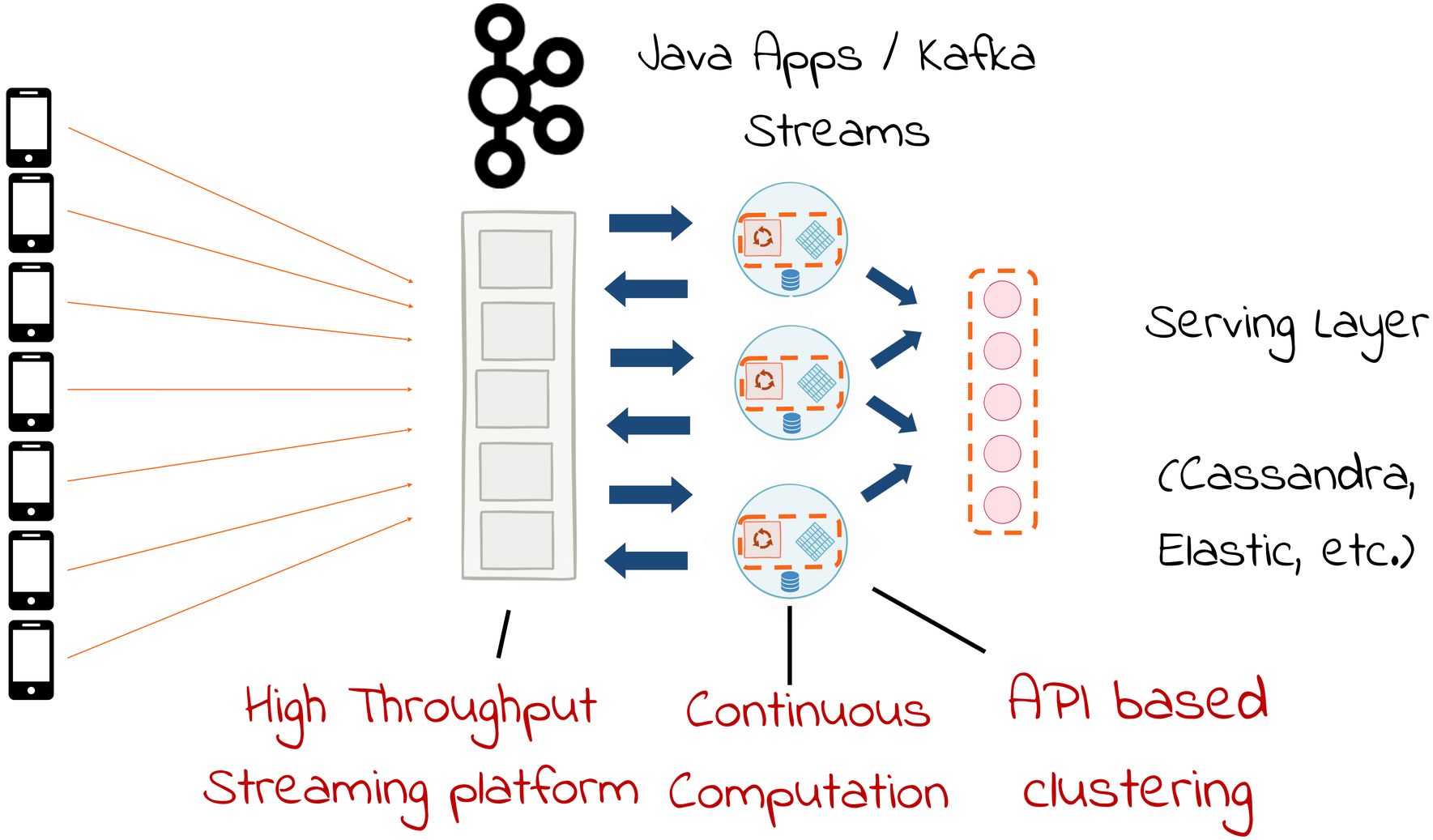

Crossing the Streams:
Rethinking Stream Processing with
Kafka Streams and KSQL
TORONTO
TORONTO KAFKA MEETUP,
NOVEMBER 2018

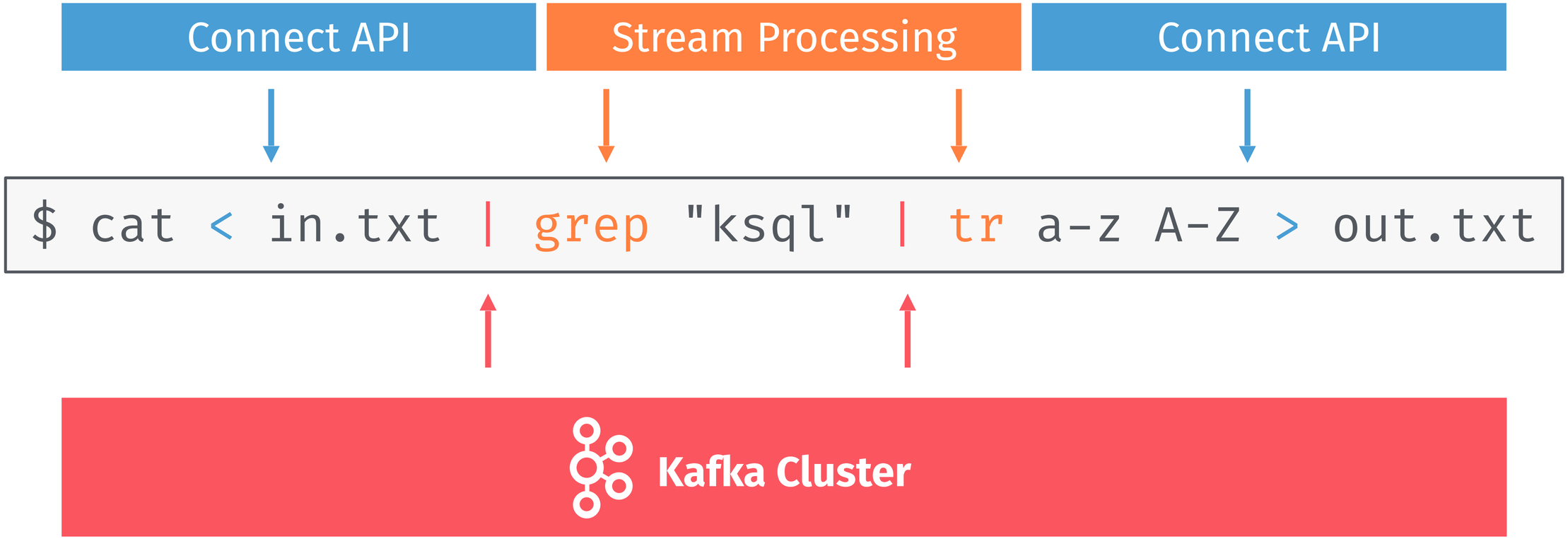
confluent

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

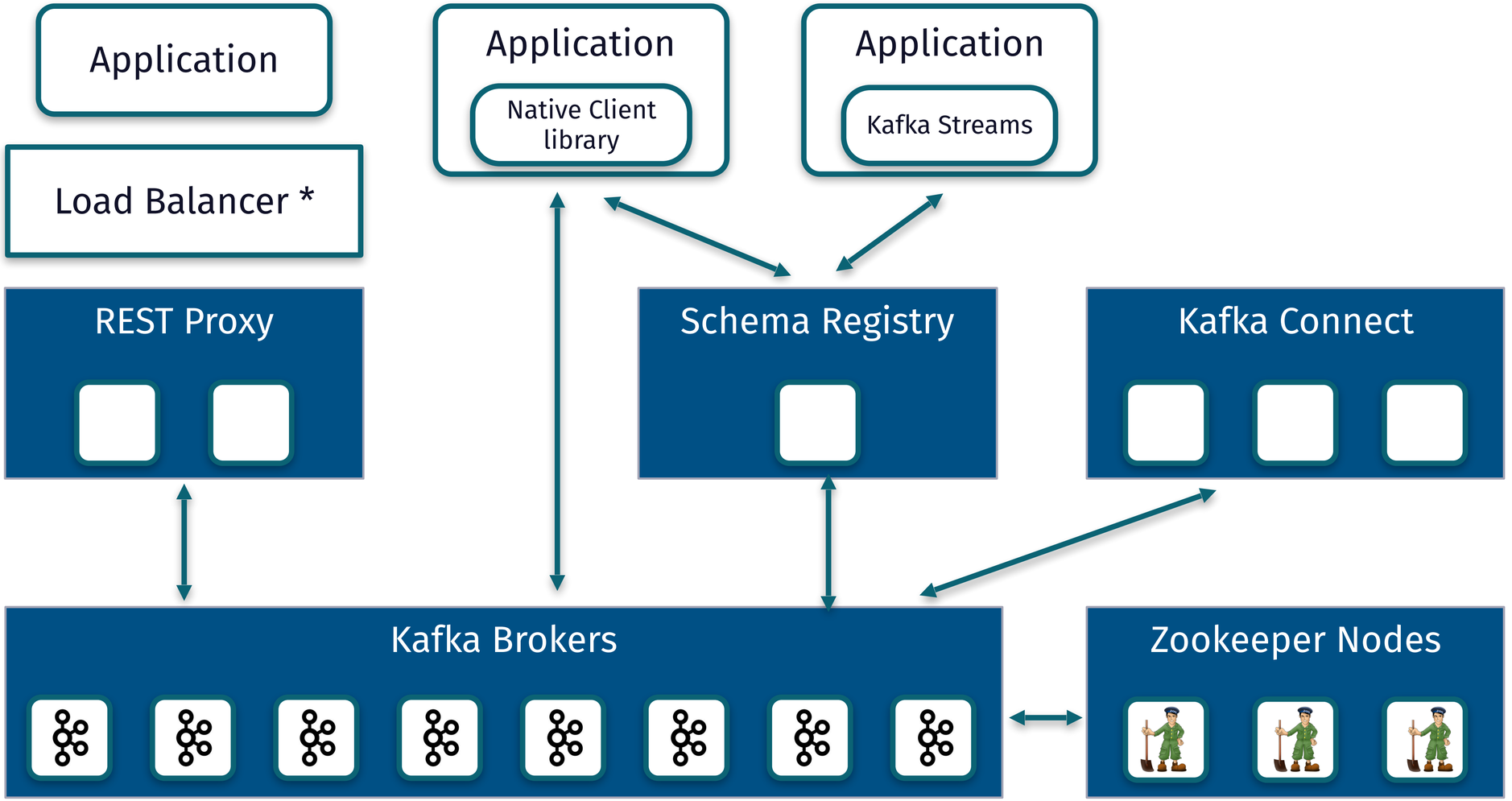
Streaming
is the toolset
for dealing
with events
as they move!


Java Apps / Kafka Streams
Serving Layer
(Cassandra, Elastic, etc.)
High Throughput Streaming platform
Continuous Computation
API based clustering

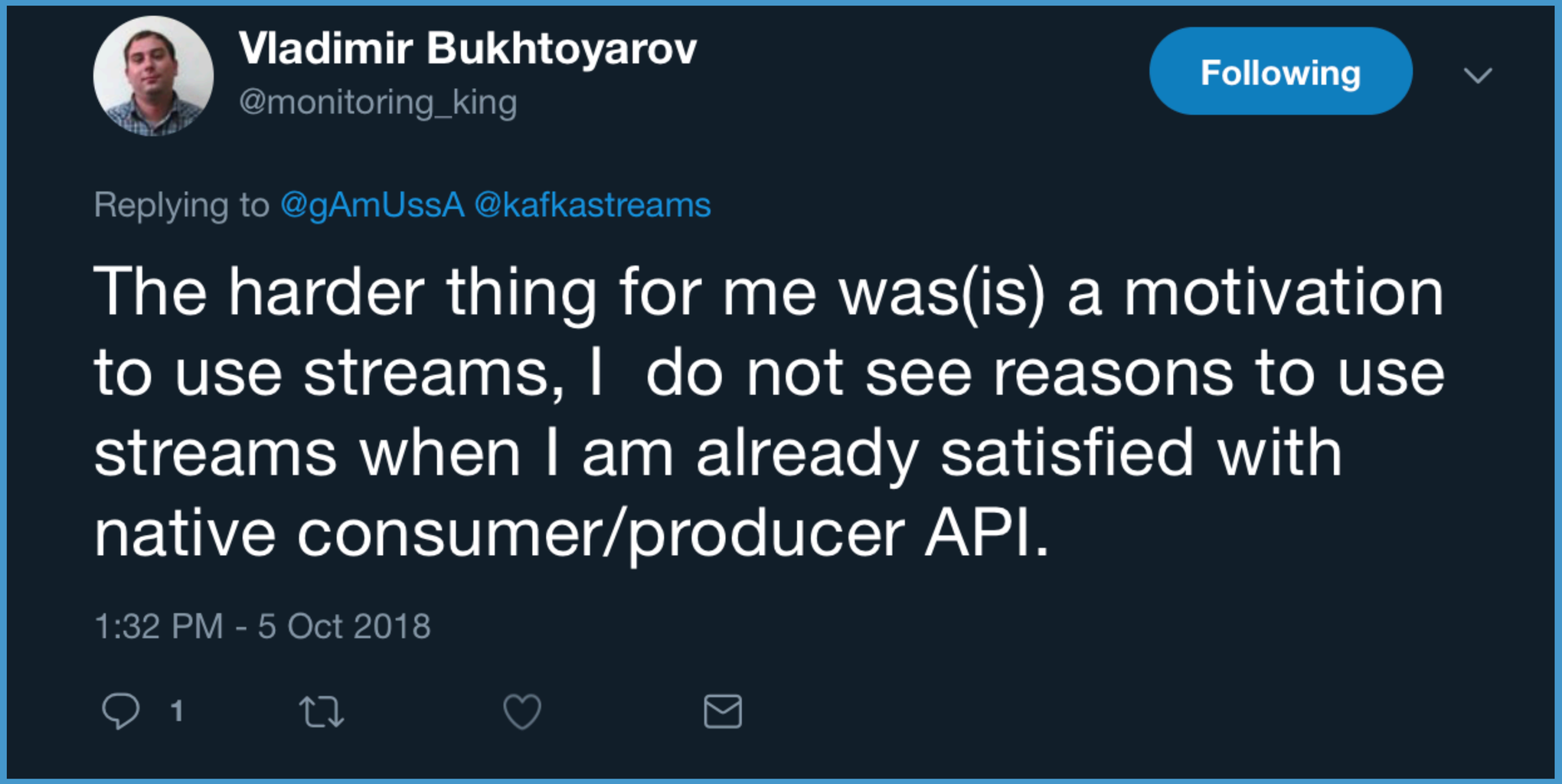
Stream Processing by Analogy



Streaming Platform Architecture

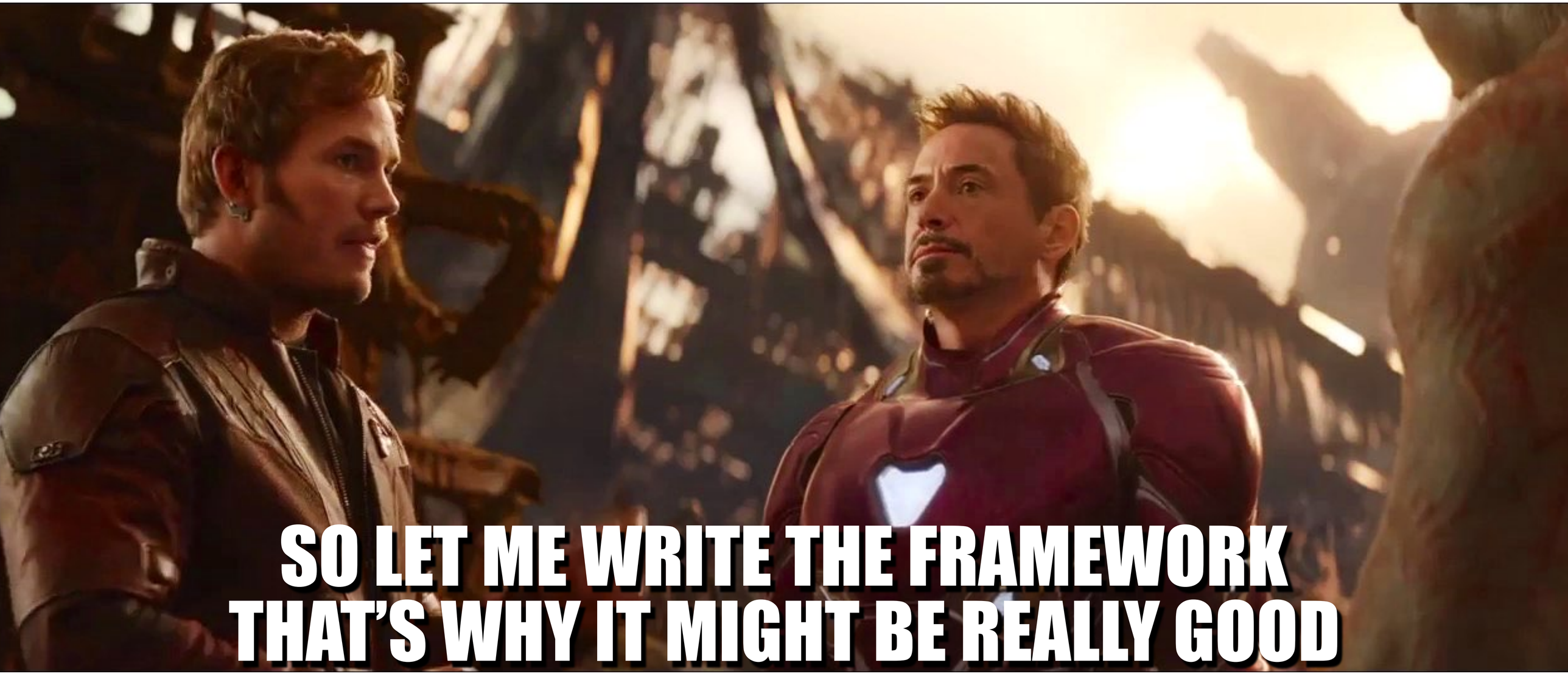


confluent



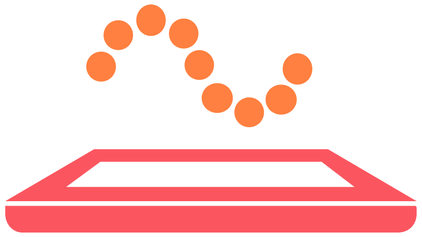
https://twitter.com/monitoring_king/status/1048264580743479296


LET'S TALK ABOUT THIS FRAMEWORK
OF YOURS.
I THINK ITS GOOD, EXCEPT IT SUCKS


SO LET ME WRITE THE FRAMEWORK
THAT'S WHY IT MIGHT BE REALLY GOOD

Every framework wants to be when it grows up

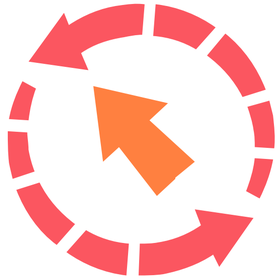
Scalable



Elastic

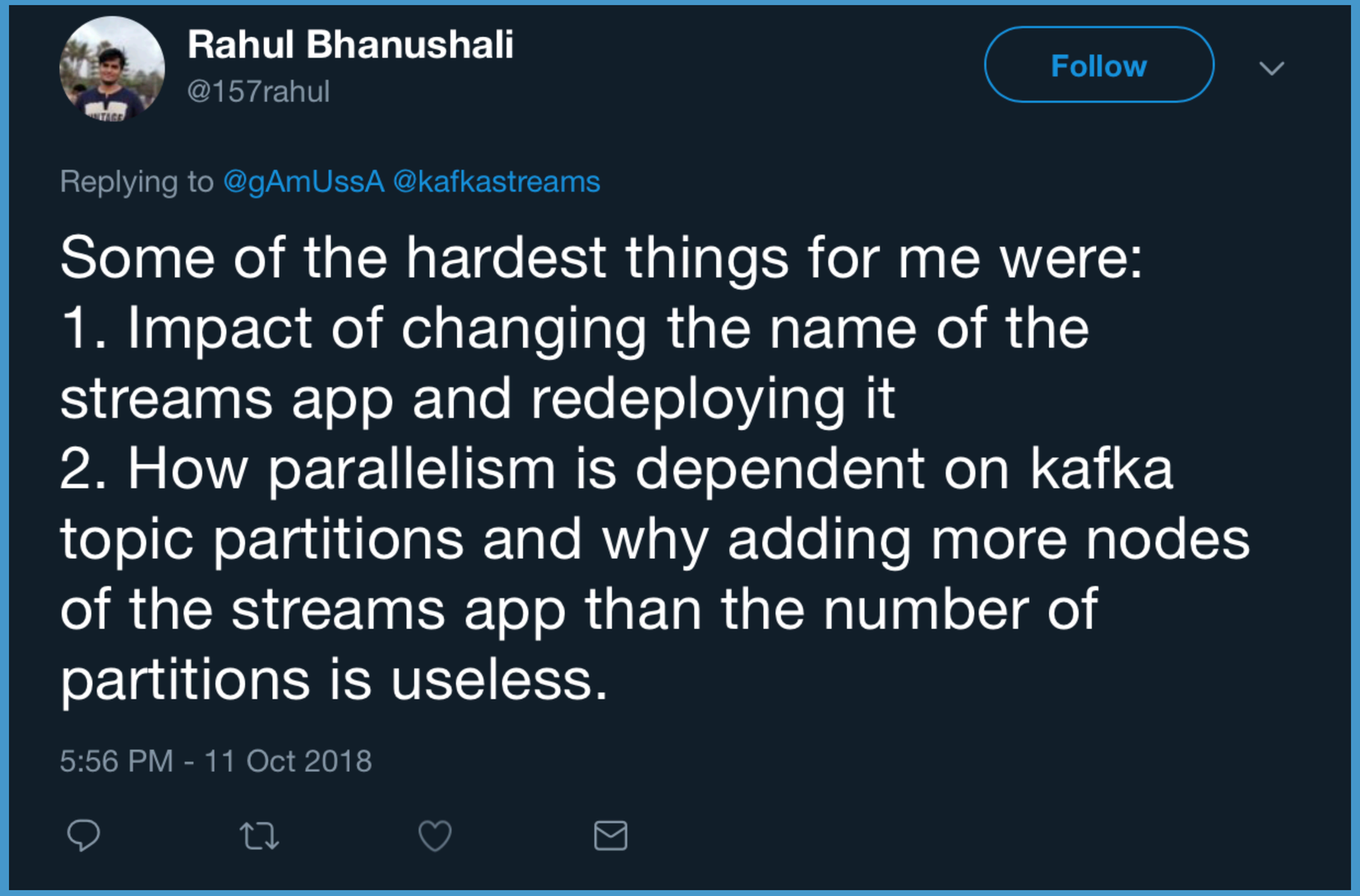
Fault-tolerant

Stateful



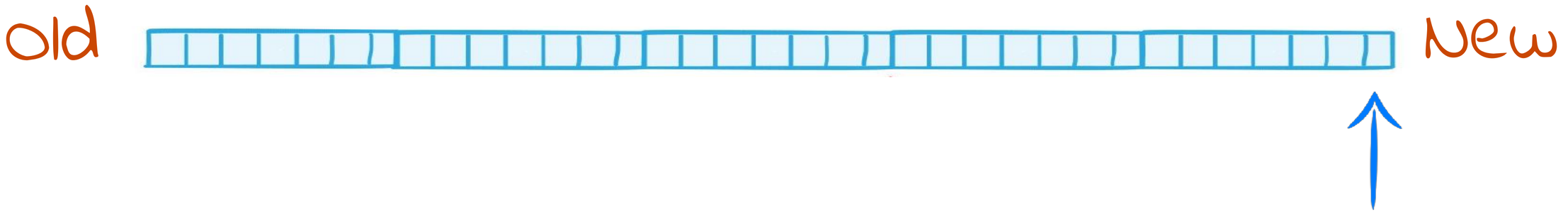
Distributed

confluent



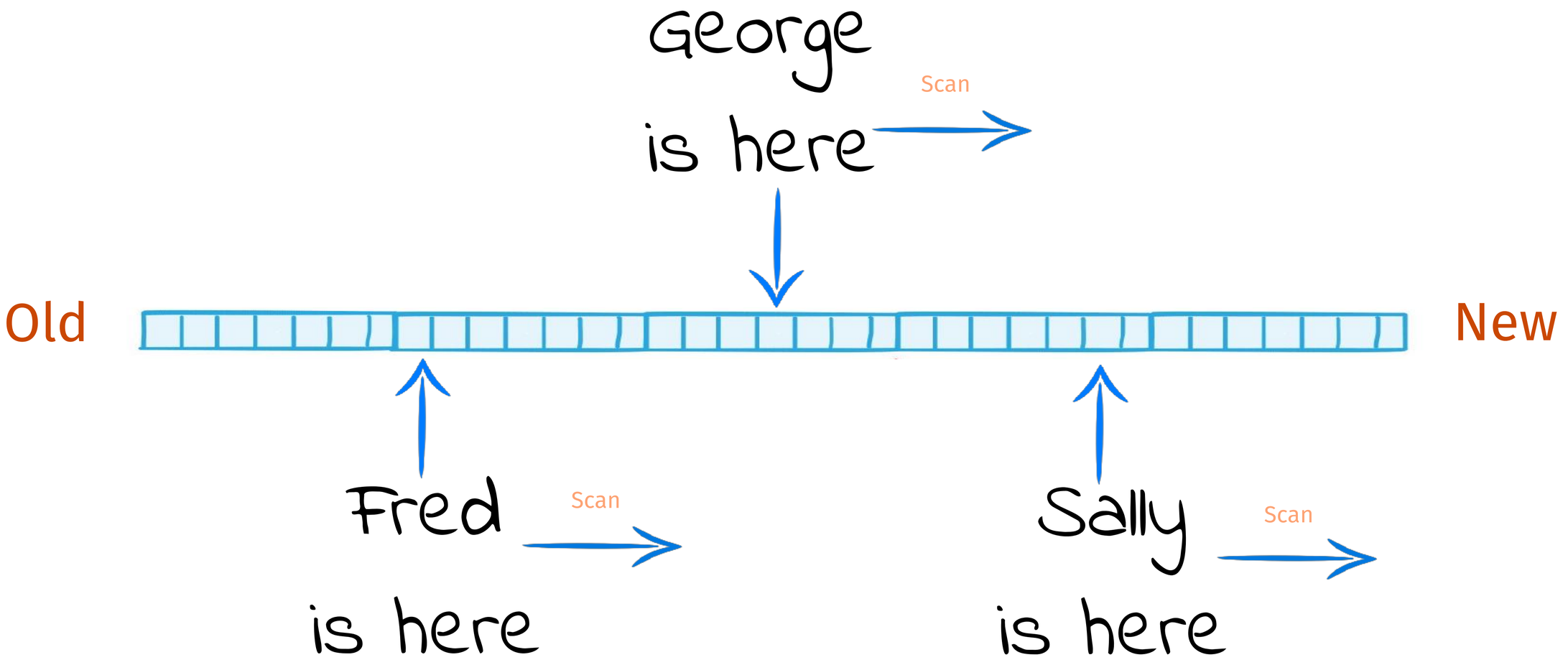
https://twitter.com/157rahul/status/1050505569746841600

The log is a simple idea

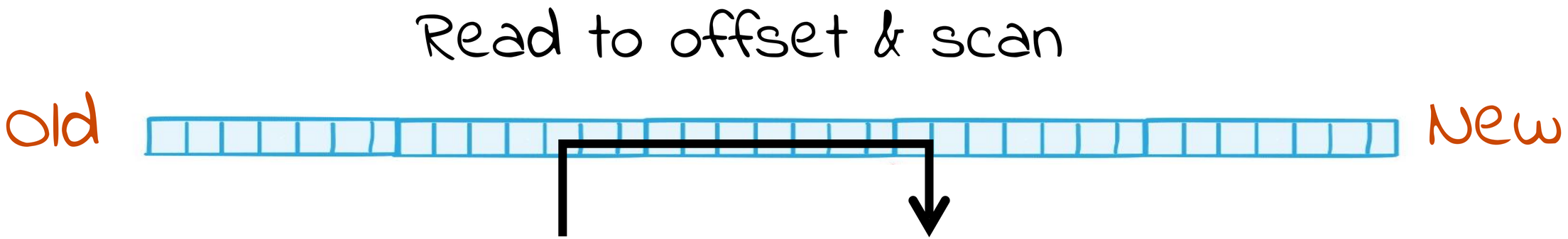


Messages are added at the end of the log

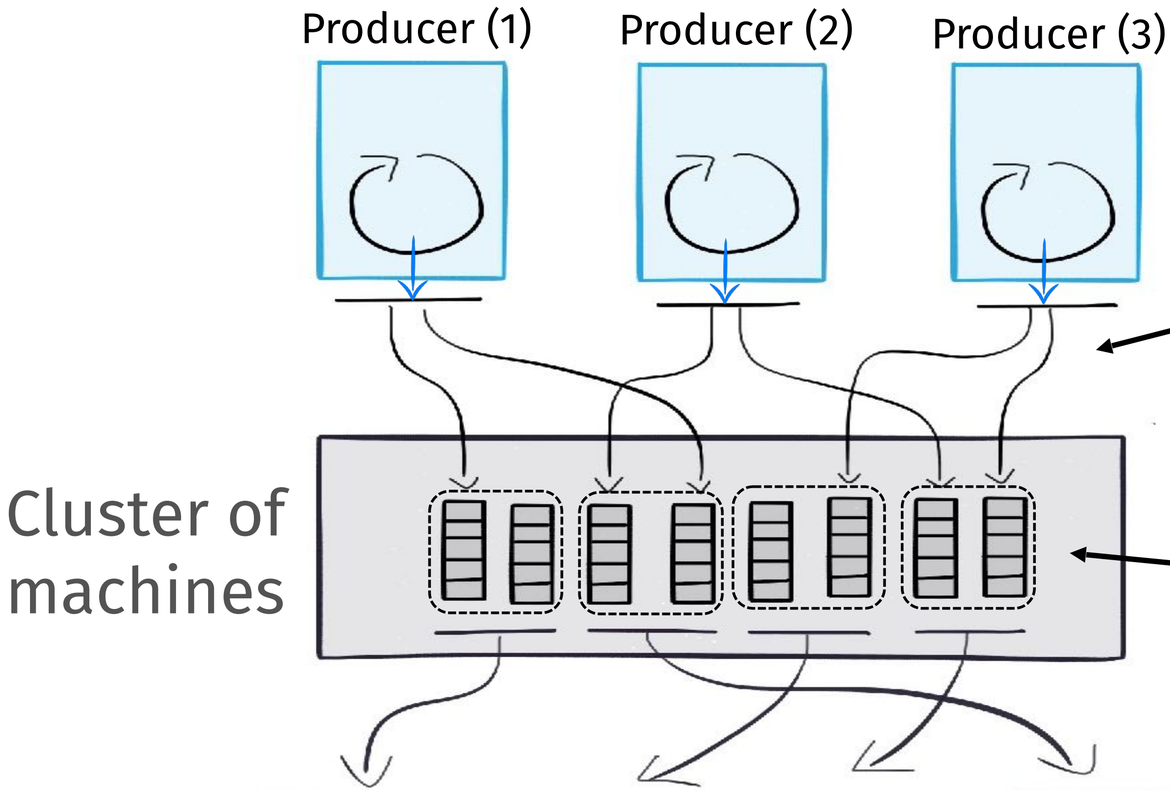
Consumers have a position all of their own



Only Sequential Access



Shard data to get scalability

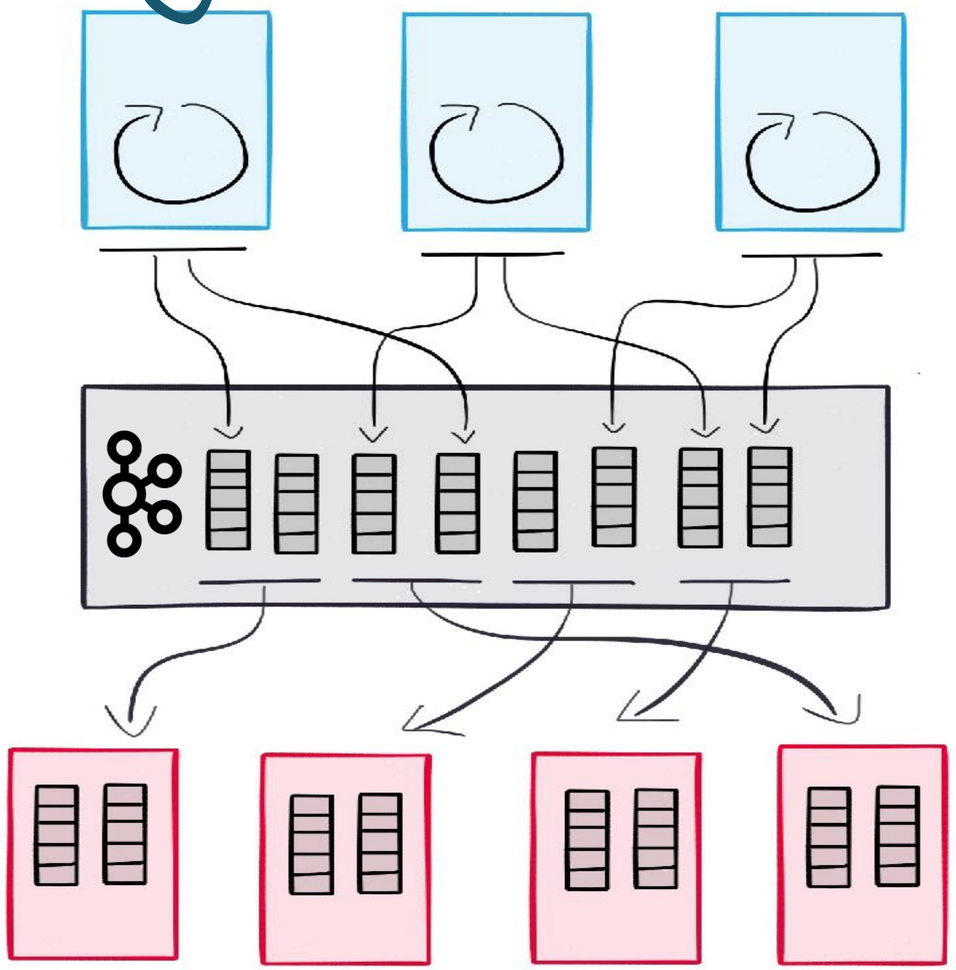


Messages are sent to different partitions

Partitions live on different machines


CONSUMERS
CONSUMER GROUP COORDINATOR
CONSUMER GROUP

Linearly Scalable Architecture



Single topic:

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

No Bottleneck!!

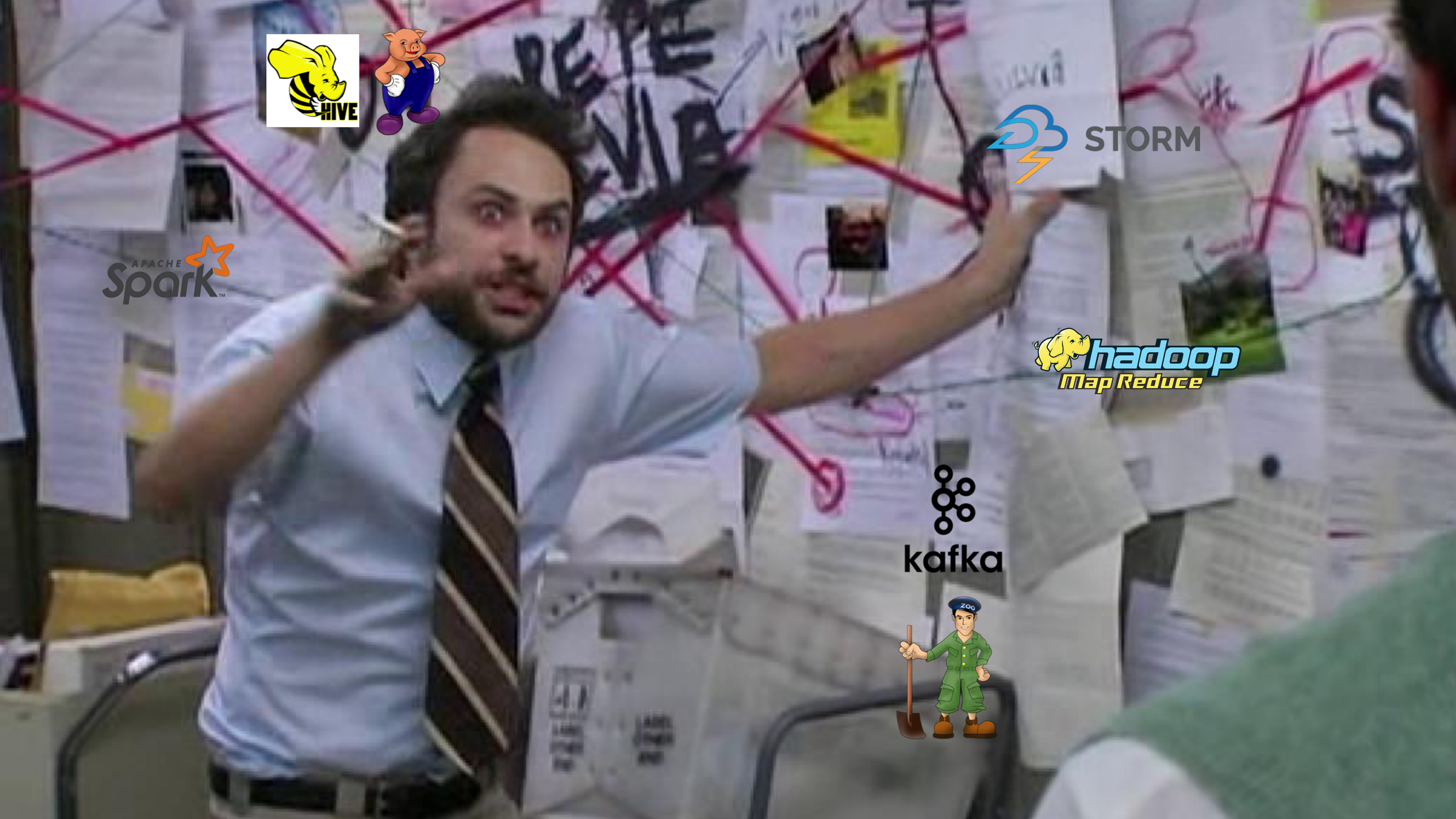
confluent

Talk is cheap! Show me code!

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

As developers,
we want to build APPS

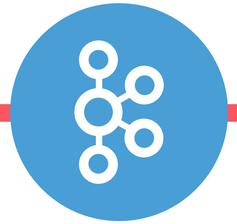
not INFRASTRUCTURE

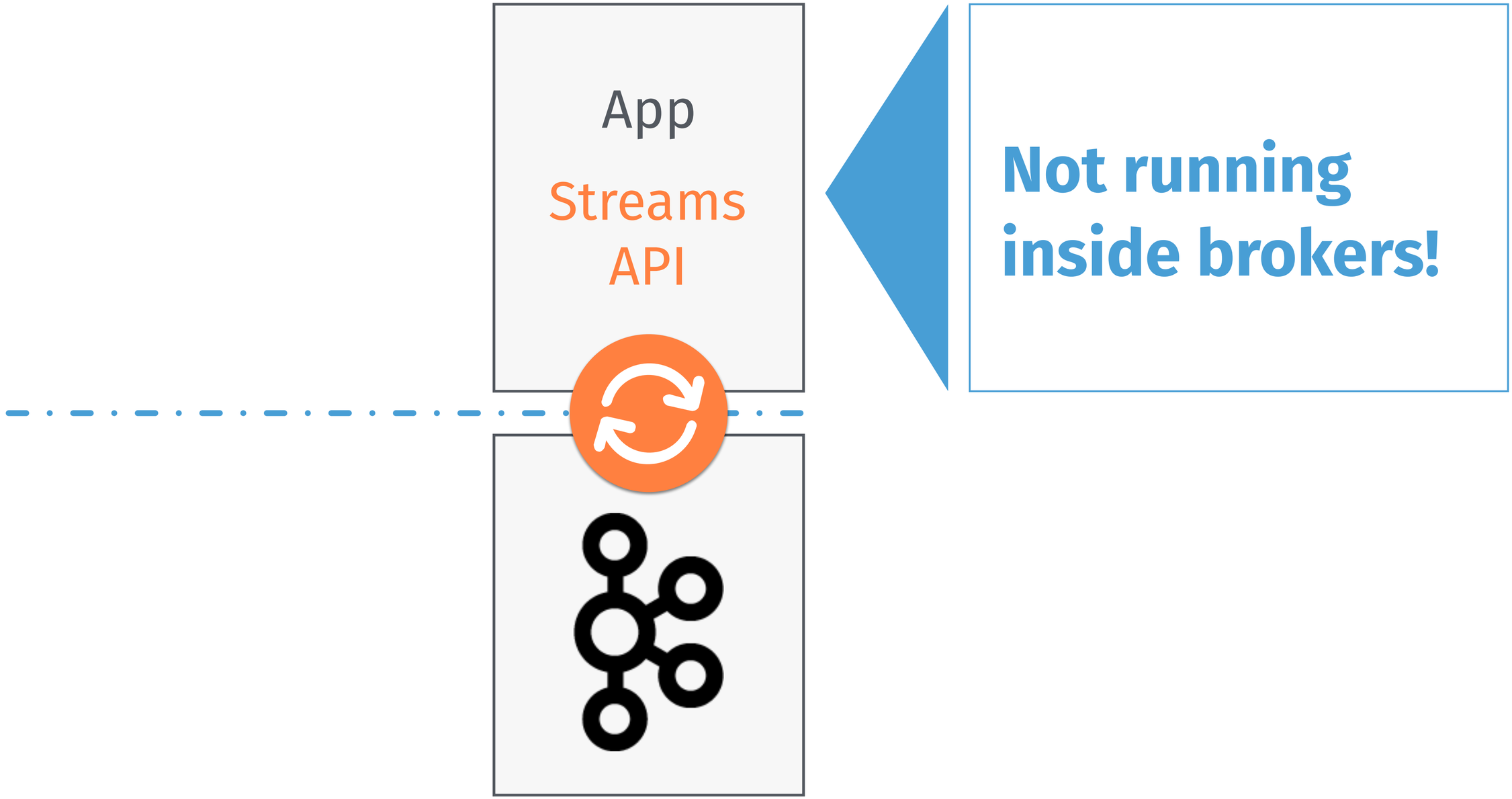

HIVE
STORM
APACHE
Spark
hadoop
Map Reduce
kafka
ZOO

the KAFKA STREAMS **API** is a

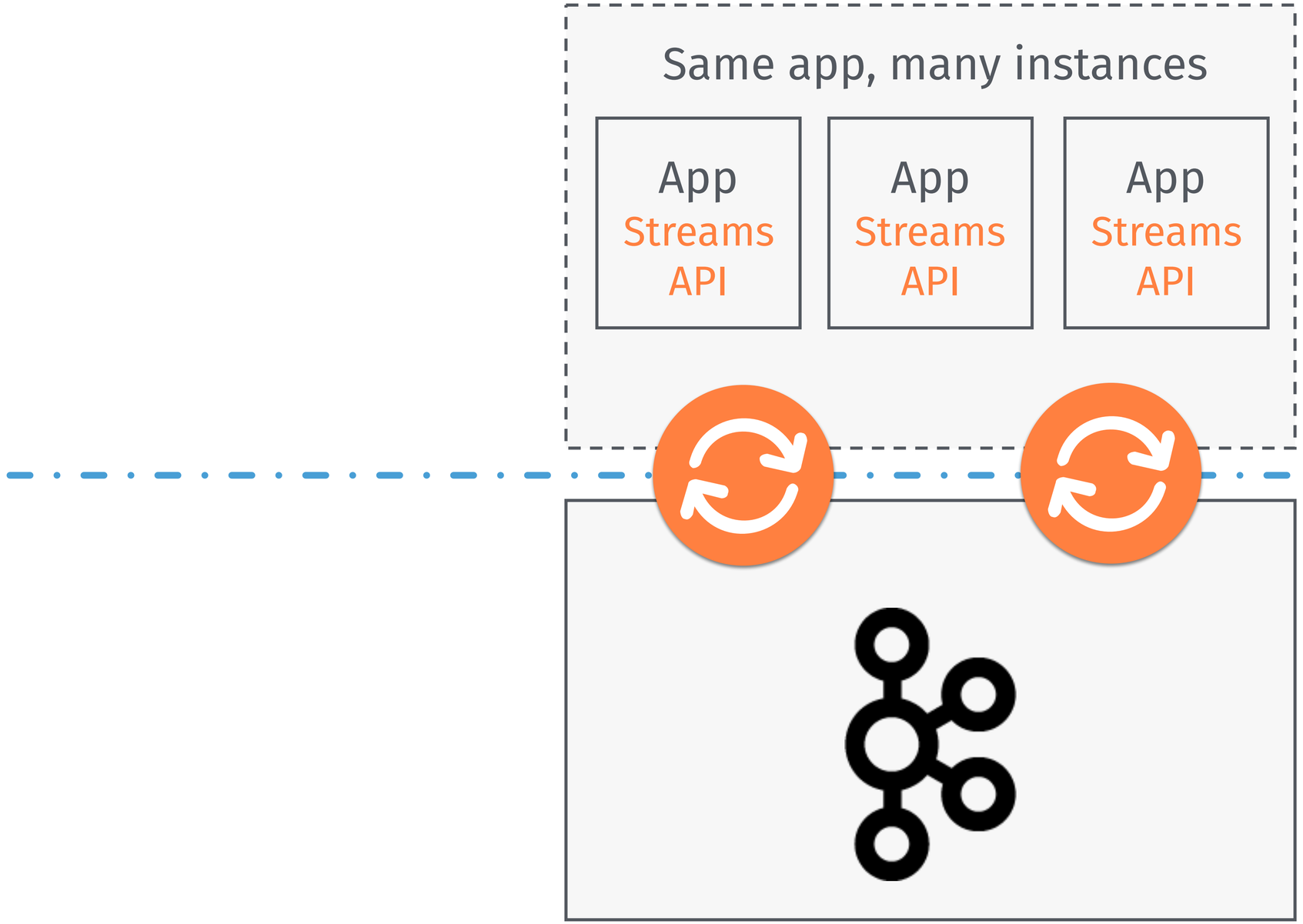
JAVA API to

BUILD REAL-TIME **APPLICATIONS**





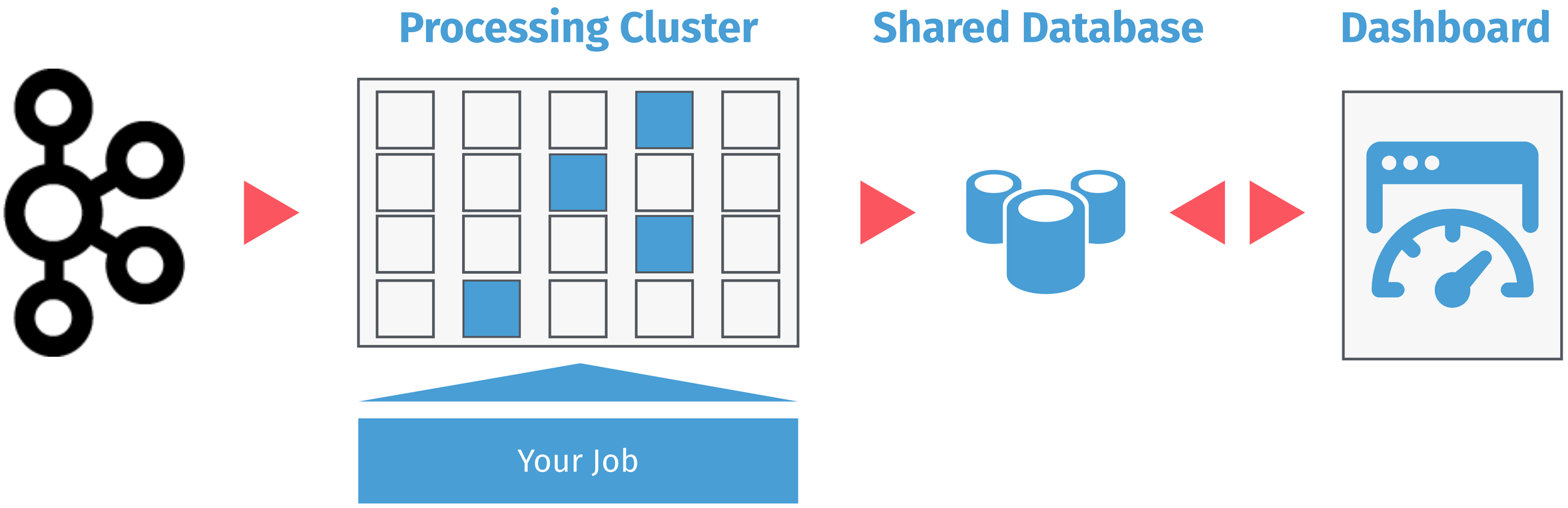
Not running inside brokers!



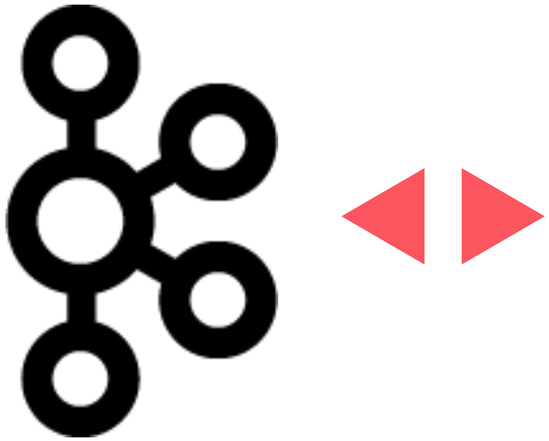
Brokers?

Nope!

Before



After



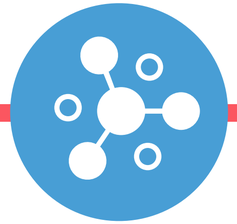
Dashboard

APP

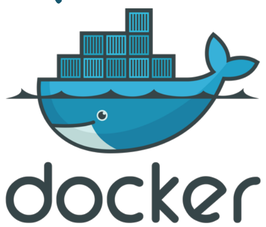
Streams API

this means you can

DEPLOY your app ANYWHERE using WHATEVER TECHNOLOGY YOU WANT



So many places to run you app!

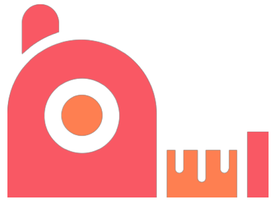


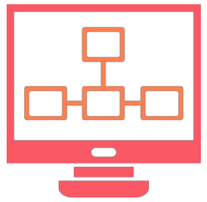

Physical

...and many more...

Things Kafka Stream Does

confluent





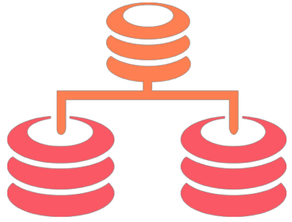
Enterprise Support

Open Source

Runs Everywhere

Elastic, Scalable, Fault-tolerant

Kafka Security Integration



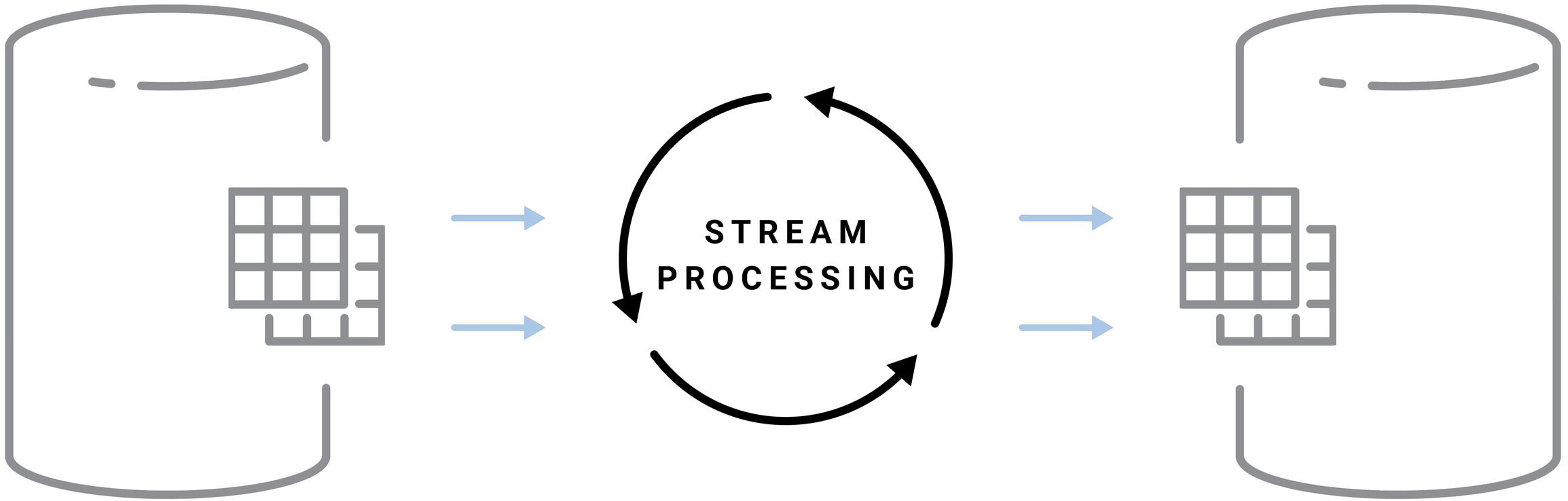
Powerful Processing incl. Filters, Transforms, Joins, Aggregations, Windowing

Supports Streams and Tables

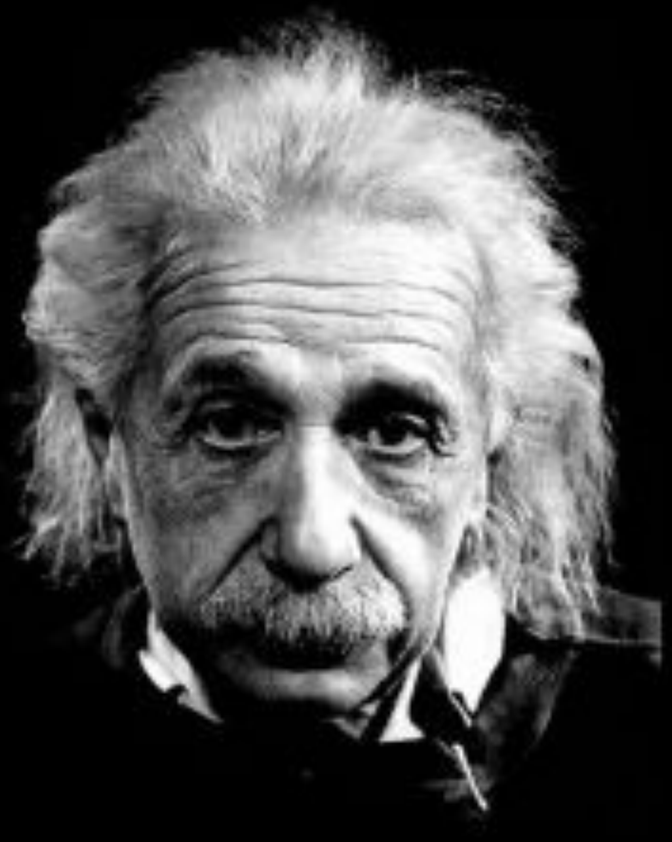
Exactly-Once Processing

Event-Time Processing

Table-Stream Duality



SPECIAL RELATIVITY



OF STREAMS AND TABLES

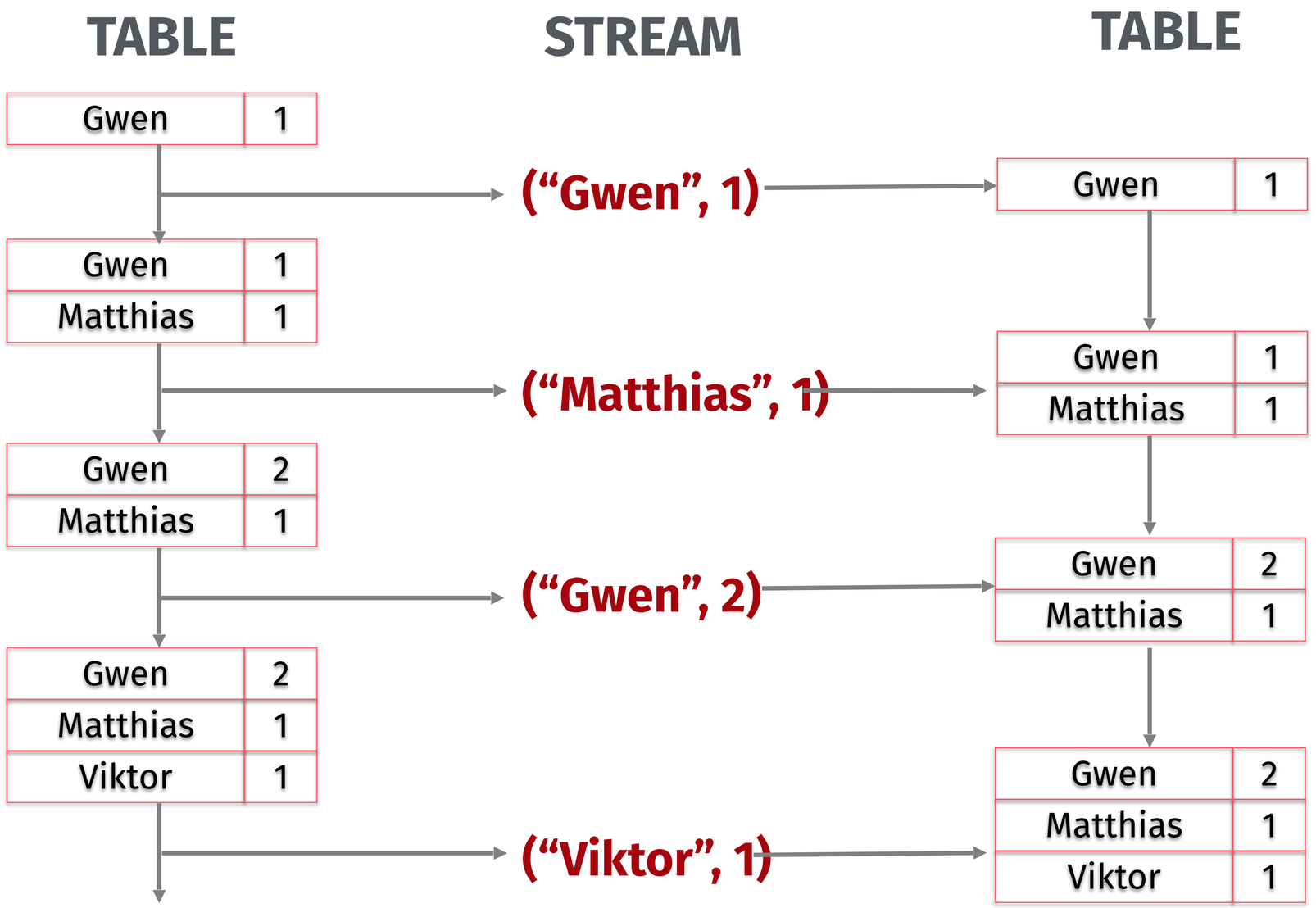
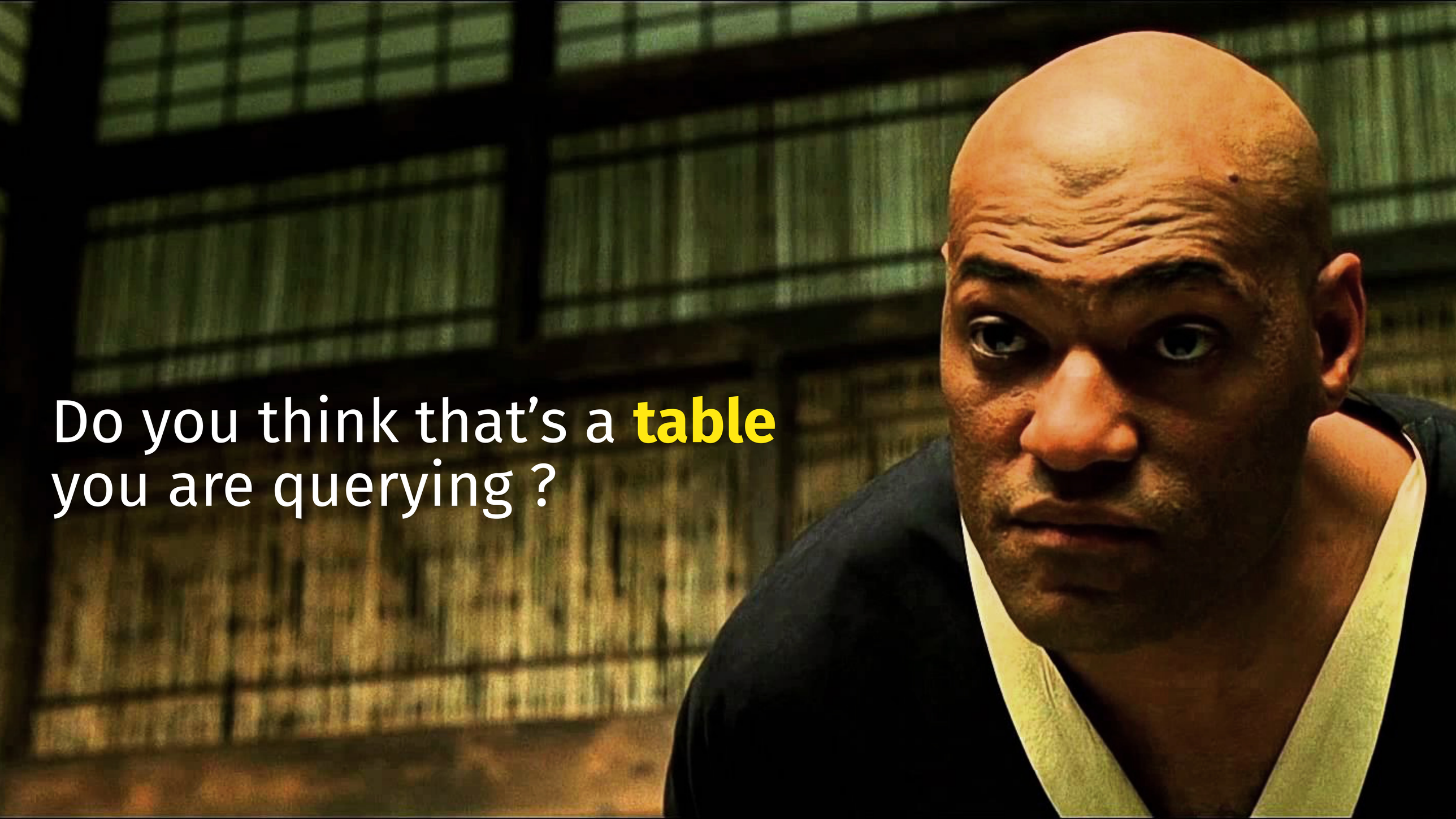


TABLE
STREAM
TABLE
Gwen 1
("Gwen", 1)
Gwen 1
Matthias 1
("Matthias", 1)
Gwen 2
Matthias 1
("Gwen", 2)
Gwen 2
Matthias 1
Viktor 1
("Viktor", 1)


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

confluent

Talk is cheap! Show me code!

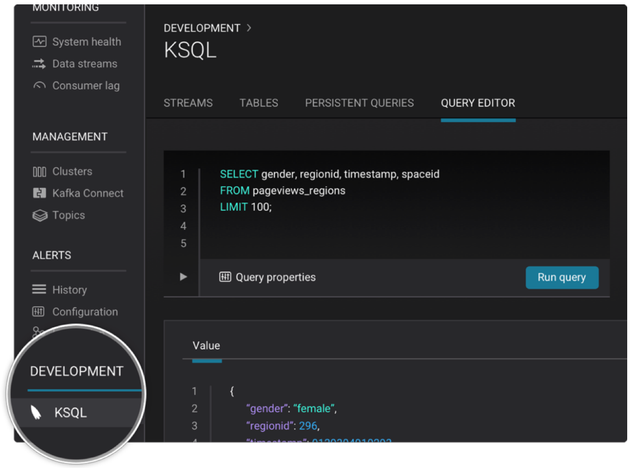
confluent

What’s next?

confluent

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

KSQL #FTW



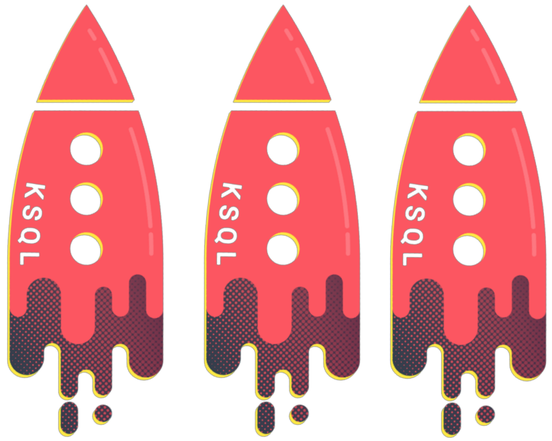

```
ksql>
```

```
POST /query
```

1. UI
2. CLI
3. REST
4. Headless

Interaction with Kafka

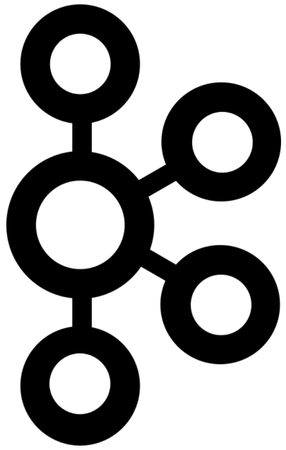
KSQL
(processing)



read,
write

network

Kafka
(data)



read,
write

network

JVM application
with Kafka Streams (processing)

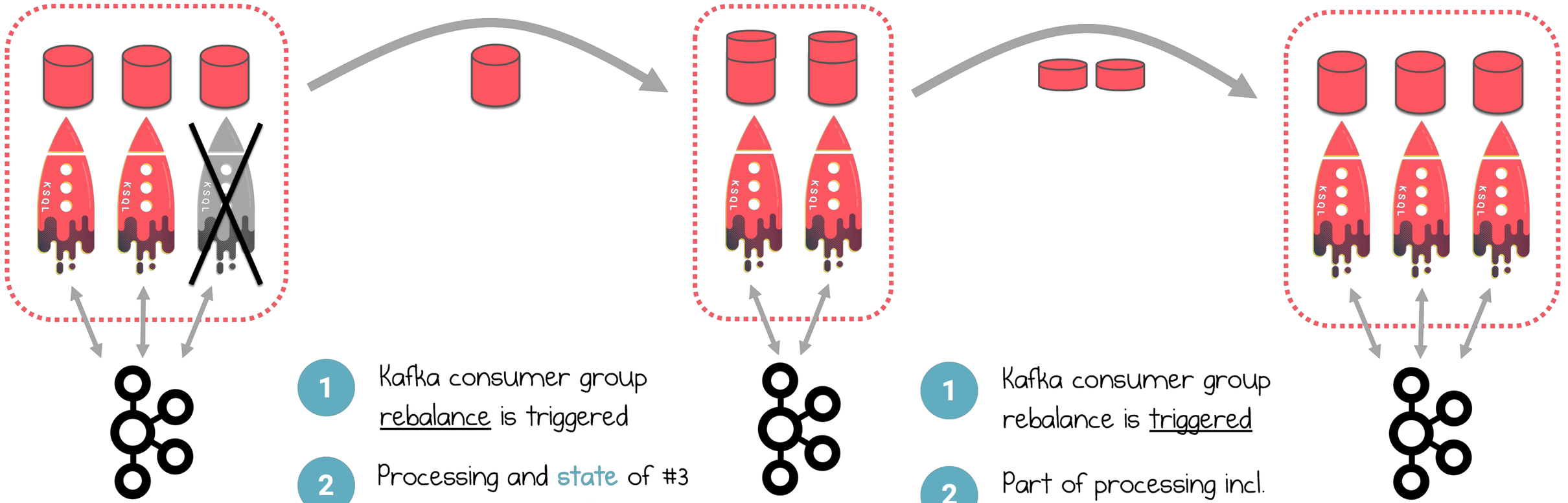
Does not run on
Kafka brokers

Does not run on
Kafka brokers

Fault-Tolerance, powered by Kafka

#3 died so #1 and #2 take over

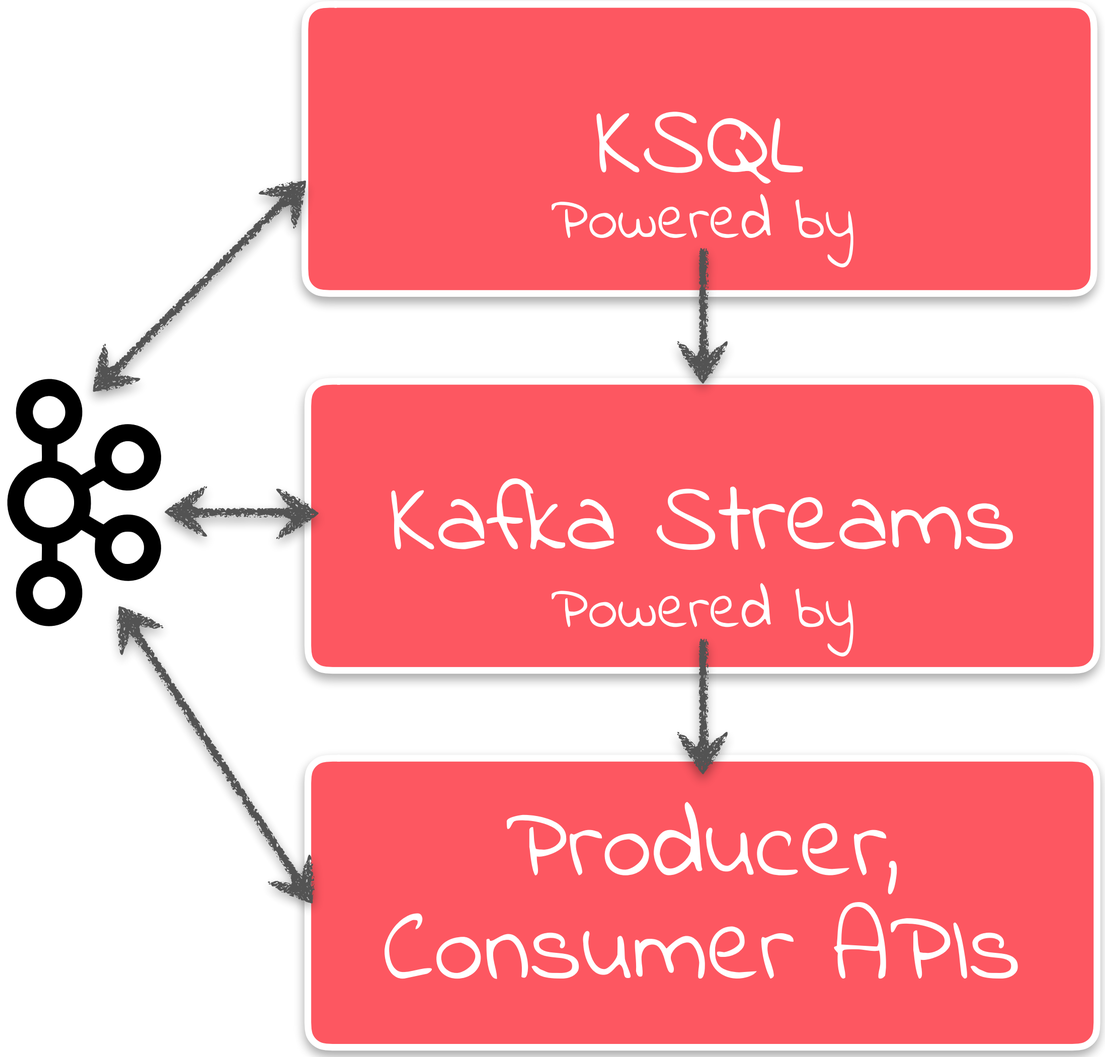
#3 is back so the work is split again



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

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



Ease of use

Flexibility

```
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/