssA/status/1048258981595111424>

Streaming
is the toolset
for dealing
with events
as they move!

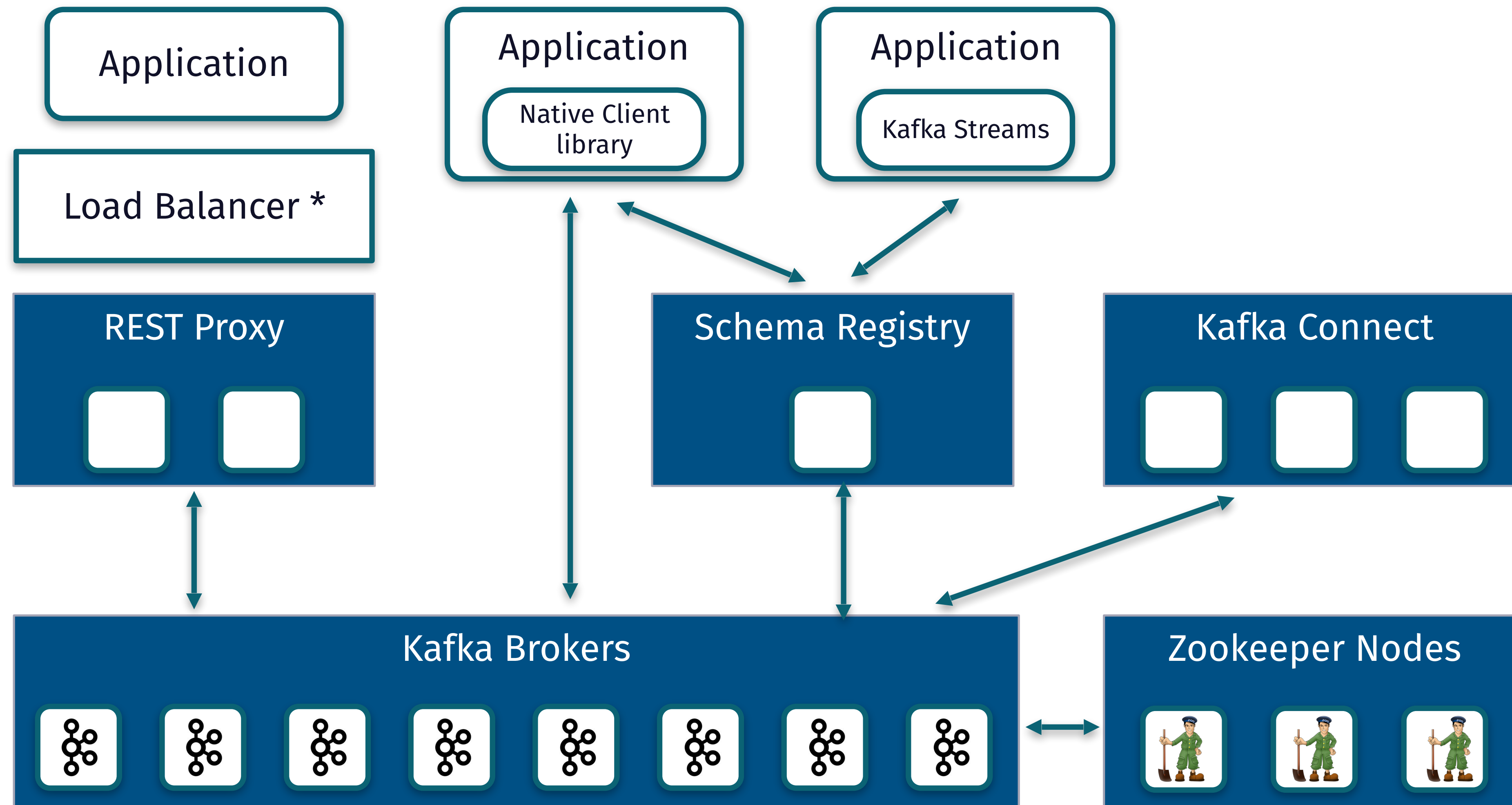




Stream Processing by Analogy



Streaming Platform Architecture





Vladimir Bukhtoyarov

@monitoring_king

Following



Replying to [@gAmUssA](#) [@kafkastreams](#)

The harder thing for me was(is) a motivation to use streams, I do not see reasons to use streams when I am already satisfied with native consumer/producer API.

1:32 PM - 5 Oct 2018

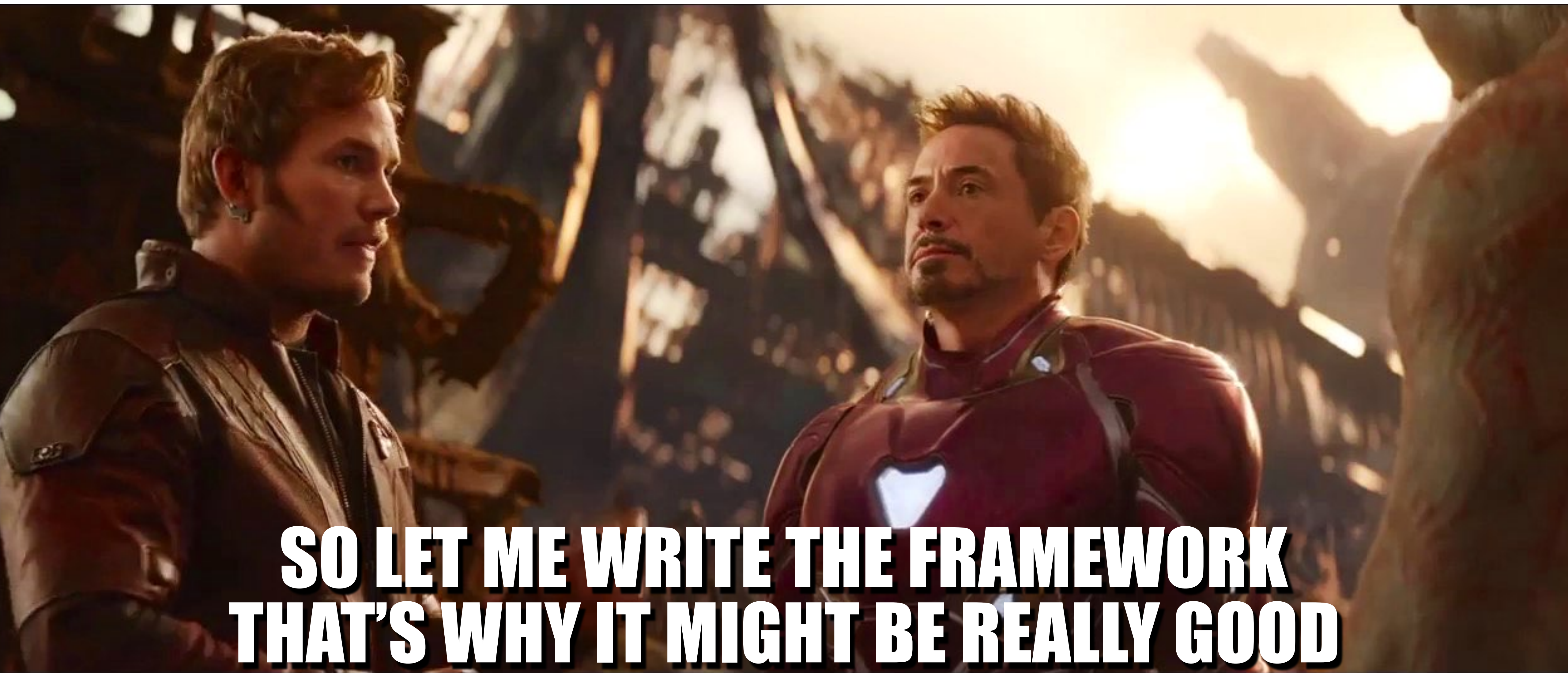


1

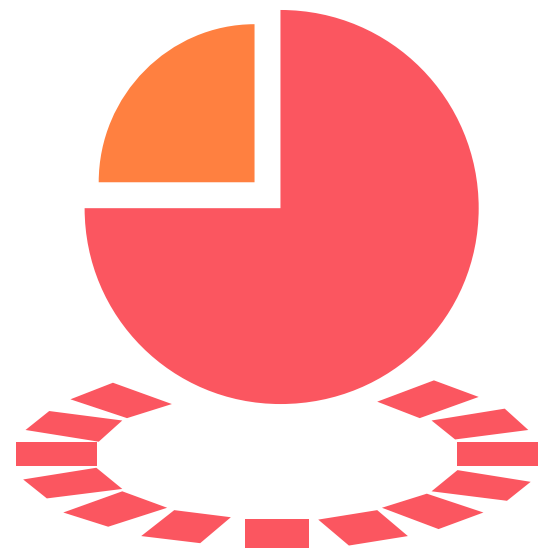


https://twitter.com/monitoring_king/status/1048264580743479296

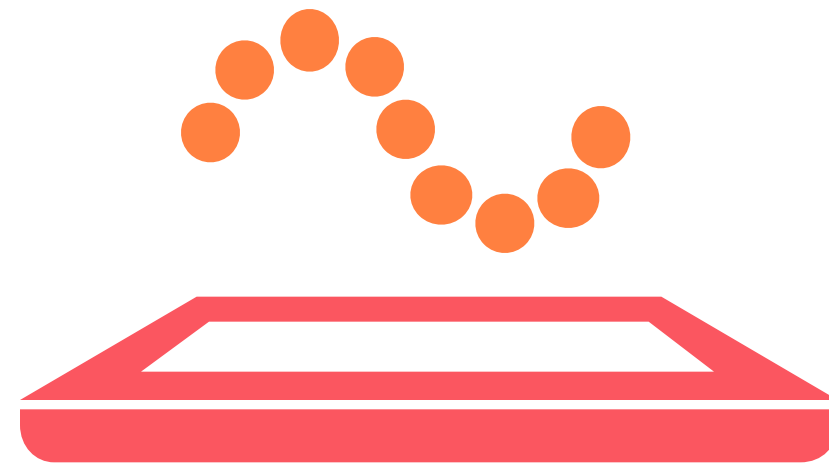




Every framework wants to be when it grows up



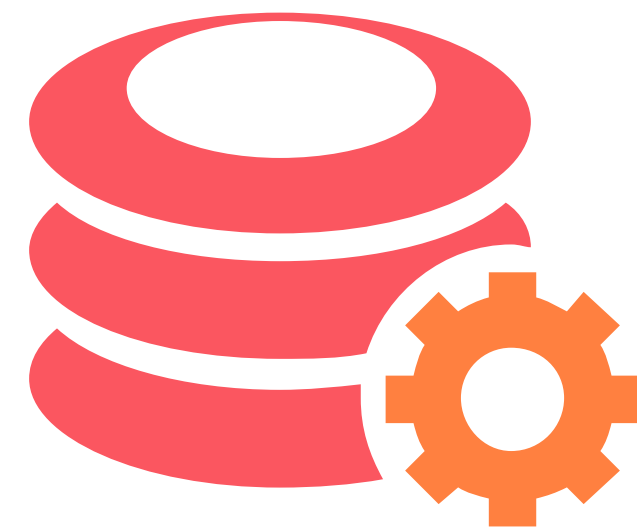
Scalable



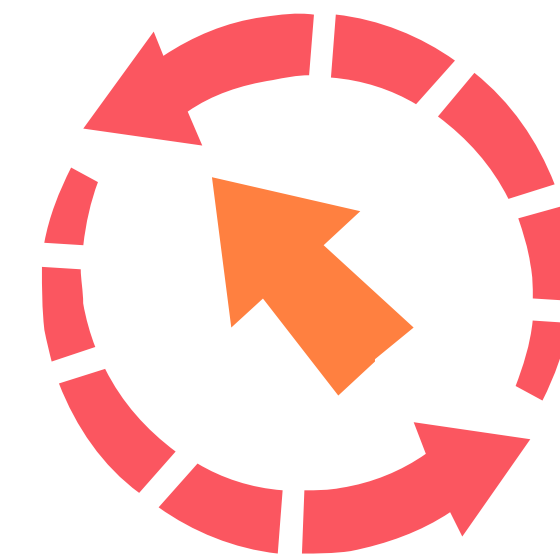
Elastic



Fault-tolerant



Stateful



Distributed



Rahul Bhanushali

@157rahul

Follow



Replying to @gAmUssA @kafkstreams

Some of the hardest things for me were:

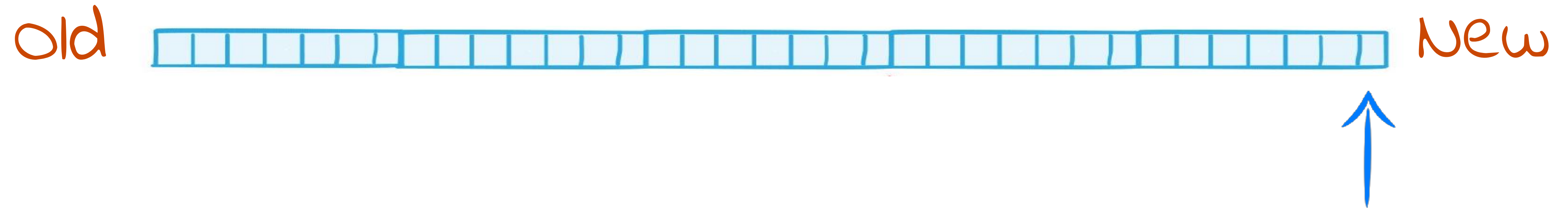
1. Impact of changing the name of the streams app and redeploying it
2. How parallelism is dependent on kafka topic partitions and why adding more nodes of the streams app than the number of partitions is useless.

5:56 PM - 11 Oct 2018



<https://twitter.com/157rahul/status/1050505569746841600>

The log is a simple idea



Messages are added at the end of the log

Consumers have a position all of their own

George
is here

Scan



Old



New

Fred

Scan



is here

Sally

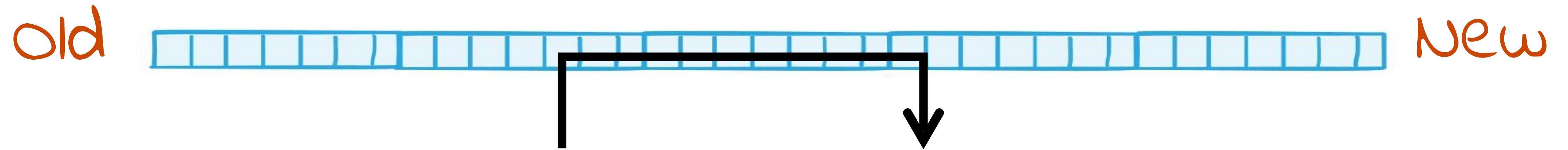
Scan



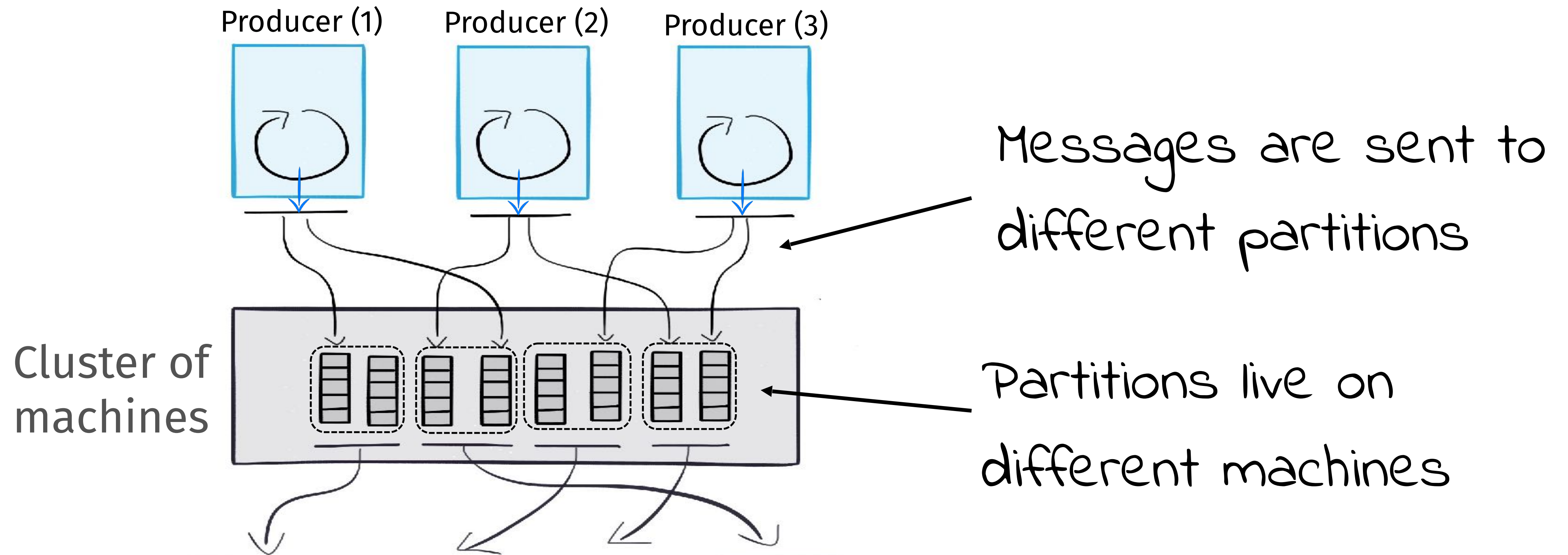
is here

Only Sequential Access

Read to offset & scan



Shard data to get scalability





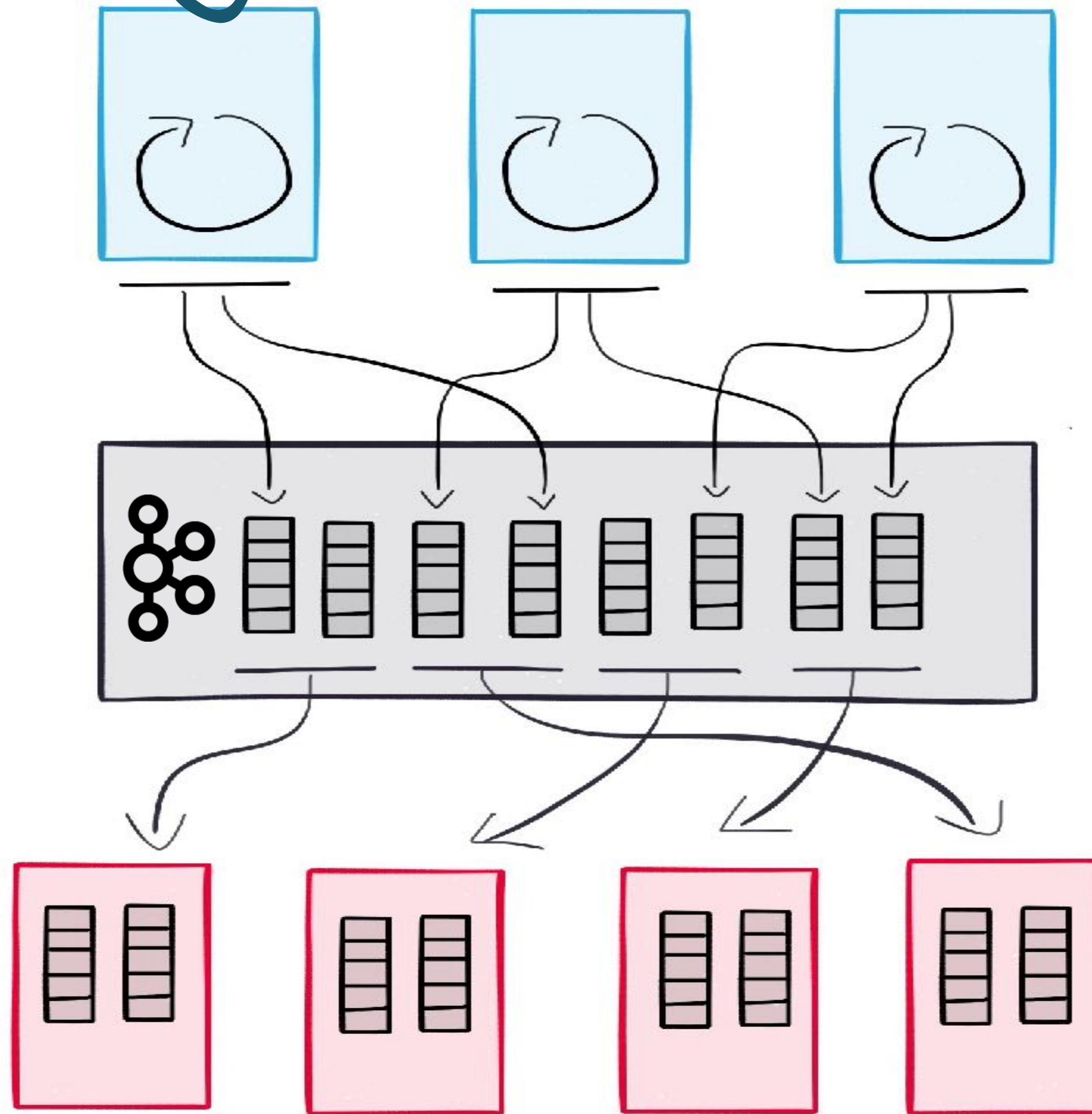
**CONSUMER GROUP
COORDINATOR**

CONSUMERS

CONSUMER GROUP

Linearly Scalable Architecture

Producers



Consumers

Single topic:

- Many producers machines
- Many consumer machines
- Many Broker machines

No Bottleneck!!

Talk is cheap! Show me code!

<https://cnfl.io/streams-movie-demo>

**As developers,
we want to build APPS
not INFRASTRUCTURE**

@gamussa

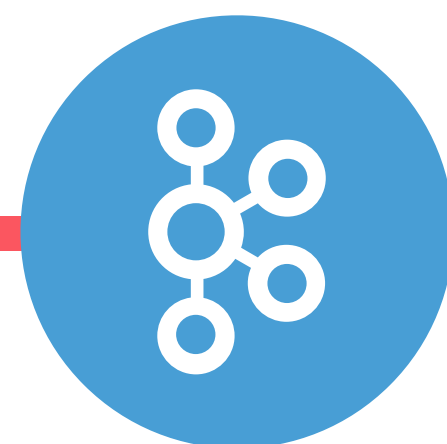
#TorontoKafka

@confluentinc





the **KAFKA STREAMS API** is a
JAVA API to
BUILD REAL-TIME APPLICATIONS



App
Streams
API

**Not running
inside brokers!**



Same app, many instances

App
Streams
API

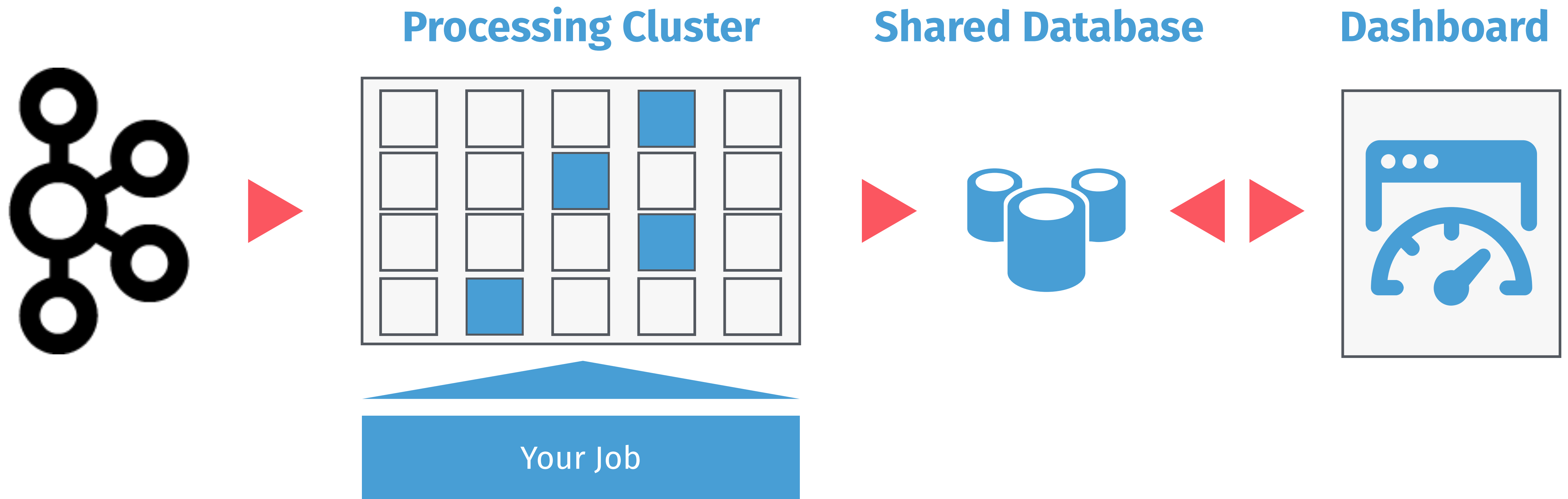
App
Streams
API

App
Streams
API

Brokers?
Nope!

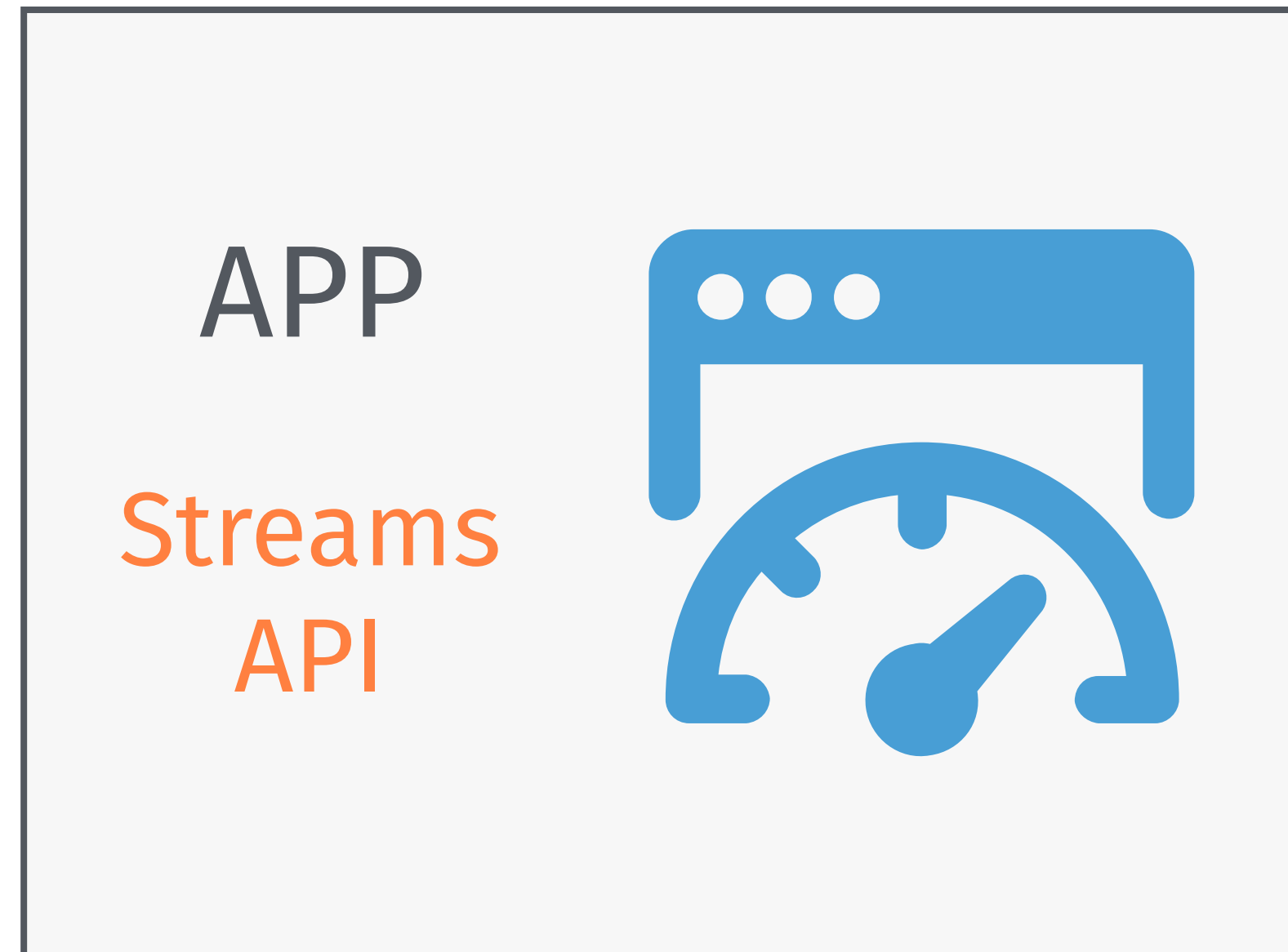
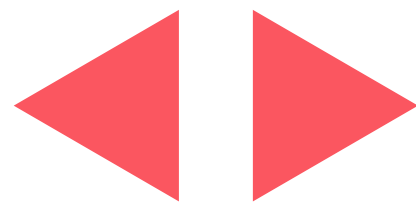


Before



After

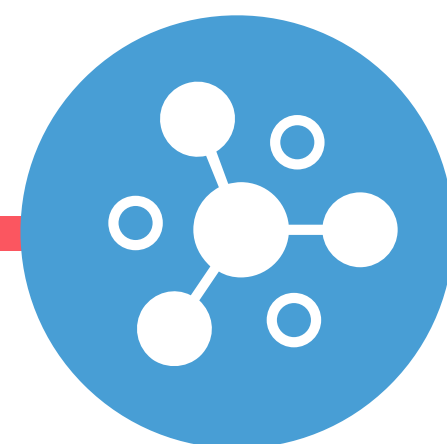
Dashboard





this means you can

DEPLOY your app **ANYWHERE** using
WHATEVER TECHNOLOGY YOU WANT



So many places to run you app!



Physical



MESOS

vmware®



TERRAFORM



ANSIBLE



Jenkins

...and many more...

confluent

@gamussa

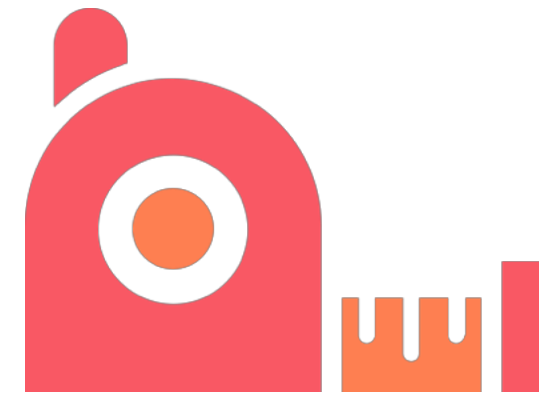
#TorontoKafka

@confluentinc

Things Kafka Stream Does



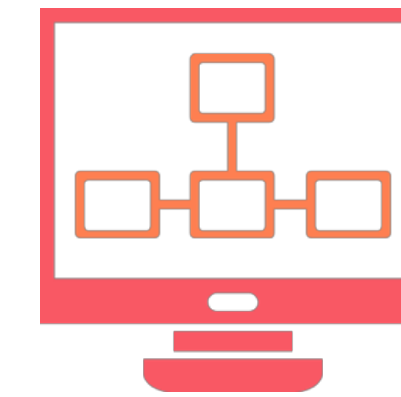
Enterprise Support



Open Source



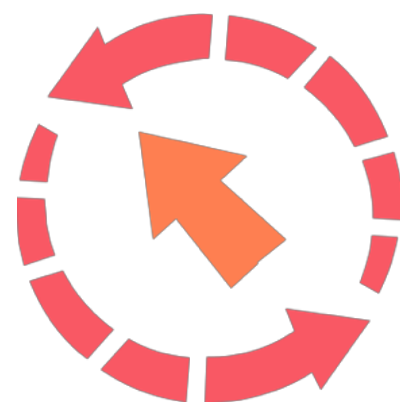
Runs Everywhere



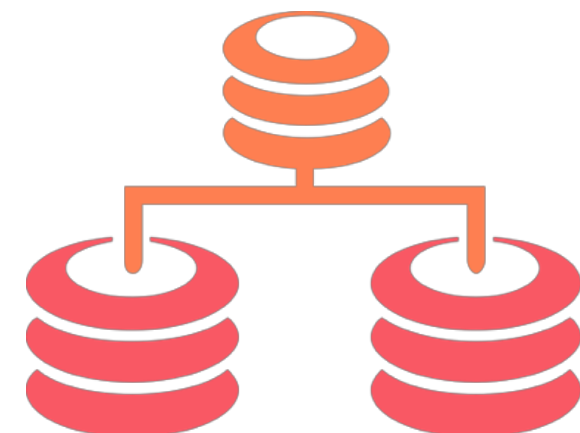
Elastic, Scalable,
Fault-tolerant



Kafka Security
Integration



Powerful Processing incl.
Filters, Transforms, Joins,
Aggregations, Windowing



Supports Streams
and Tables

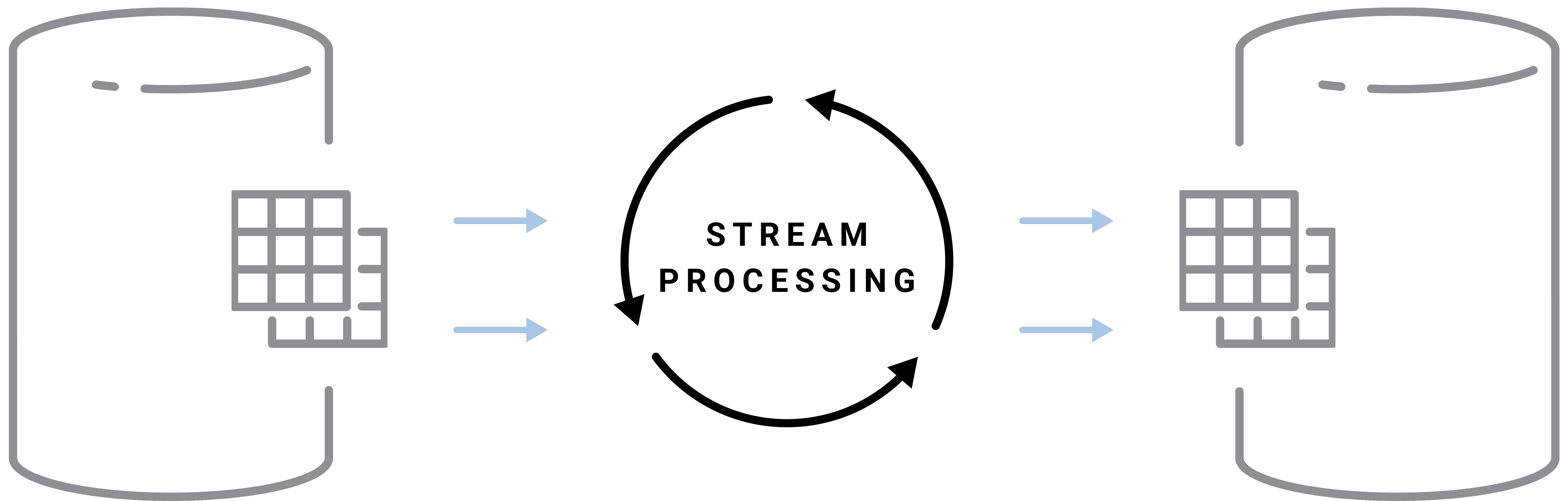


Exactly-Once
Processing

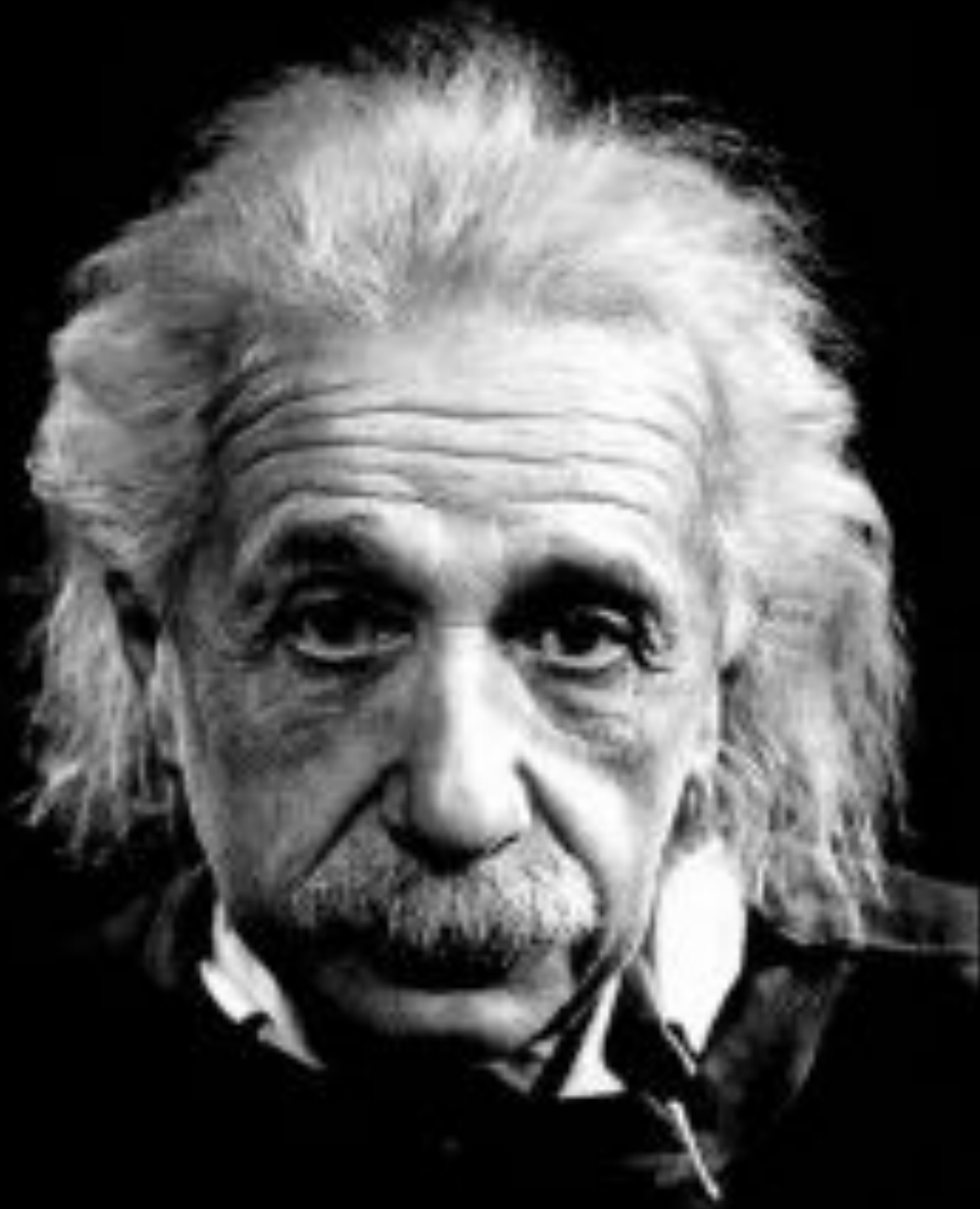


Event-Time
Processing

Table-Stream Duality



SPECIAL RELATIVITY



OF STREAMS AND TABLES

TABLE

Gwen	1
------	---

Gwen	1
Matthias	1

Gwen	2
Matthias	1

Gwen	2
Matthias	1
Viktor	1

STREAM

("Gwen", 1)

("Matthias", 1)

("Gwen", 2)

("Viktor", 1)

TABLE

Gwen	1
------	---

Gwen	1
Matthias	1

Gwen	2
Matthias	1

Gwen	2
Matthias	1
Viktor	1

Do you think that's a **table**
you are querying ?



Talk is cheap! Show me code!

What's next?

**Rams**

@IDispose

[Follow](#)Replying to [@gAmUssA](#) [@rmoff](#) [@kafkastreams](#)

Having to learn Java. My background is non Java. Learned Scala just you be able yo put together a streams app. But now that same feature is available in ksql via udaf, hopefully experience will be better.

11:56 AM - 6 Oct 2018

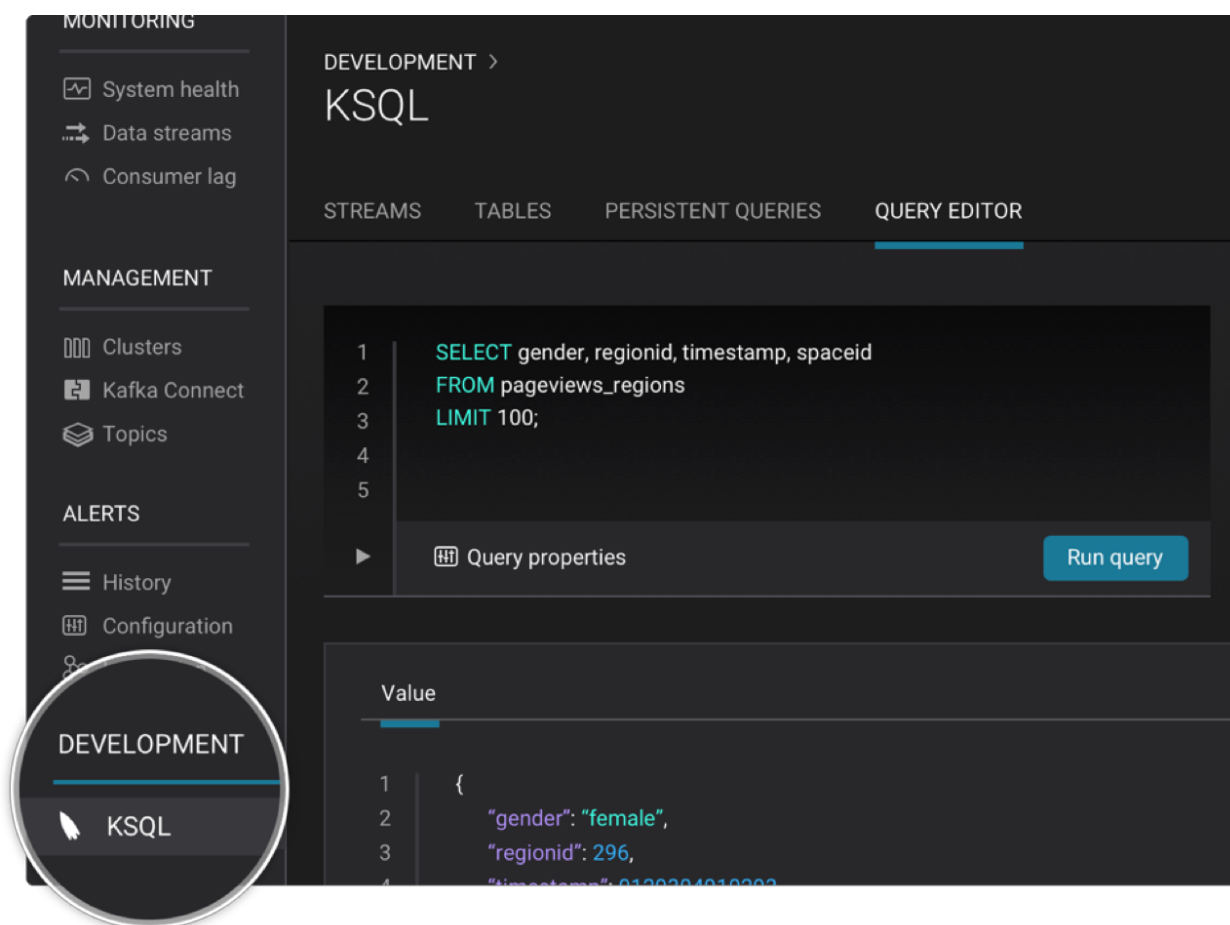


1



<https://twitter.com/IDispose/status/1048602857191170054>

KSQL #FTW



1 UI

ksql>

2 CLI

POST /query

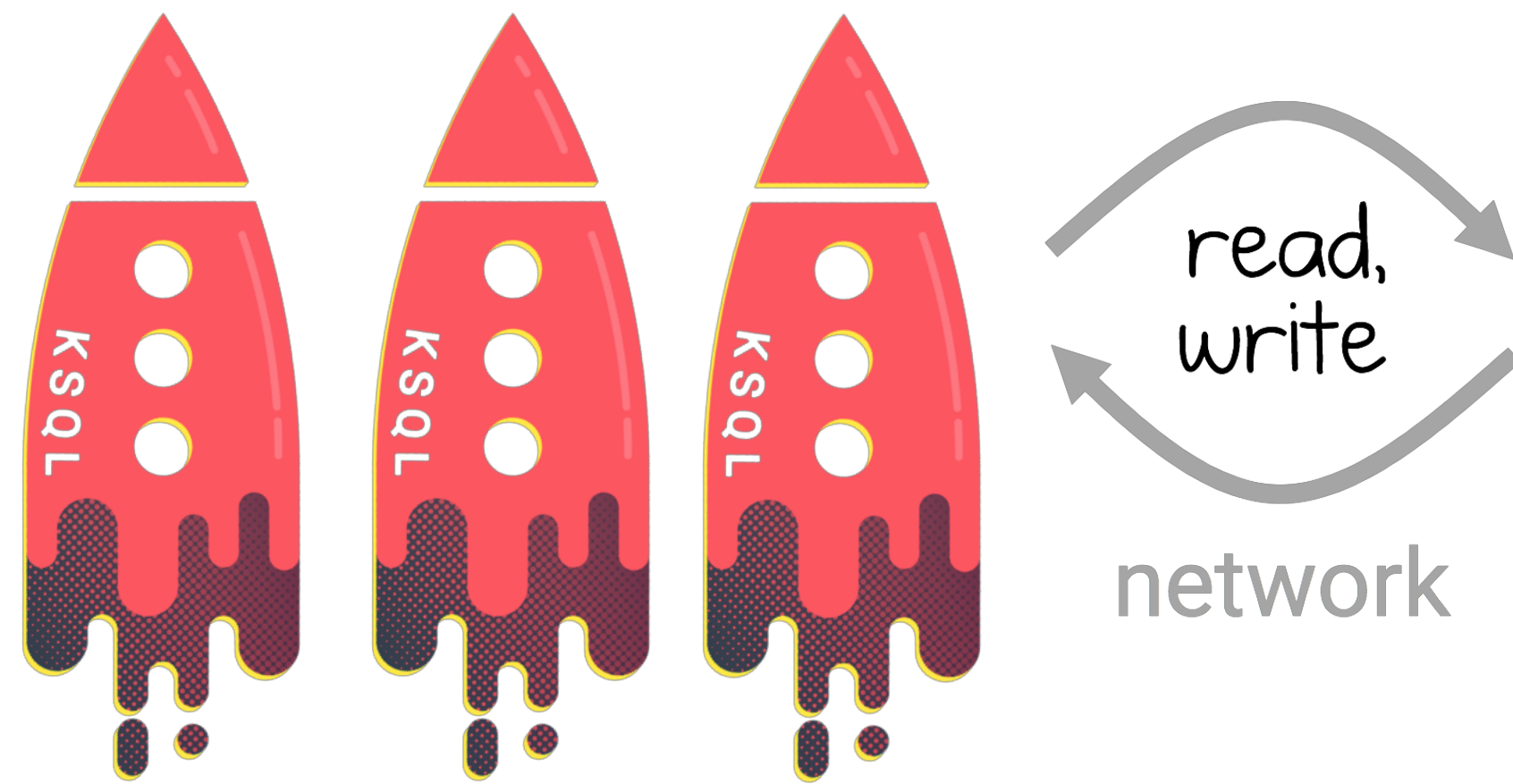
3 REST



4 Headless

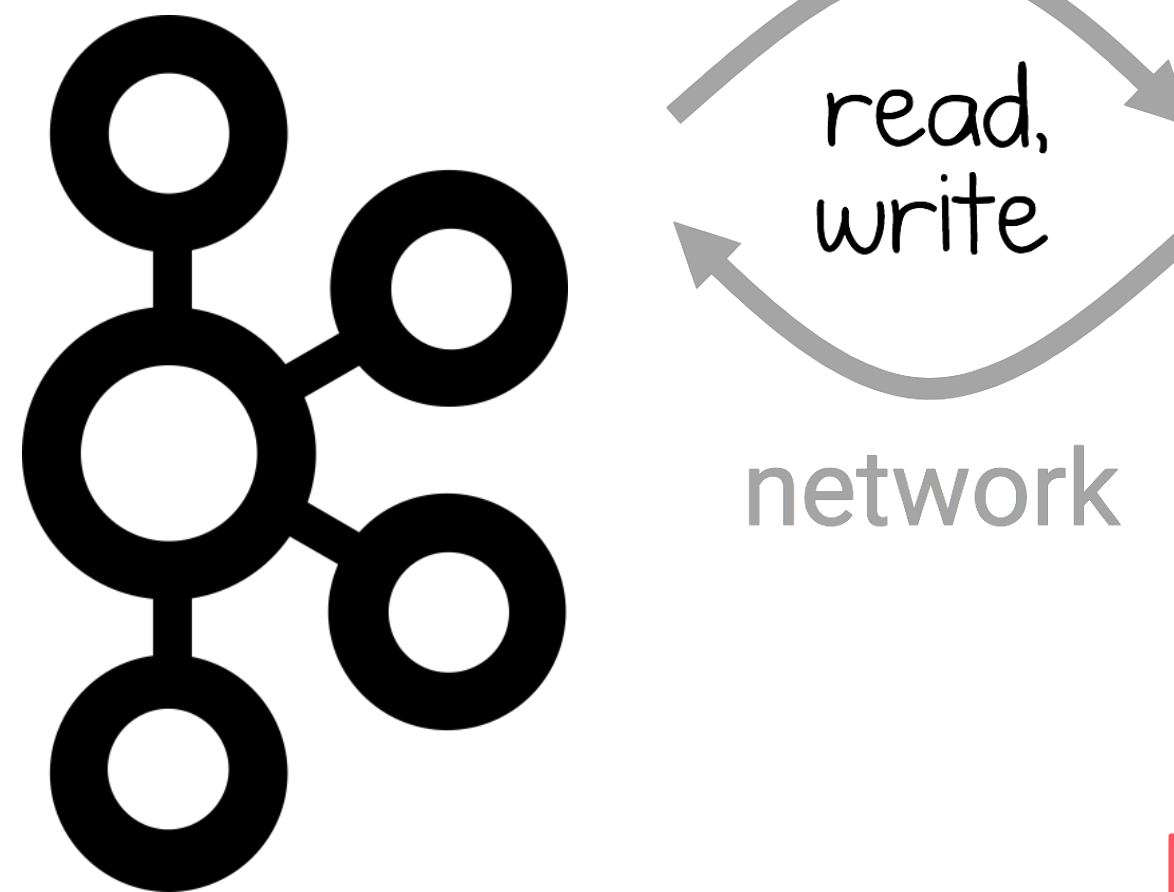
Interaction with Kafka

KSQL
(processing)



Does not run on
Kafka brokers

Kafka
(data)



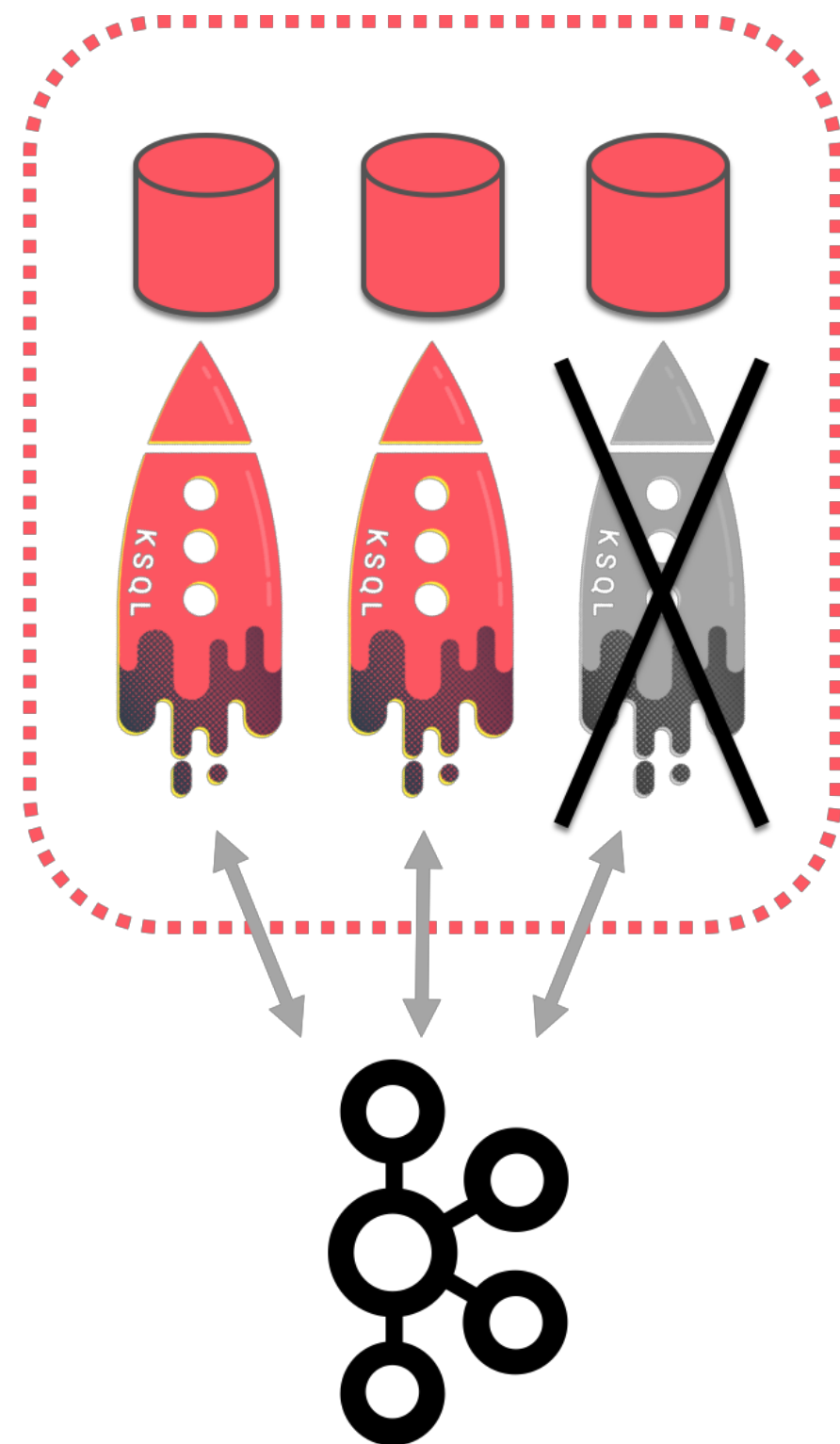
JVM application
with Kafka Streams (processing)



Does not run on
Kafka brokers

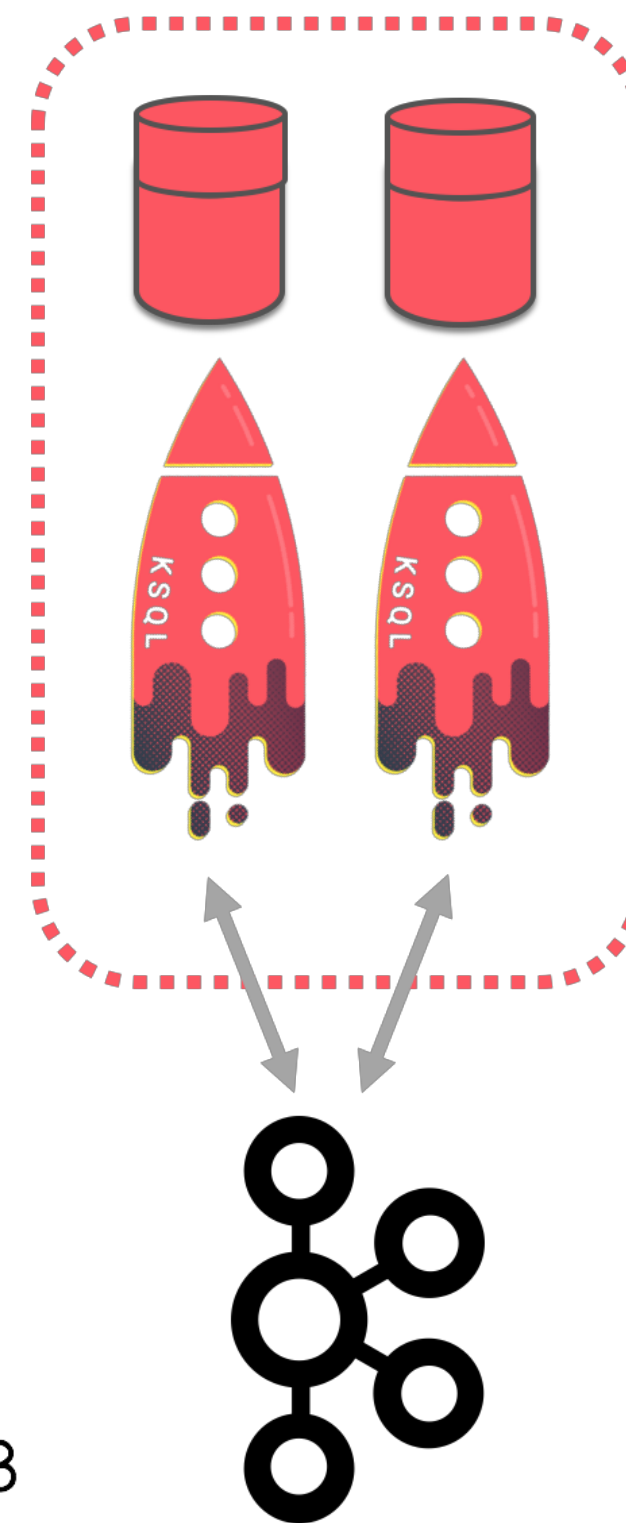
Fault-Tolerance, powered by Kafka

#3 died so #1 and #2 take over



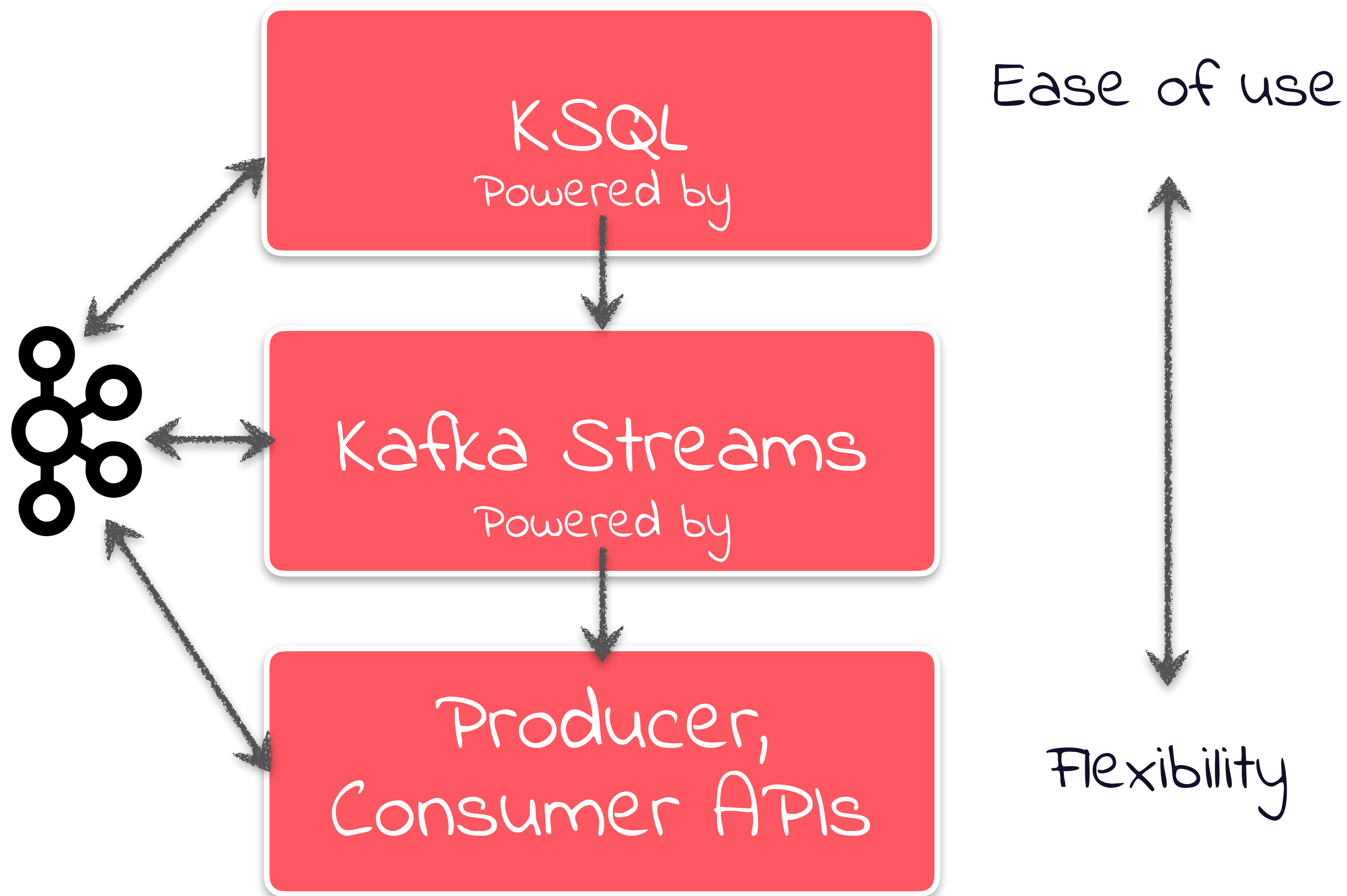
- 1 Kafka consumer group rebalance is triggered
- 2 Processing and **state** of #3 is migrated via Kafka to remaining servers #1 + #2

#3 is back so the work is split again



- 1 Kafka consumer group rebalance is triggered
- 2 Part of processing incl. **state** is migrated via Kafka from #1 + #2 to server #3

Standing on the shoulders of Streaming Giants



```
CREATE STREAM,
CREATE TABLE, SELECT, JOIN,
GROUP BY, SUM, ...
```

KSQL UDFs

```
KStream<>, KTable<>, filter(), map(),
flatMap(), join(), aggregate(),
transform(), ...
```

```
subscribe(), poll(), send(),
flush(), beginTransaction(), ...
```

THANKS!

@gamussa

viktor@confluent.io

We are hiring!

<https://www.confluent.io/careers/>