ETL IS DEAD; LONG-LIVE STREAMS

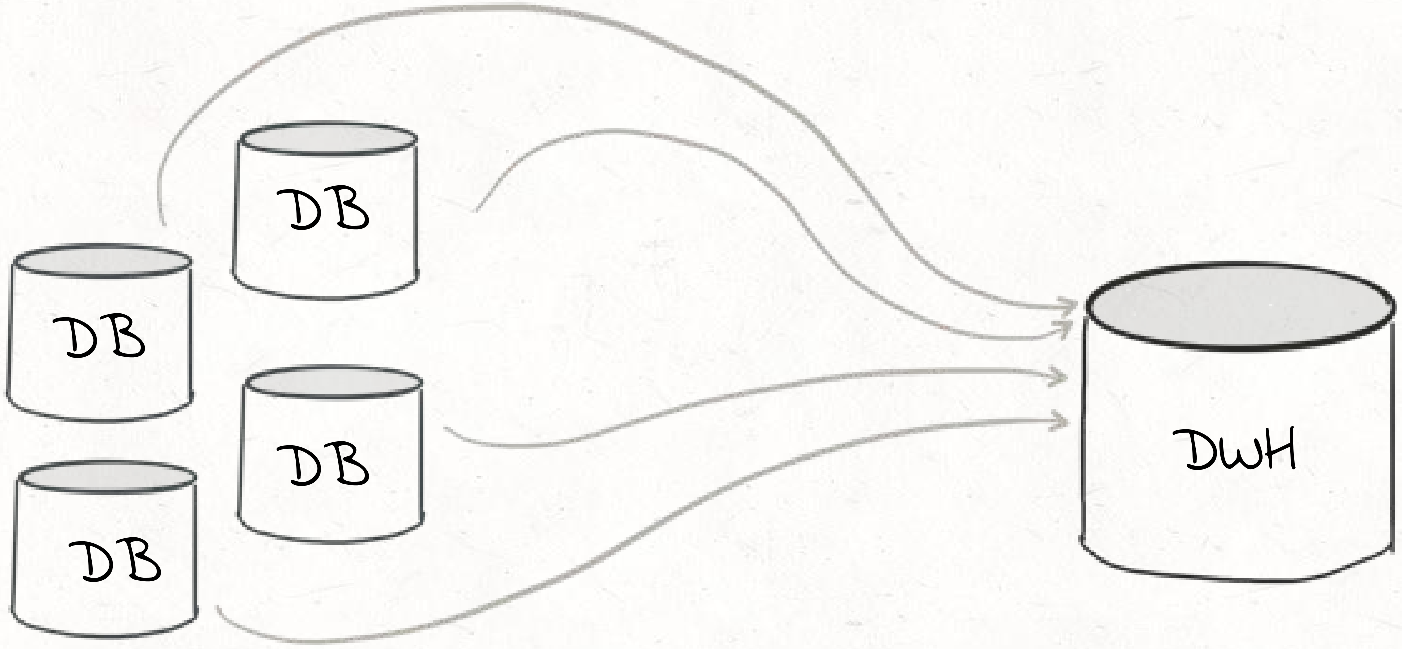
Neha Narkhede,
Co-founder & CTO, Confluent

"

Data and data systems have really changed in the past decade

confluent

Old world: Two popular locations for data



Operational databases

Relational data warehouse

confluent

"

Several recent data trends are driving a dramatic change in the ETL architecture

confluent

"

#1: Single-server databases are replaced by a myriad of **distributed** data platforms that operate at company-wide scale

"

#2: Many more types of data sources beyond transactional data - logs, sensors, metrics...

confluent

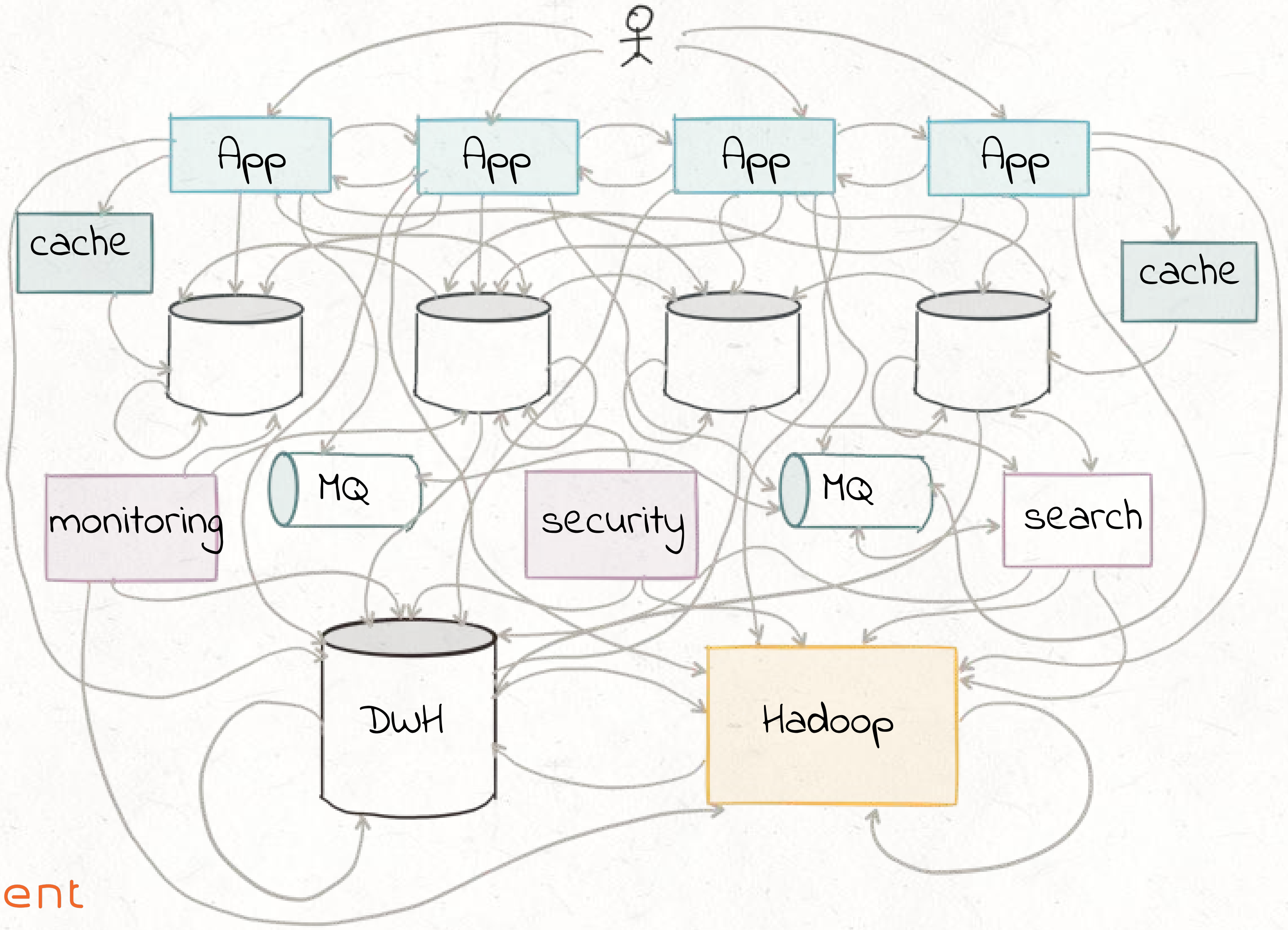
"

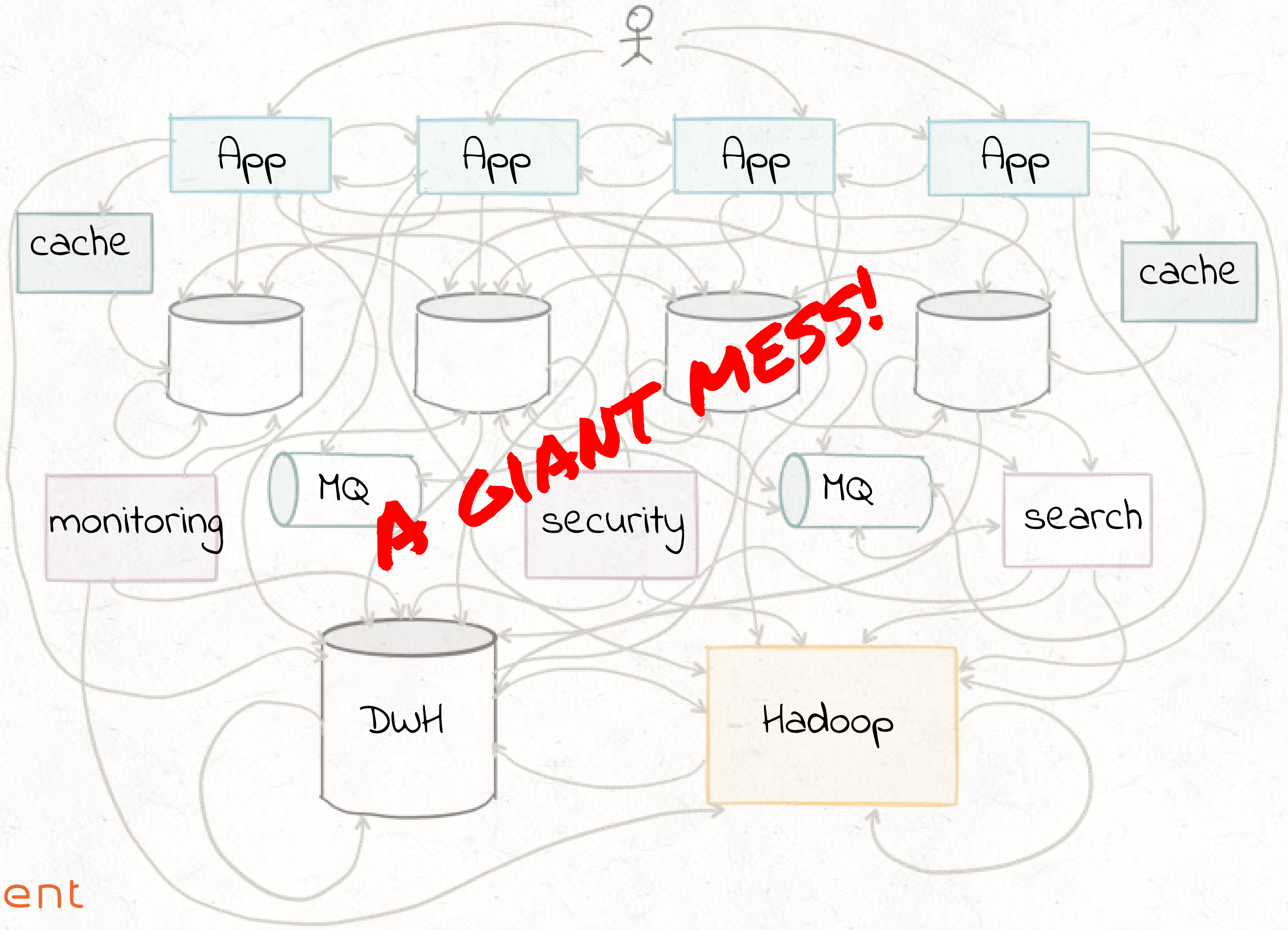
#3: Stream data is increasingly ubiquitous; need for faster processing than daily

confluent

"

The end result? This is what data integration ends up looking like in practice


App
App
App
App
cache
cache
MQ
MQ
monitoring
security
search
DWH
Hadoop

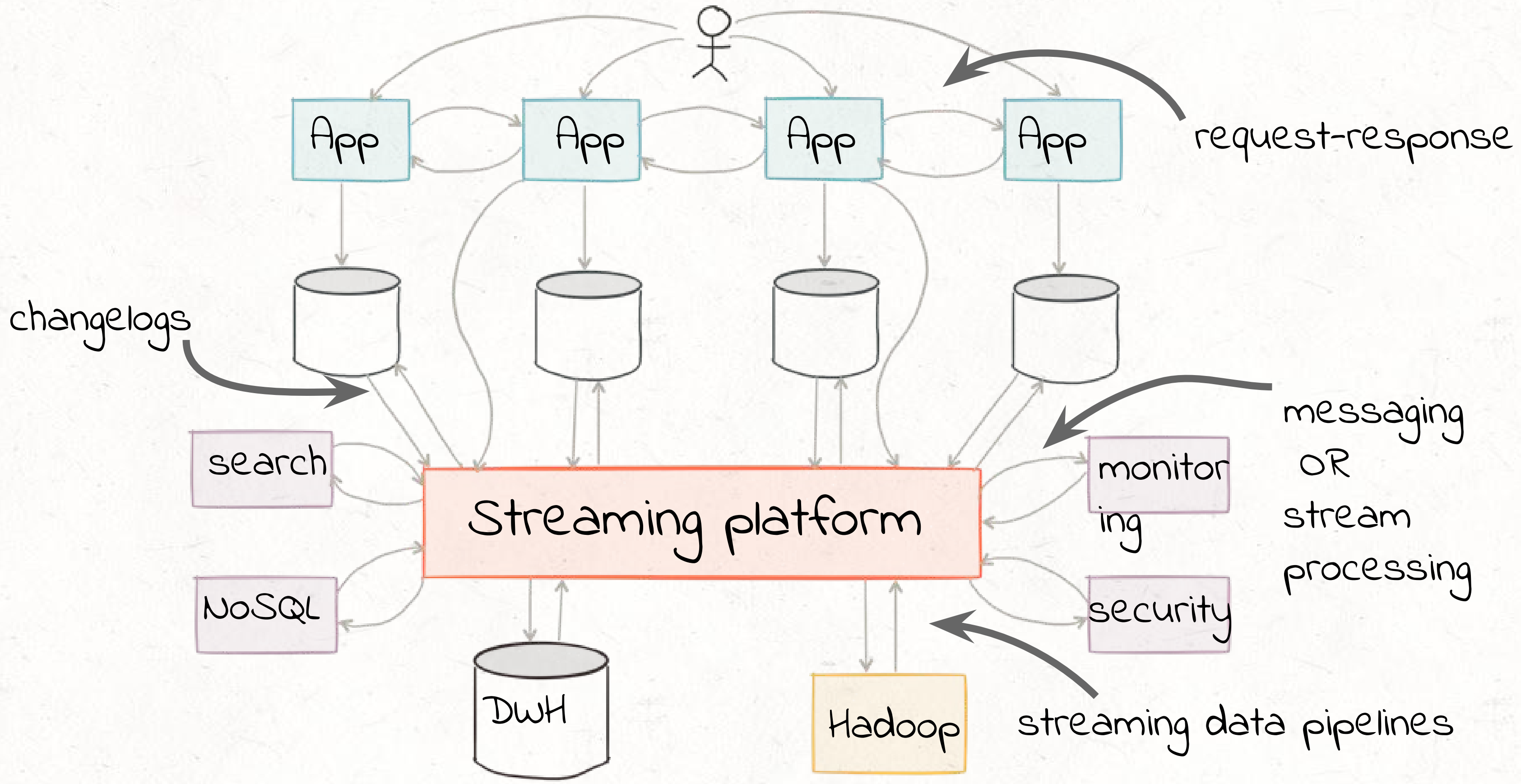

App
App
App
App
cache
cache
MQ
MQ
monitoring
security
search
DWH
Hadoop
A GIANT MESS!

confluent

"

we will see how transitioning to streams cleans up this mess and works towards...

confluent


request-response
App
App
App
App
changelogs
search
Streaming platform
monitoring
messaging
OR
stream
processing
NoSQL
security
DWH
Hadoop
streaming data pipelines

confluent

A SHORT HISTORY OF DATA INTEGRATION

"

Surfaced in the 1990s in retail organizations for analyzing buyer trends

confluent

"

Extract data from databases

Transform into destination warehouse schema

Load into a central data warehouse

"

BUT … ETL tools have been around for a long time, **data coverage** in data warehouses is still **low**! WHY?

ETL has drawbacks

"

#1: The need for a global schema

"

#2: Data cleansing and curation is manual and fundamentally **error-prone**

confluent

"

#3: Operational cost of ETL is high; it is slow; time and **resource intensive**

"

#4: **ETL** tools were built to narrowly focus on connecting databases and the data warehouse in a **batch** fashion

"

Early take on real-time ETL

=

Enterprise Application Integration (EAI)

confluent

"

EAI: A different class of data integration technology for connecting applications in real-time

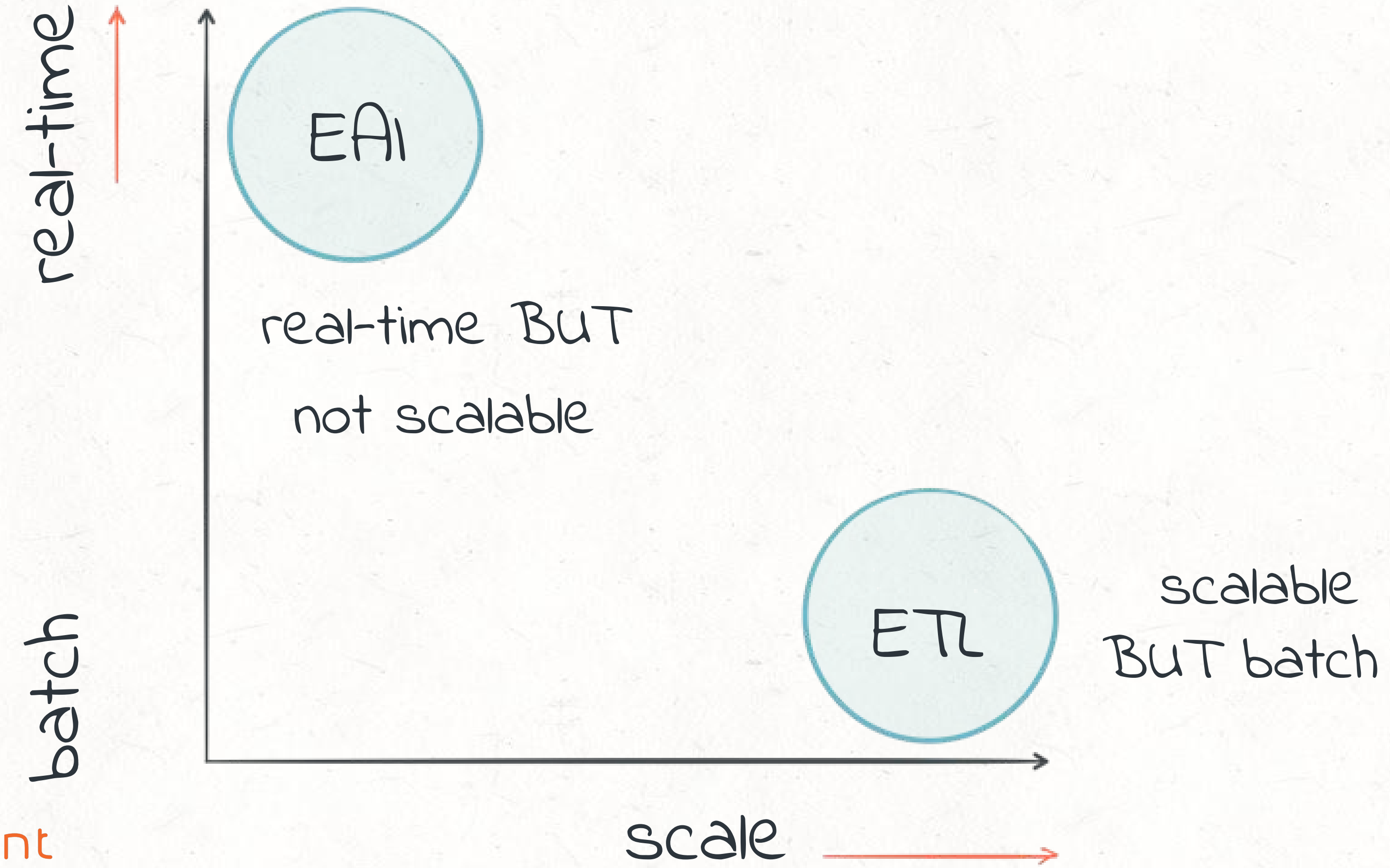
confluent

"

EAI employed Enterprise Service Buses and MQs; weren't scalable

ETL AND EAI ARE OUTDATED!

OLD WORLD: SCALE OR TIMELY DATA, PICK ONE



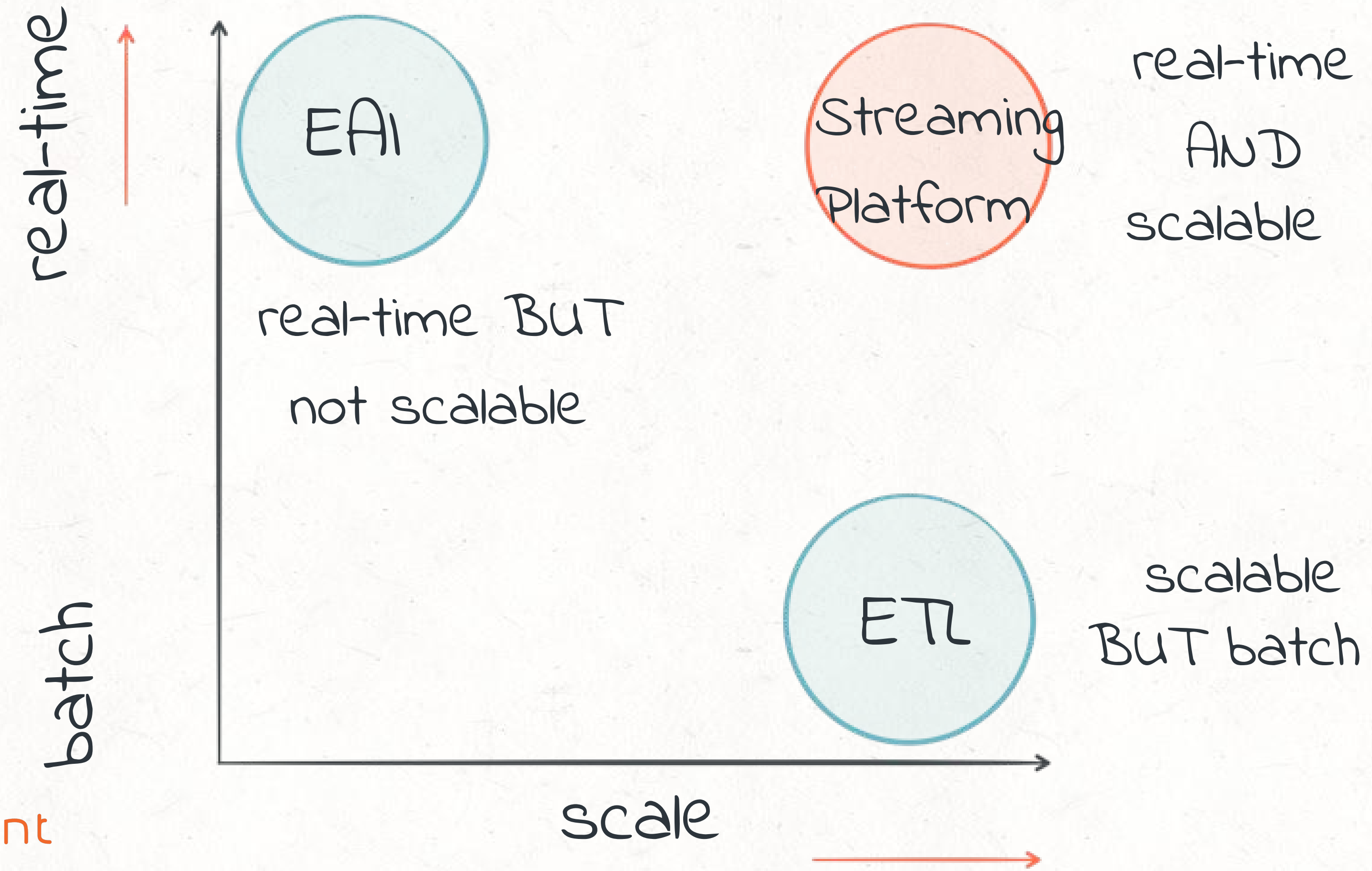
confluent

"

Data integration and ETL in the modern world need a complete revamp

confluent

NEW WORLD: STREAMING, REAL-TIME AND SCALABLE



"

Modern streaming world has new set of **requirements** for data integration

"

#1: Ability to process high-volume and high-diversity data

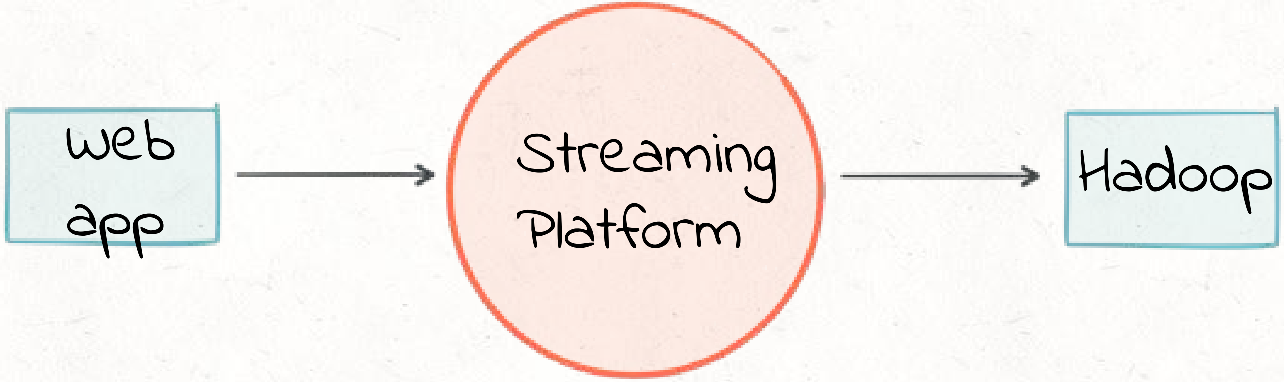
confluent

"

#2 Real-time from the grounds up; a fundamental transition to event-centric thinking

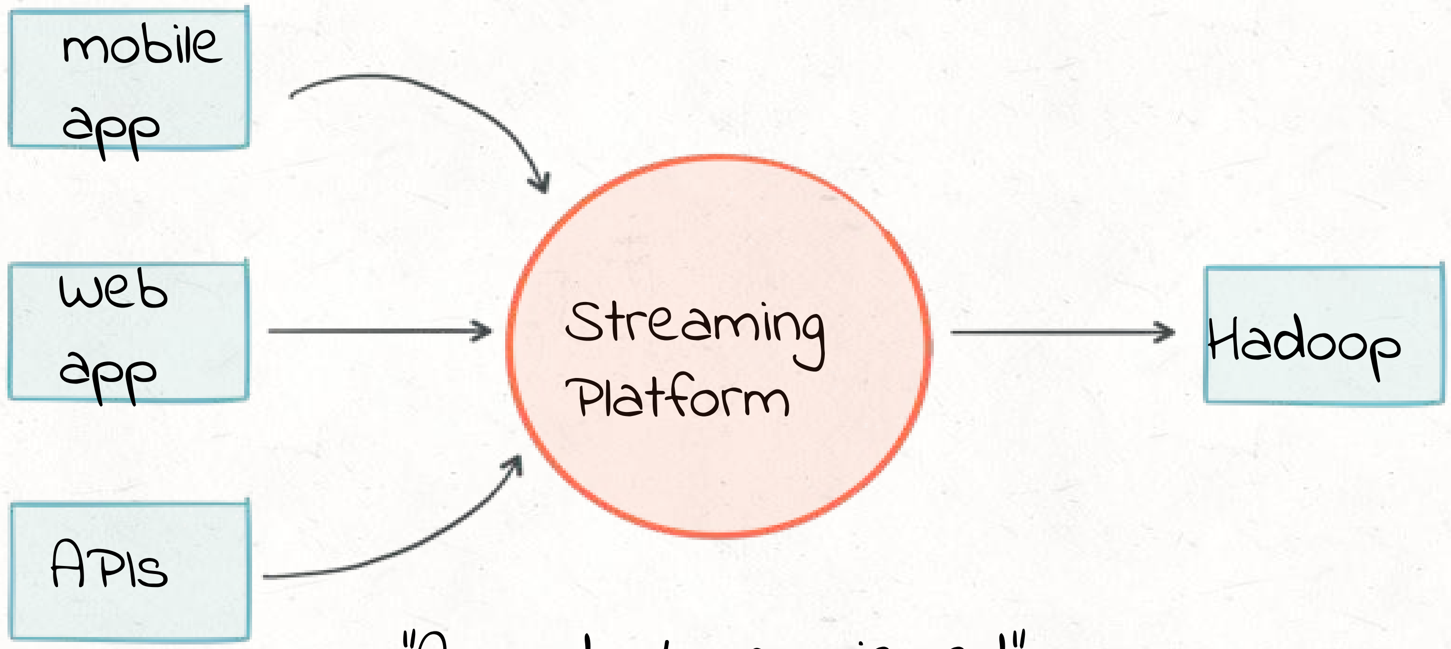
confluent

Event-Centric Thinking



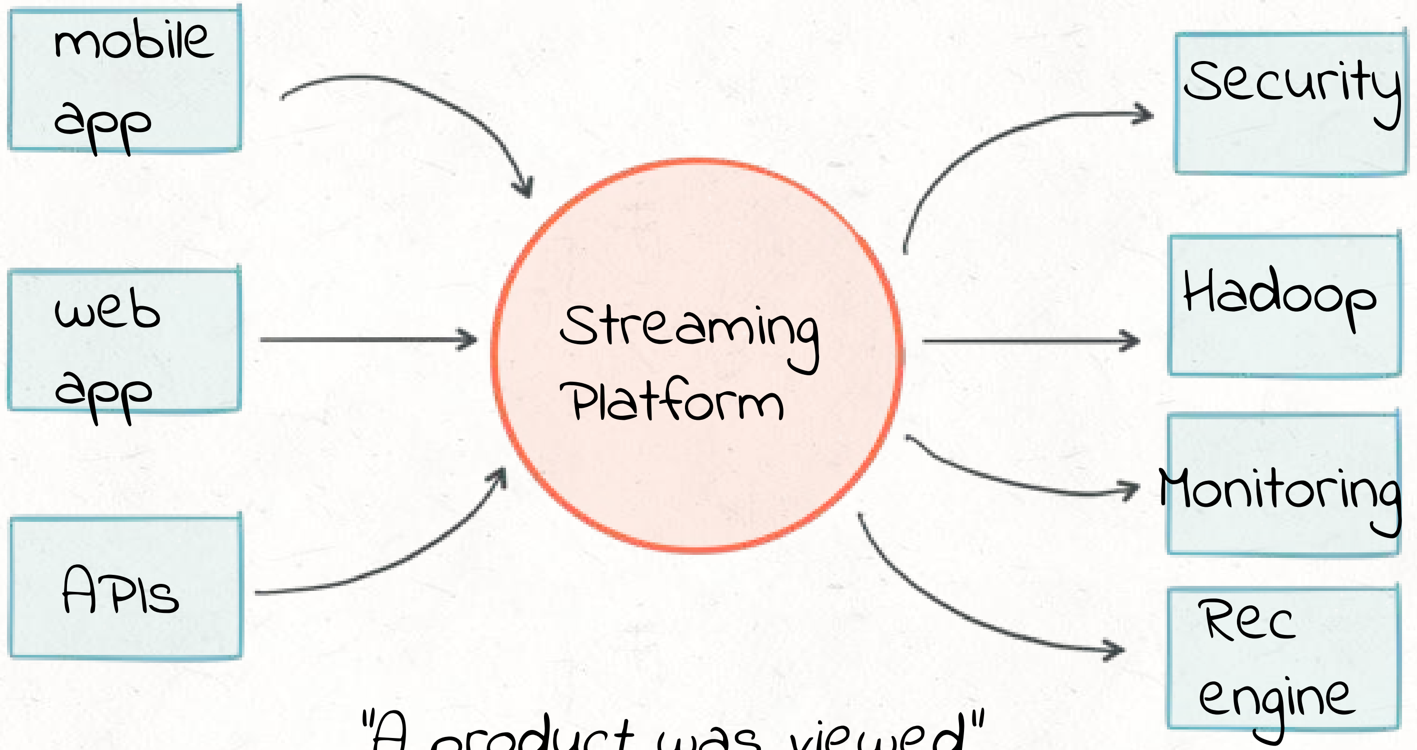
"A product was viewed"

Event-Centric Thinking



"A product was viewed"

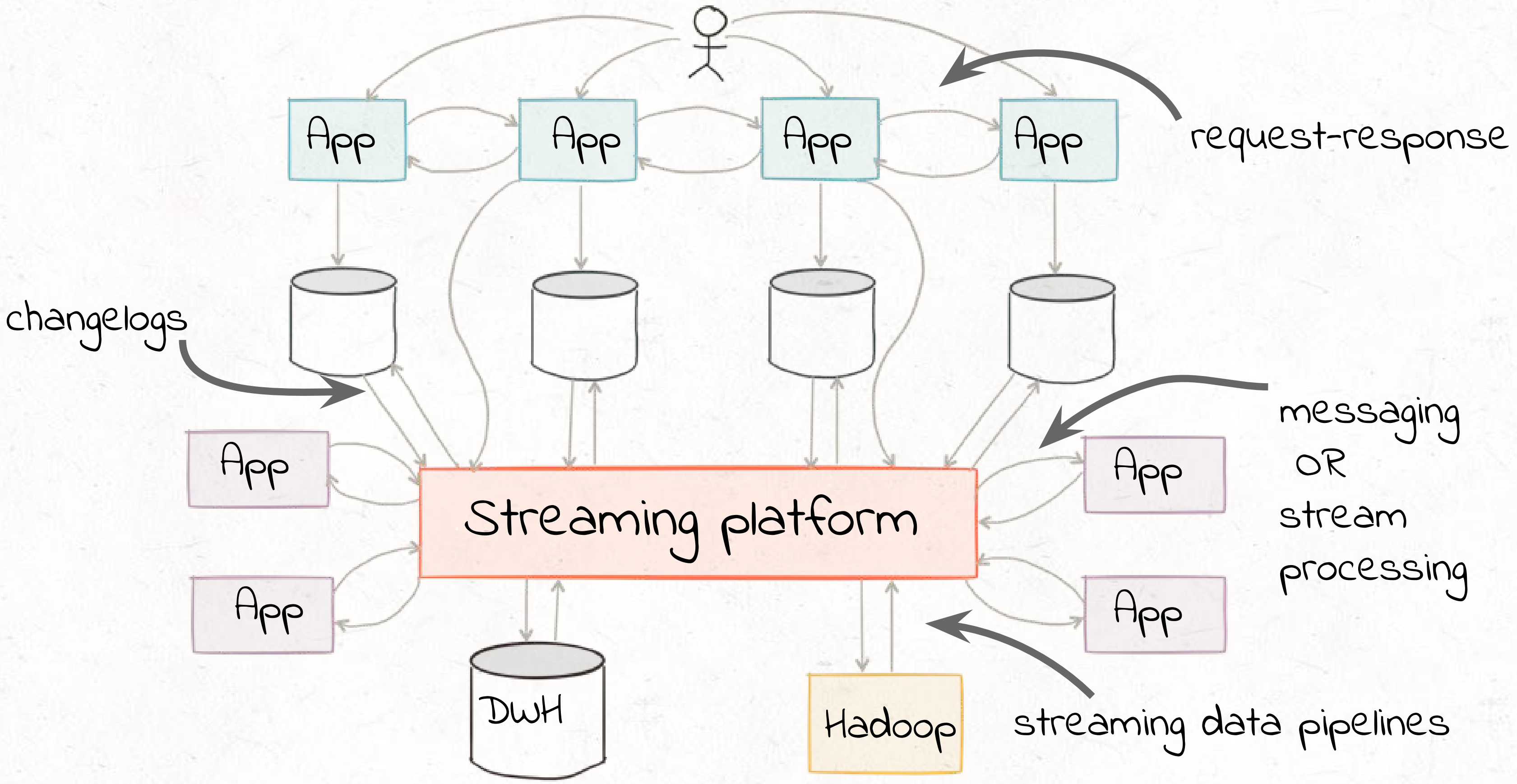
Event-Centric Thinking



"A product was viewed"

"

Event-centric thinking, when applied at a company-wide scale, leads to this simplification ...


App
App
App
App
request-response
changelogs
App
Streaming platform
App
messaging
OR
stream
processing
App
App
DWH
Hadoop
streaming data pipelines

confluent

"

#3: Enable forward-compatible data architecture; the ability to add more applications that need to process the same data … differently

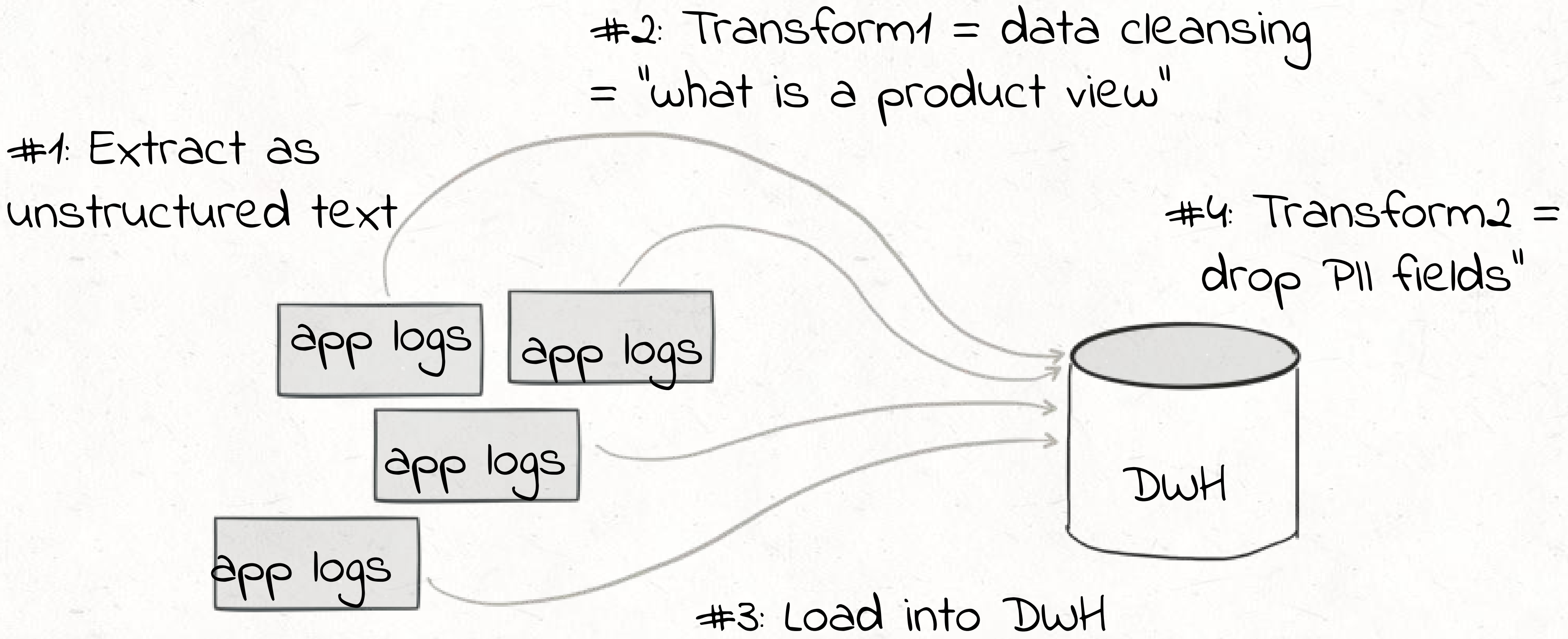
confluent

"

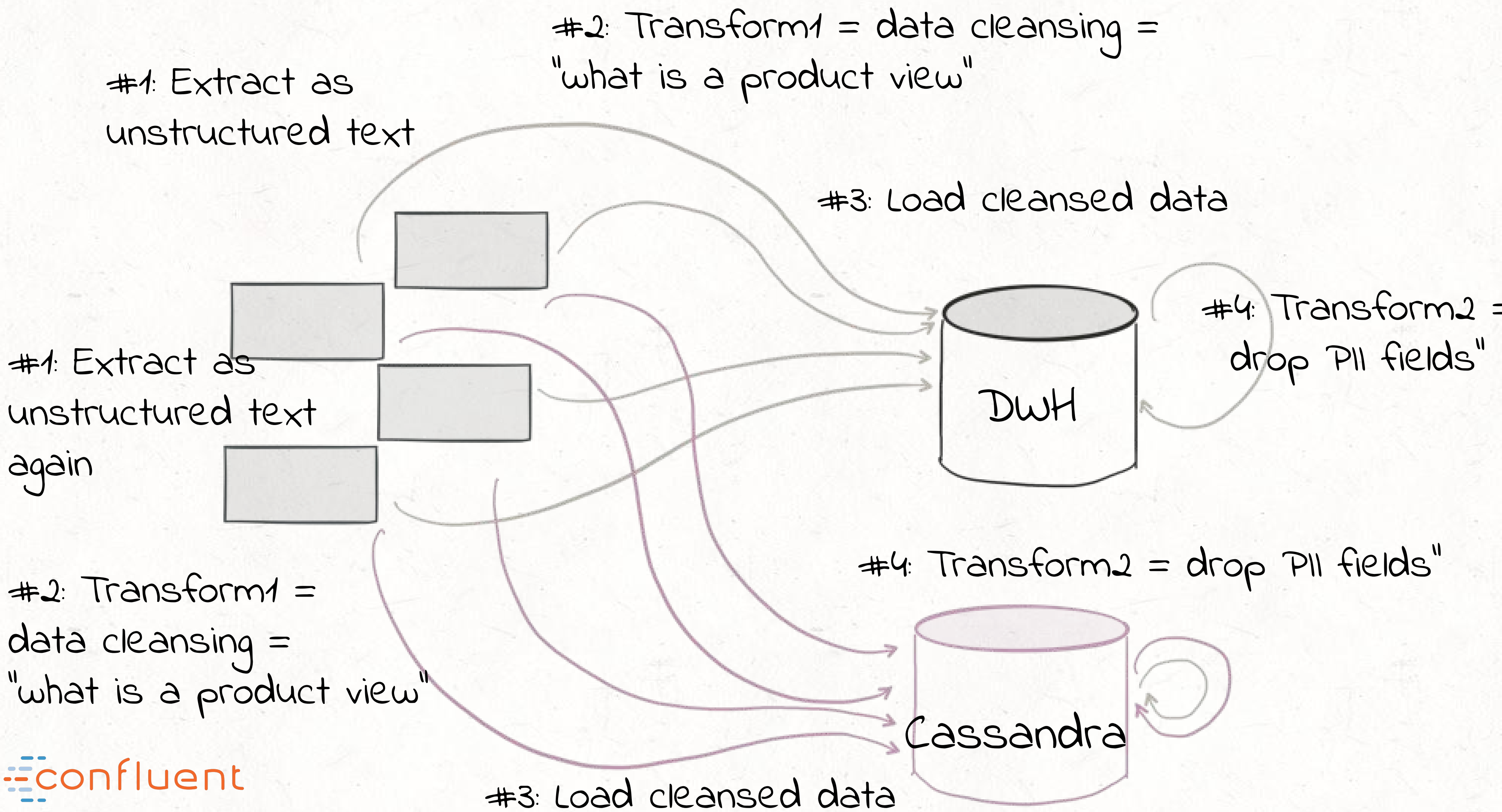
To enable forward compatibility, redefine the T in ETL:

Clean data in; Clean data out

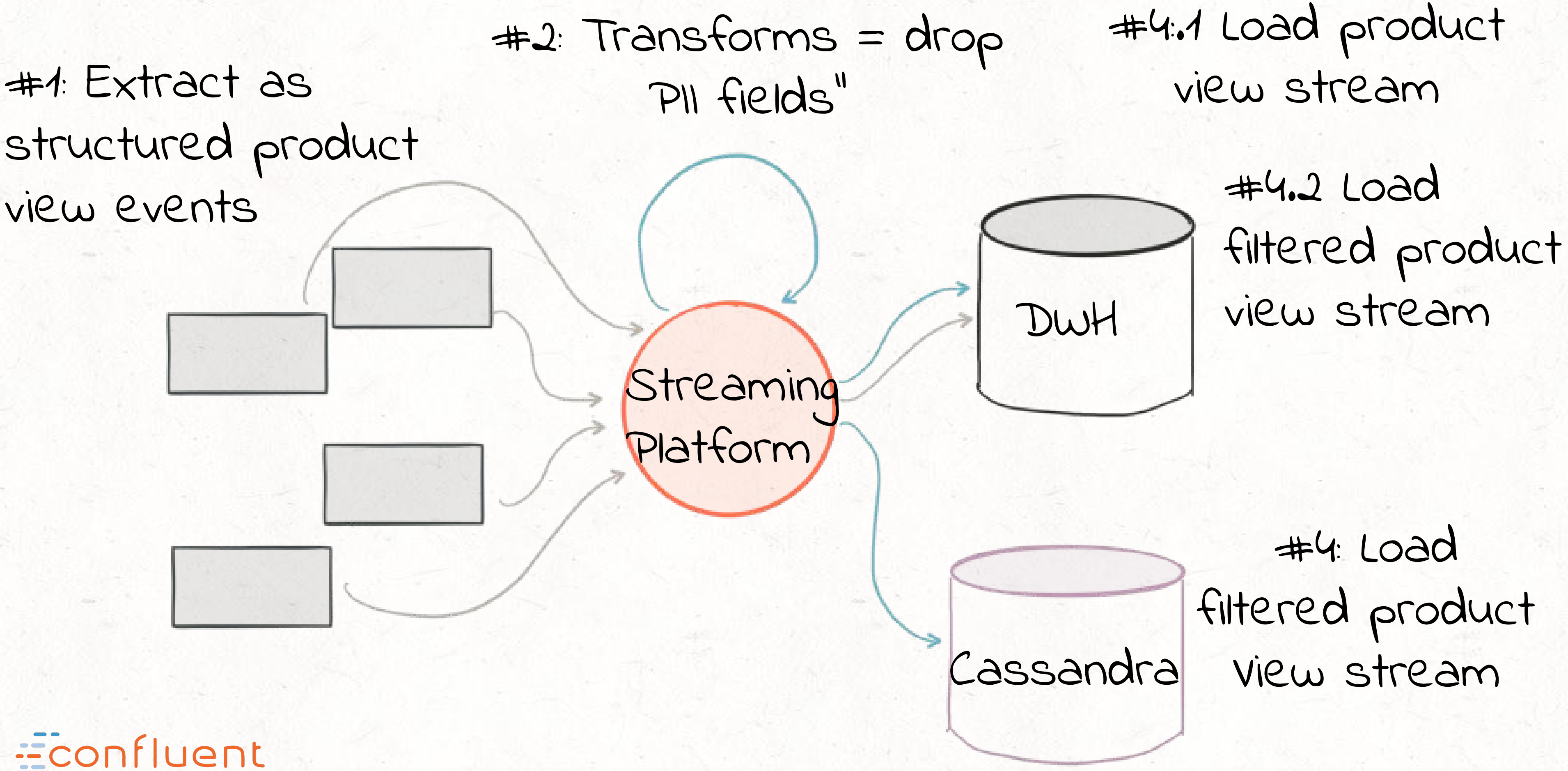
confluent


#2: Transform1 = data cleansing
= "what is a product view"
#1: Extract as
unstructured text
#4: Transform2 =
drop PII fields"
app logs
app logs
app logs
app logs
DWH
#3: Load into DWH

confluent


#2: Transform1 = data cleansing = "what is a product view"
#1: Extract as unstructured text
#3: Load cleansed data
#4: Transform2 = drop PII fields"
DWH
#1: Extract as unstructured text again
#2: Transform1 = data cleansing = "what is a product view"
#4: Transform2 = drop PII fields"
Cassandra
#3: Load cleansed data

confluent


#1: Extract as structured product view events
#2: Transforms = drop PII fields"
#4.1 Load product view stream
#4.2 Load filtered product view stream
Streaming Platform
DWH
#4: Load filtered product view stream
Cassandra

confluent

"

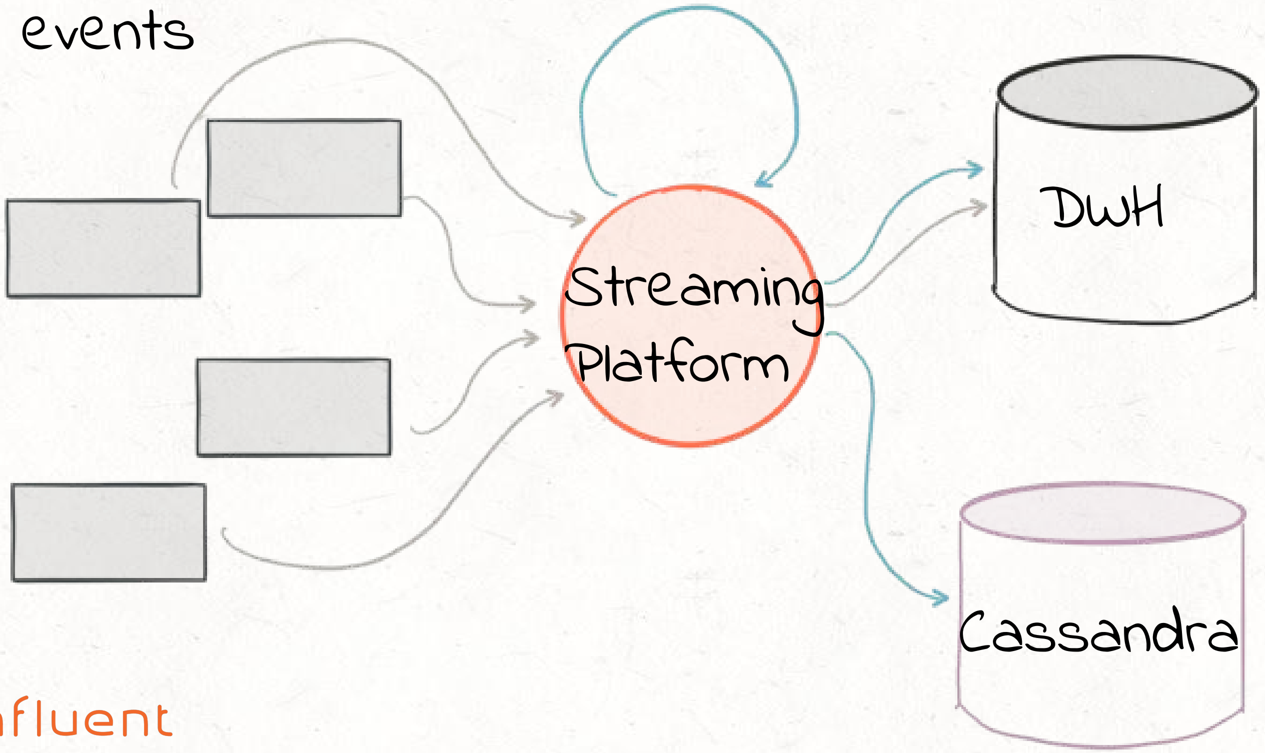
To enable forward compatibility, redefine the T in ETL:
Data transformations, not data cleansing!

#1: Extract **once** as structured product view events

#2: Transform **once** = drop PII fields" and enrich with product metadata

#4.1: Load product views stream

#4.2: Load filtered and enriched product views stream



#4: Load filtered and enriched product views stream

confluent

"

Forward compatibility =
Extract clean-data once; Transform many different ways before Loading into respective destinations … as and when required

"

In summary, needs of modern data integration solution?
Scale, diversity, latency and forward compatibility

Requirements for a modern streaming data integration solution

- Fault tolerance
- Parallelism
- Latency
- Delivery semantics
- Operations and monitoring
- Schema management

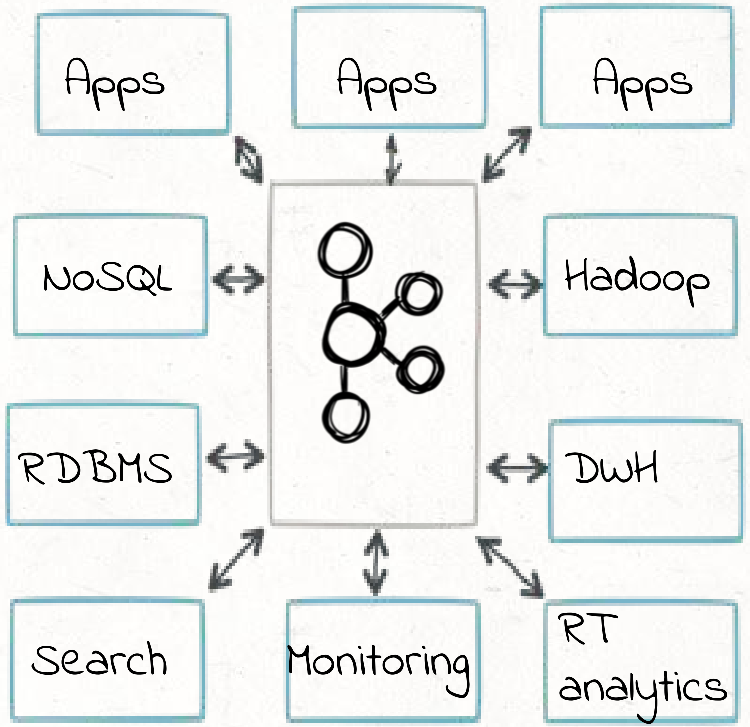
confluent

Data integration: Platform vs tool

Central, reusable infrastructure for many use cases

One-off, non-reusable solution for a particular use case

New shiny future of ETL: a streaming platform



"

Streaming platform serves as the central nervous system for a company's data in the following ways ...

"#1: Serves as the **real-time**, scalable **messaging bus** for applications; no EAI

"

#2: Serves as the **source-of-truth** pipeline for feeding all data processing destinations; Hadoop, DWH, NoSQL systems and more

"

#3: Serves as the **building block** for stateful **stream processing** microservices

confluent

"

Streaming

~~Batch~~ data integration

confluent

"

Streaming

~~Batch~~ ETL

a short history of data integration

drawbacks of ETL

needs and requirements for a streaming platform

new, shiny future of ETL: a streaming platform

what does a streaming platform look like and how it enables Streaming ETL?

confluent

Apache Kafka: A Distributed Streaming Platform

confluent

Apache kafka 6 years ago

> 1,400,000,000,000
messages processed / day

Now Adopted at 1000s of companies worldwide

NETFLIX
intuit
Pinterest
Goldman Sachs
box
ebay
airbnb
Adobe
cisco
salesforce
yelp
Square
dish NETWORK
Cerner
WIKIPEDIA
The Free Encyclopedia
Y!
PayPal
verizon

confluent

"

what role does Kafka play in the new shiny future for data integration?

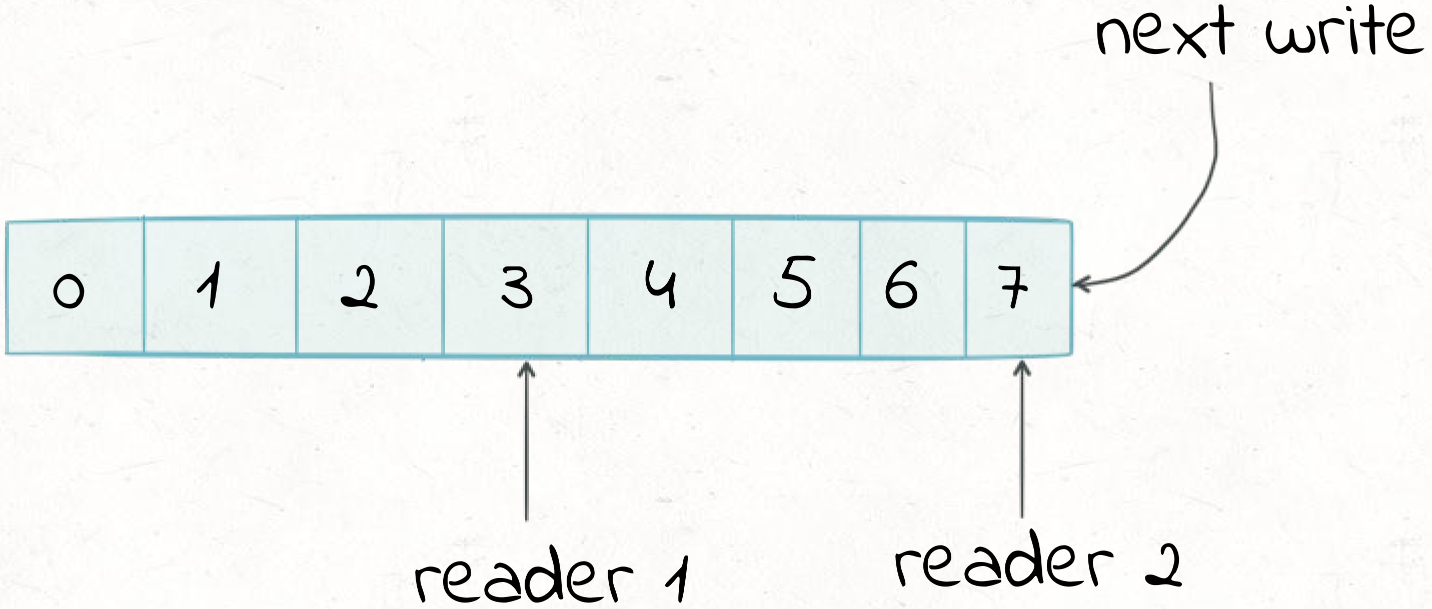
confluent

"

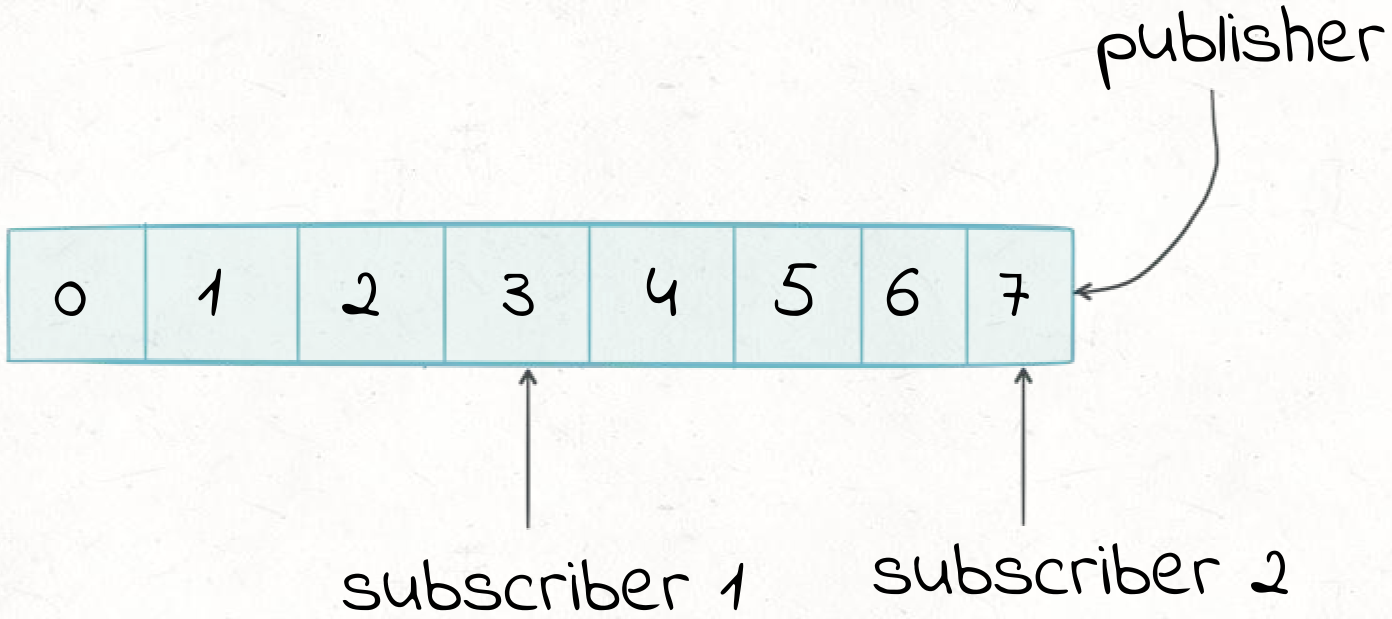
#1: Kafka is the de-facto storage of choice for stream data

confluent

THE LOG



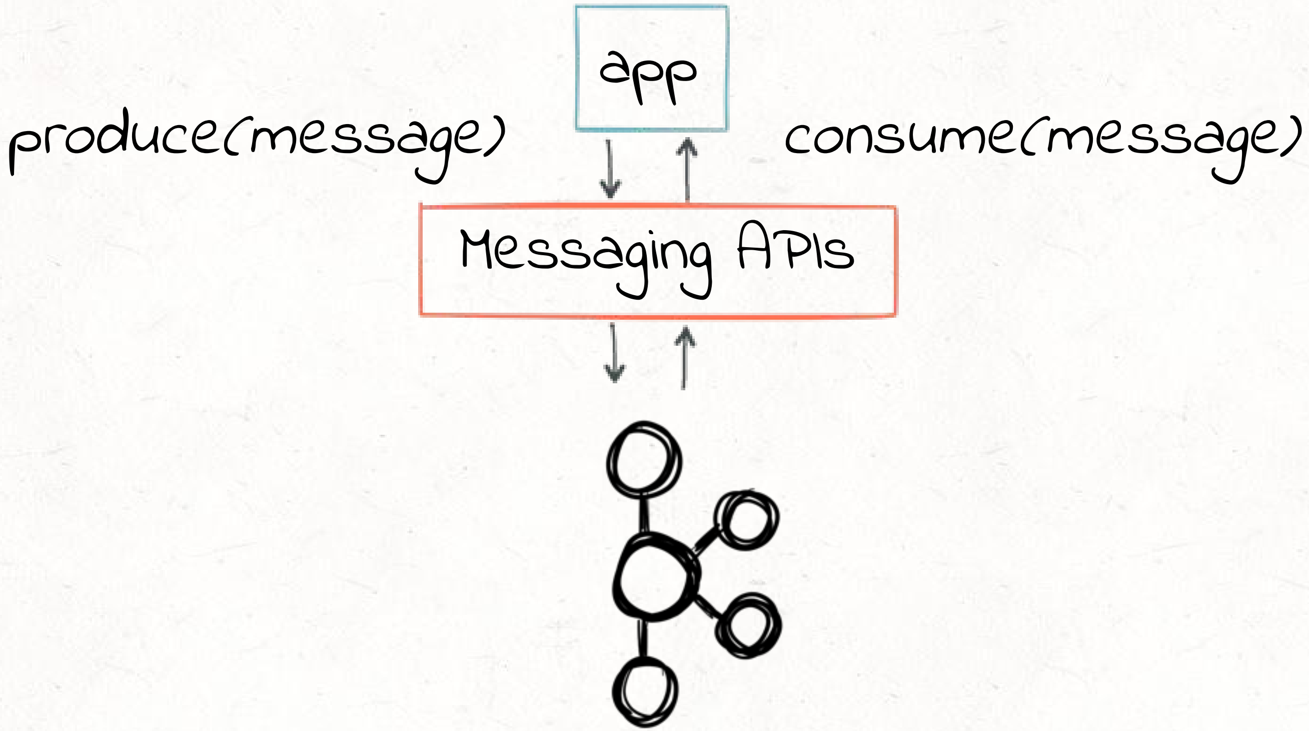
THE LOG & PUB-SUB



"

#2: Kafka offers a scalable **messaging** backbone for application integration

Kafka messaging APIs: scalable EAI

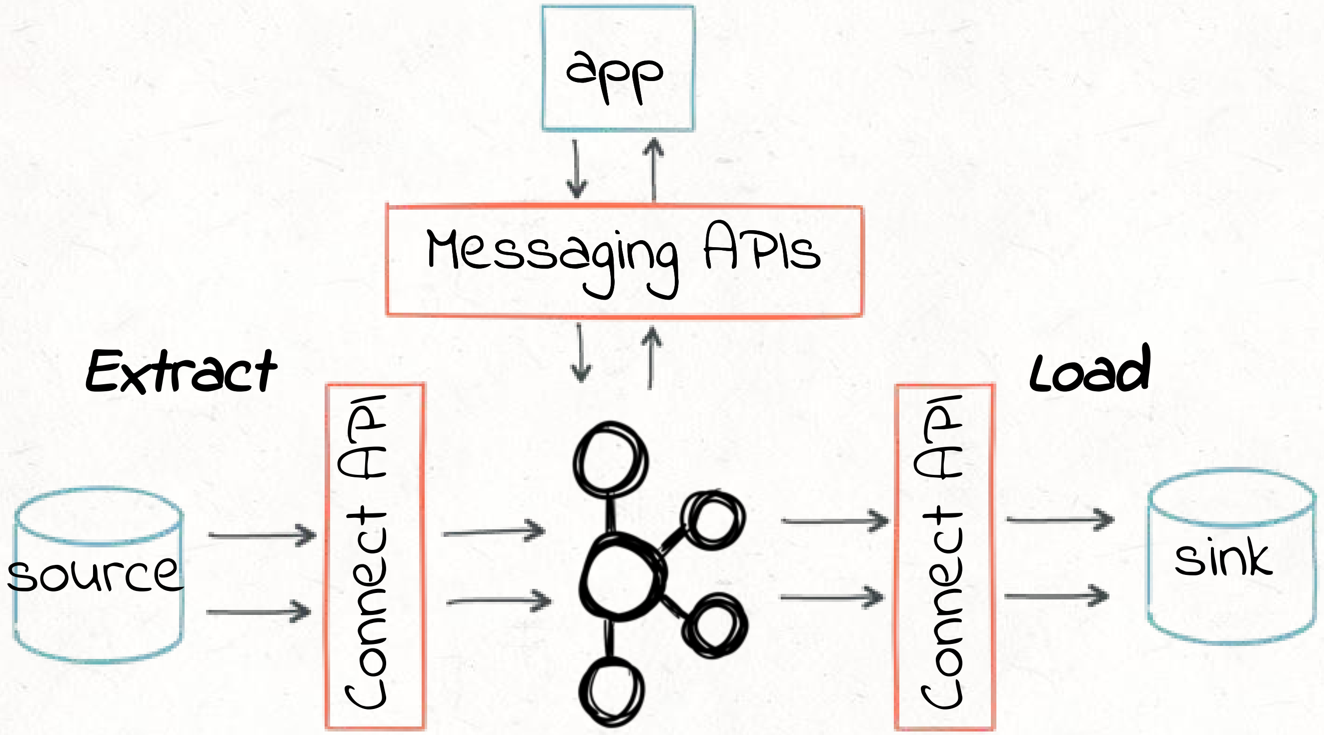


"

#3: Kafka enables building streaming data pipelines (E & L in ETL)

confluent

Kafka's Connect API: Streaming data ingestion

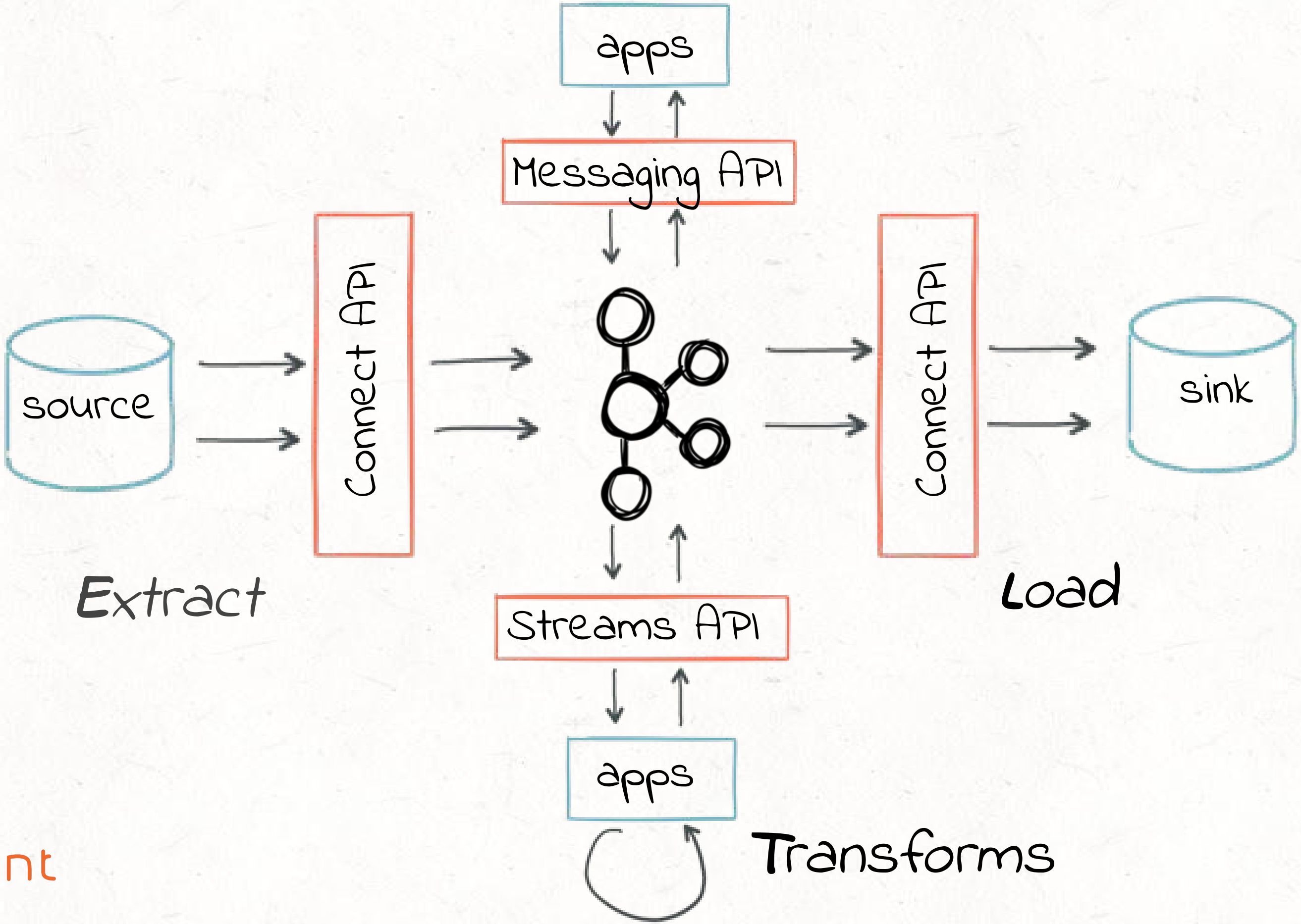


"

#4: Kafka is the basis for stream processing and transformations

confluent

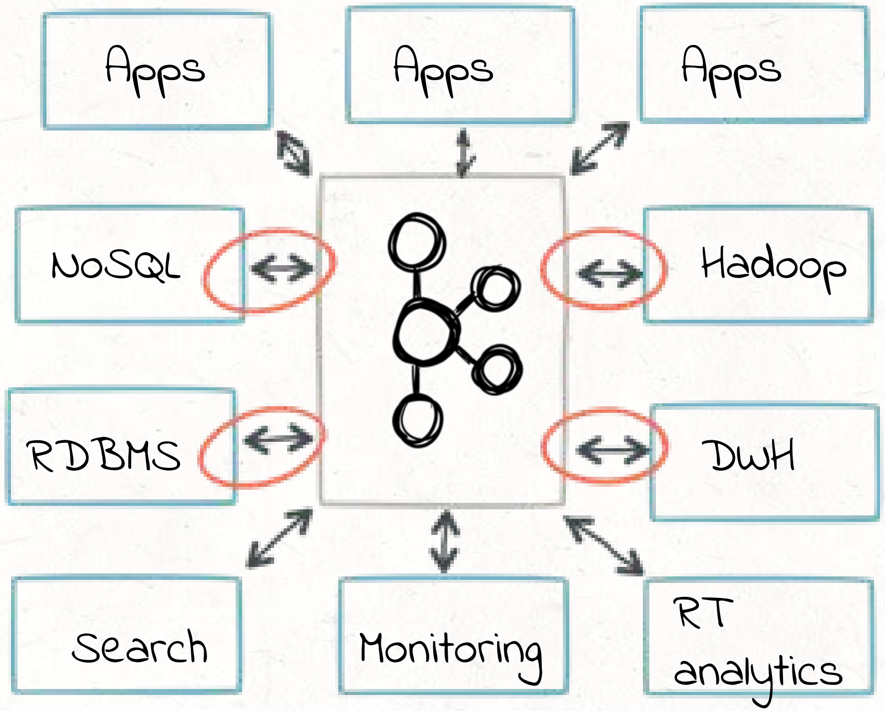
Kafka's Streams API: stream processing (transforms)



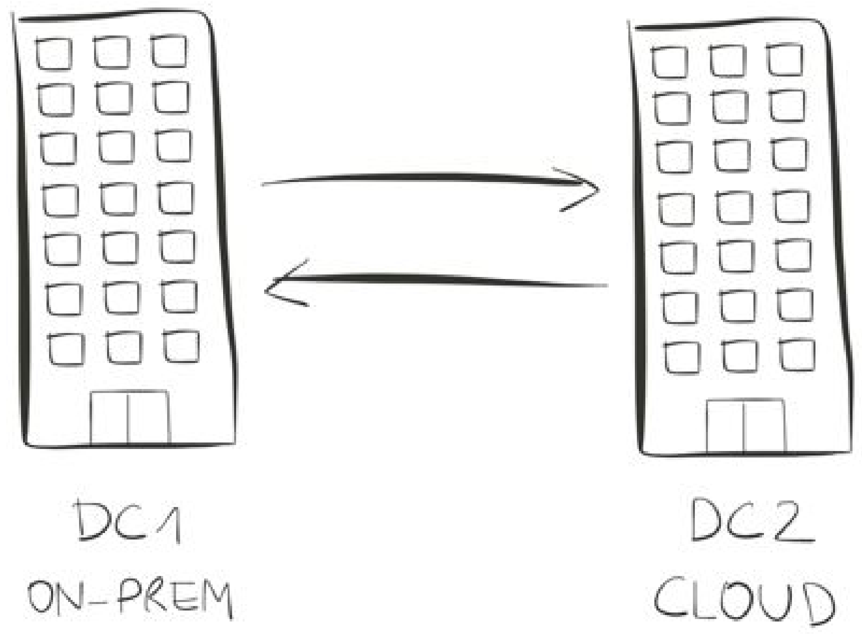
confluent

Kafka's connect API
=
E and L in Streaming ETL

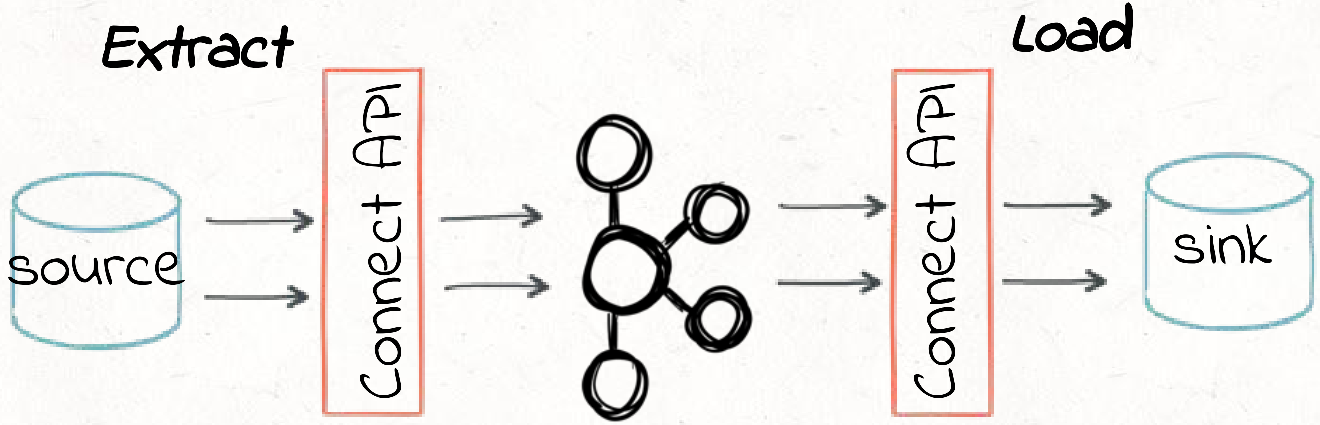
Connectors!


Apps
Apps
Apps
NoSQL
Hadoop
RDBMS
DWH
Search
Monitoring
RT analytics

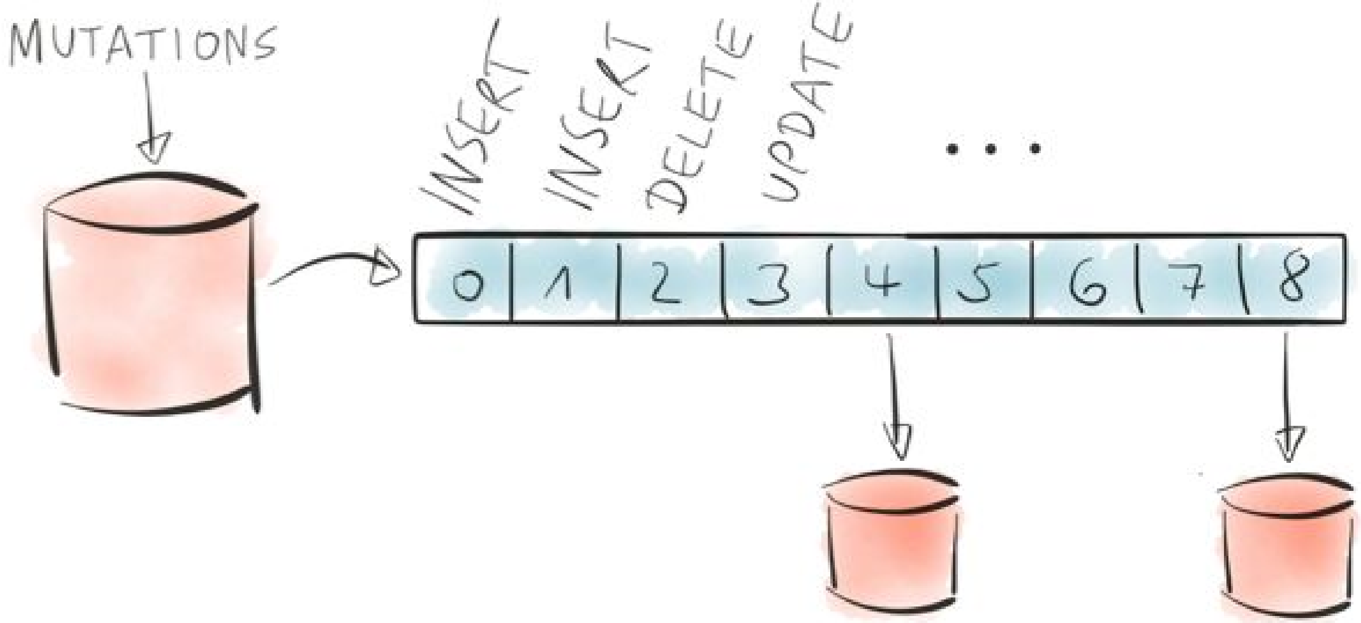
How to keep data centers in-sync?



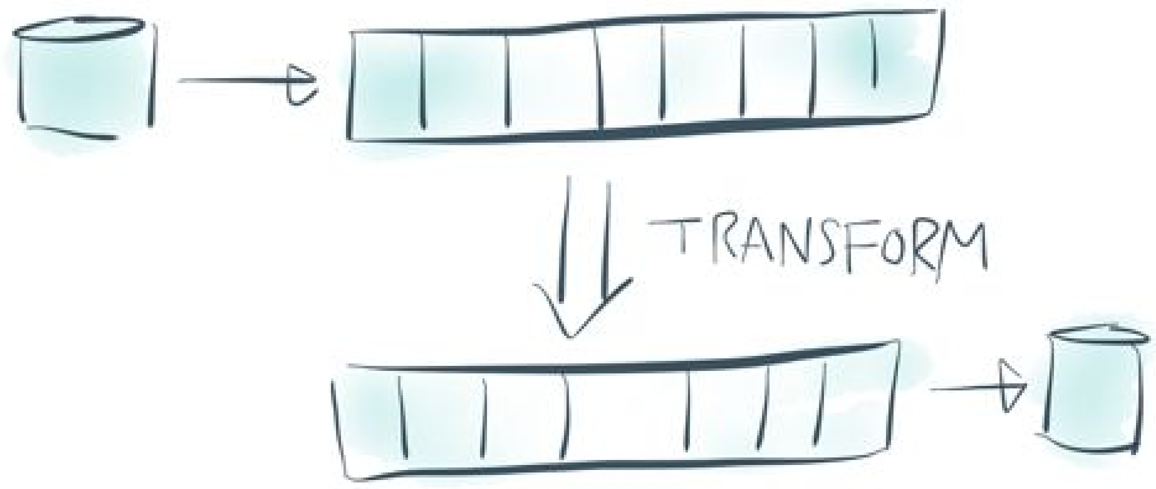
Sources and sinks



CHANGELOGS



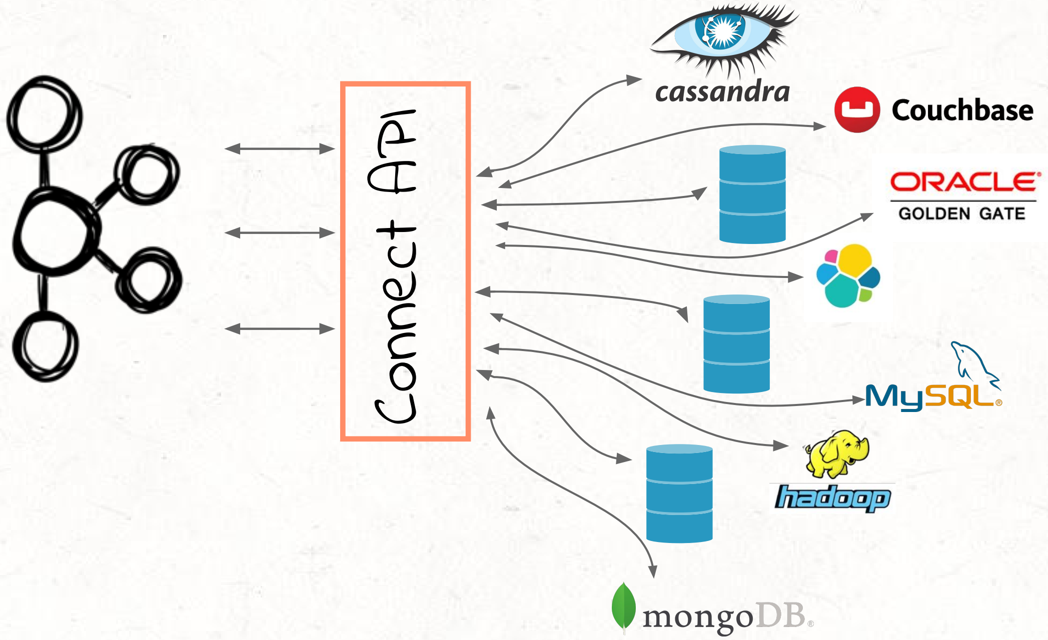
Transforming changelogs



Kafka's Connect API = Connectors Made Easy!

- **Scalability**: Leverages Kafka for scalability
- **Fault tolerance**: Builds on Kafka's fault tolerance model
- **Management and monitoring**: One way of monitoring all connectors
- **Schemas**: Offers an option for preserving schemas from source to sink

Kafka all the things!



confluent

Kafka's Streams API = The T in Streaming ETL

confluent

"

Stream processing =

transformations on stream data

2 VISIONS FOR STREAM PROCESSING

Real-time Mapreduce VS Event-driven microservices

confluent

2 VISIONS FOR STREAM PROCESSING

Real-time Mapreduce VS Event-driven microservices

- Central cluster
- Custom packaging, deployment & monitoring
- Suitable for analytics-type use cases

- Embedded library in any Java app
- Just Kafka and your app
- Makes stream processing accessible to any use case

confluent

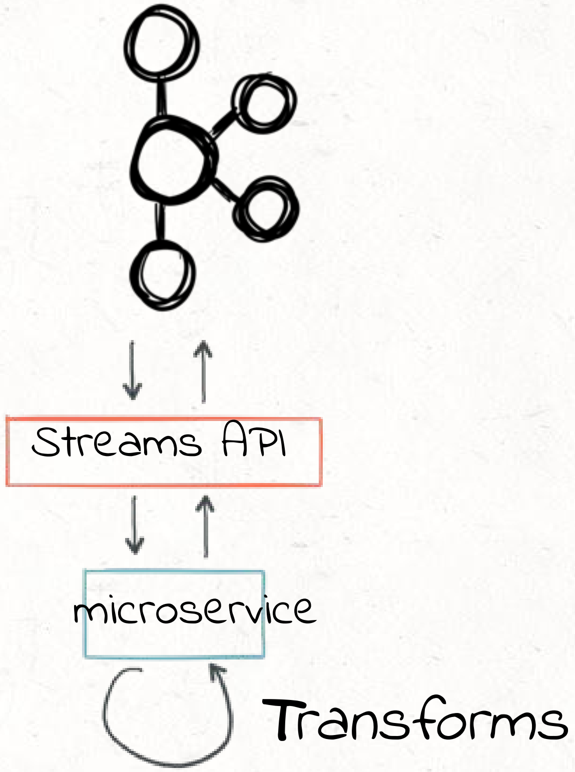
Vision 1: real-time mapreduce

STORM

Spark Streaming

Flink

Vision 2: event-driven microservices => Kafka's streams API



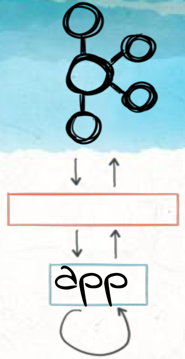
confluent

"

Kafka's Streams API = Easiest way to do stream processing using Kafka

"

#1: Powerful and lightweight Java library; need just Kafka and your app



confluent

"

#2: Convenient DSL with all sorts of operators: join(), map(), filter(), windowed aggregates etc

confluent

Word count program using Kafka's streams API

```
KStreamBuilder builder = new KStreamBuilder();
KStream<String, String> textLines = builder.stream(stringDeserializer, stringDeserializer, "TextLinesTopic");

KStream<String, Long> wordCounts = textLines
    .flatMapValues(value -> Arrays.asList(value.toLowerCase().split("\\W+")))
    .map((key, value) -> new KeyValue<>(value, value))
    .countByKey(stringSerializer, longSerializer, stringDeserializer, longDeserializer, "Counts")
    .toStream();
wordCounts.to("WordsWithCountsTopic", stringSerializer, longSerializer);

KafkaStreams streams = new KafkaStreams(builder, config);
streams.start();
```

confluent

"

#3: True event-at-a-time stream processing; no microbatching

"

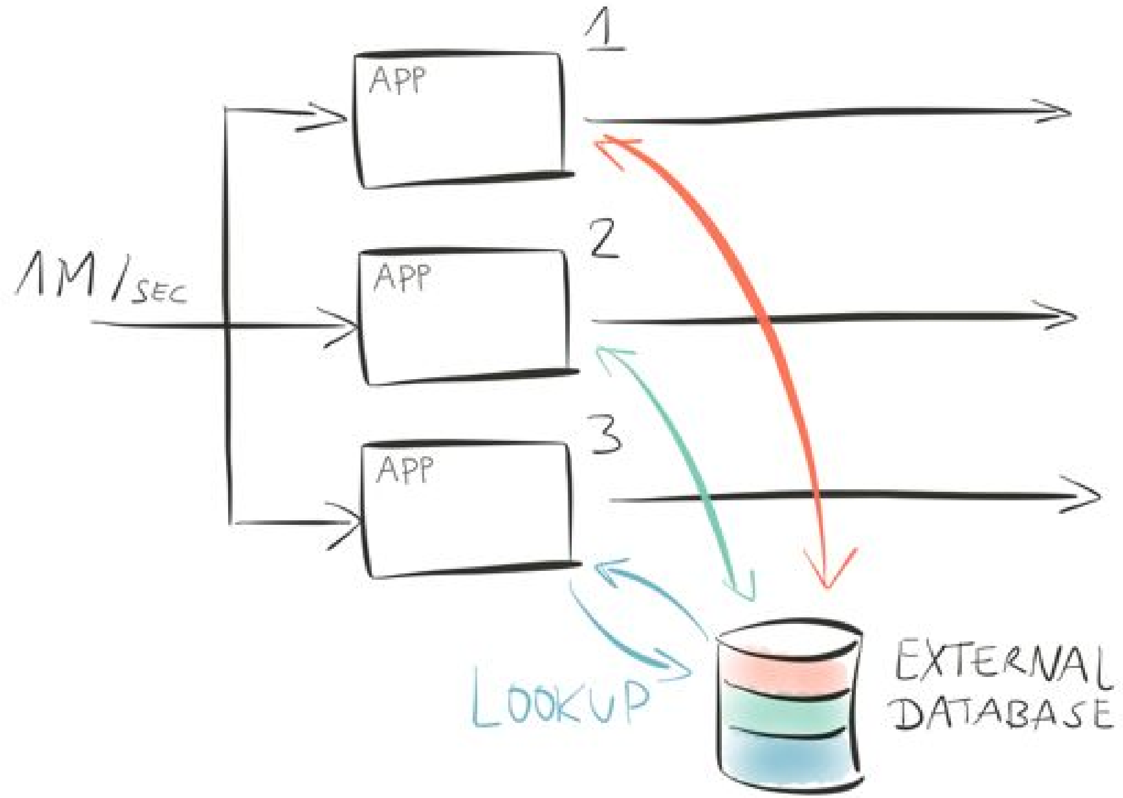
#4: Dataflow-style windowing based on event-time; handles late-arriving data

confluent

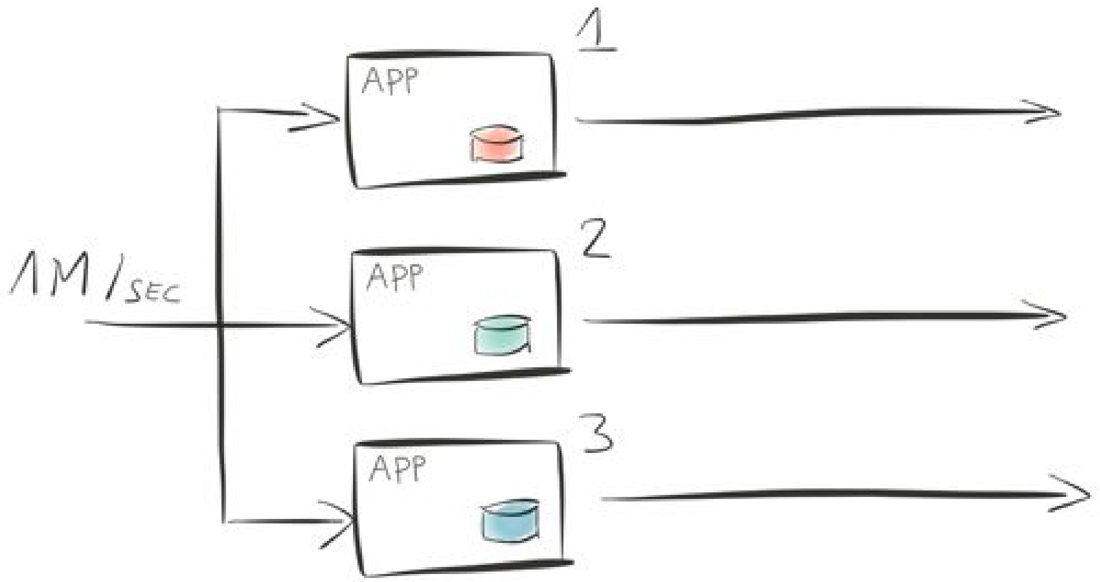
"

#5: out-of-the-box support for local state; supports fast stateful processing

External state

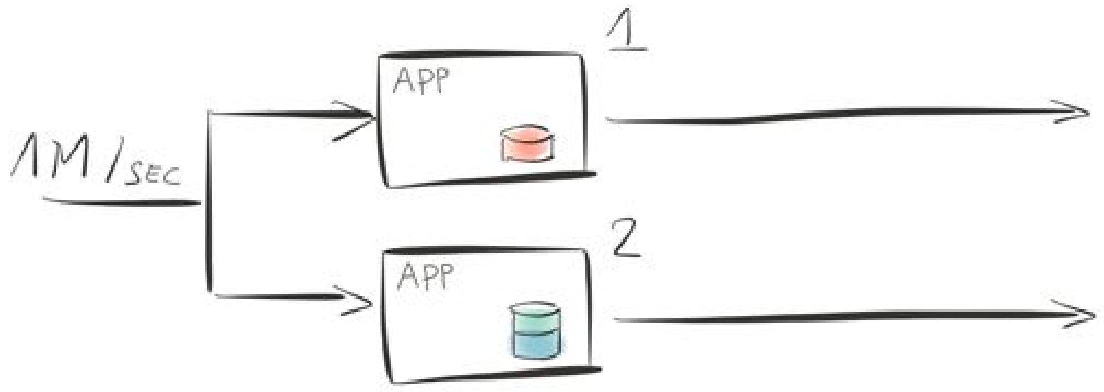


LOCAL STATE



confluent

Fault-tolerant local state

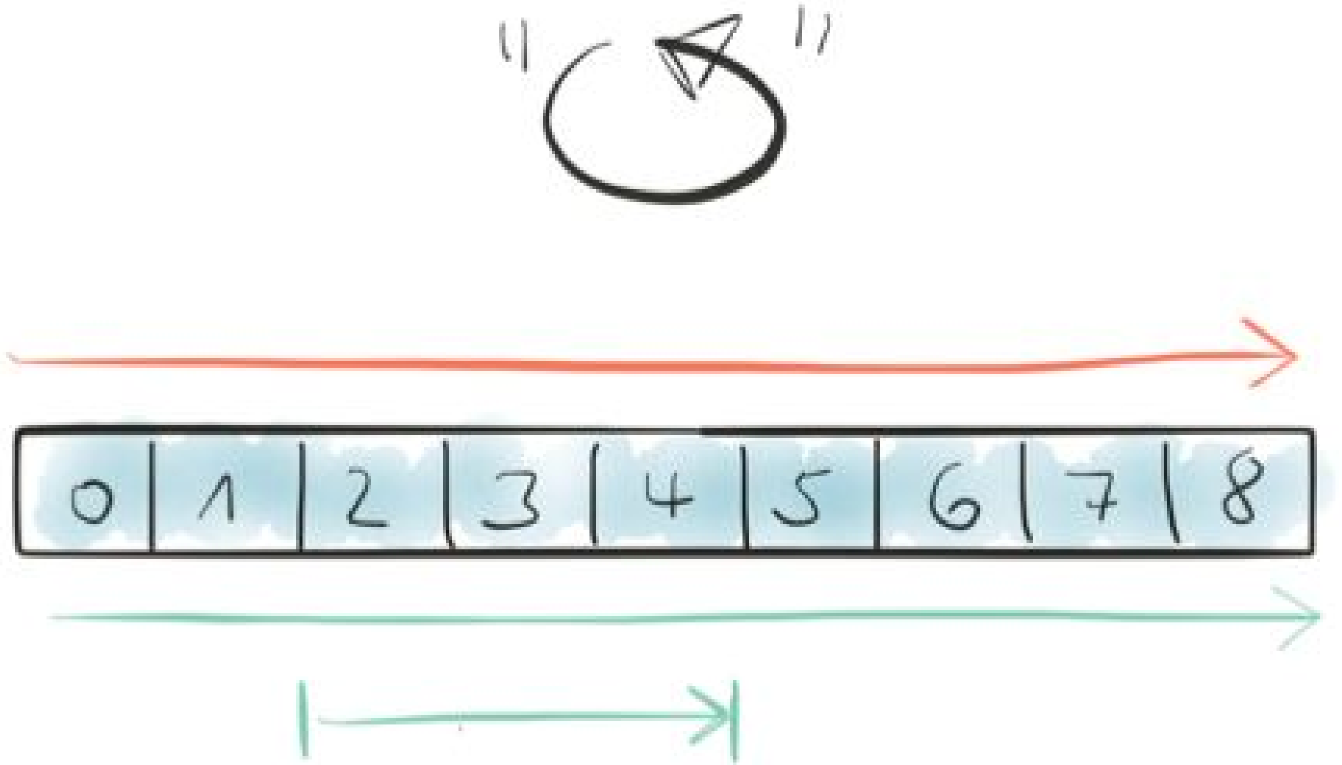


"

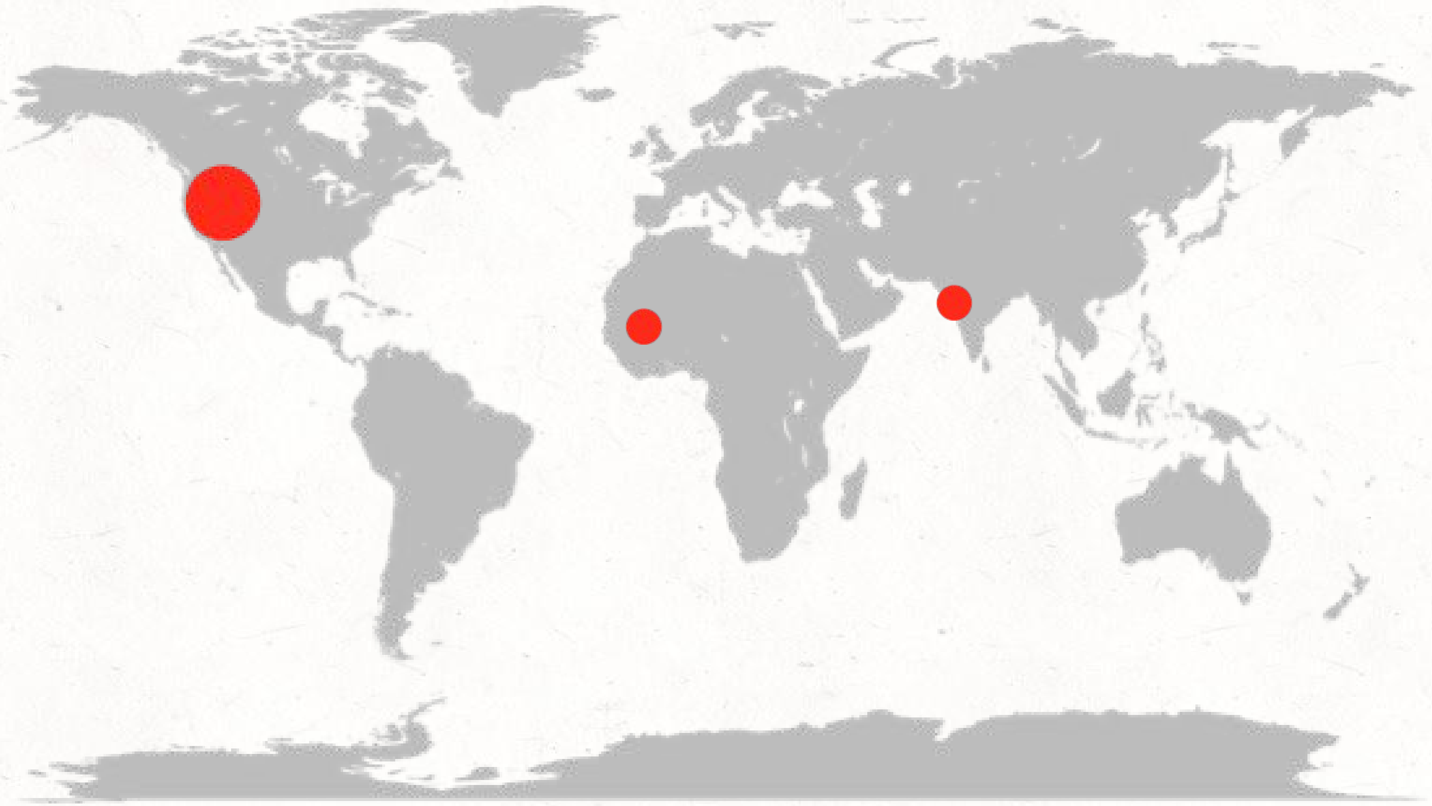
#6: Kafka's Streams API allows reprocessing; useful to upgrade apps or do A/B testing

confluent

REPROCESSING

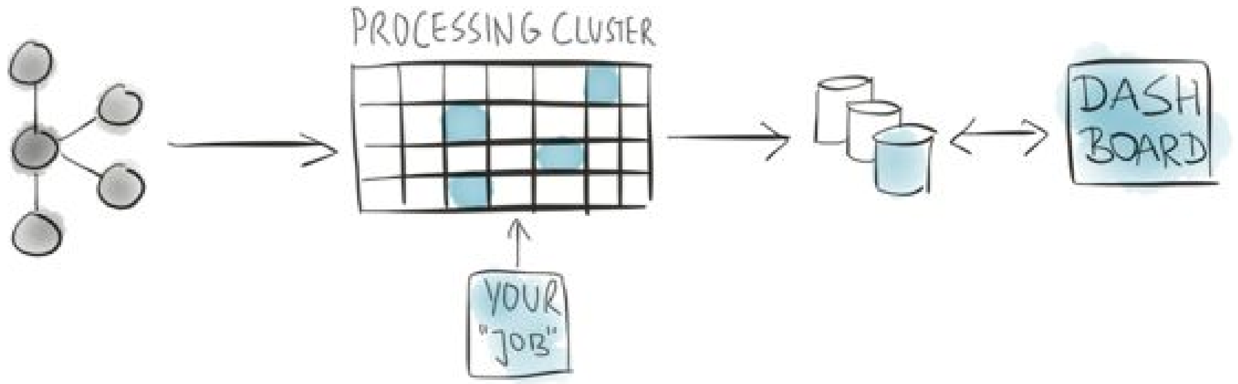


Real-time dashboard for security monitoring

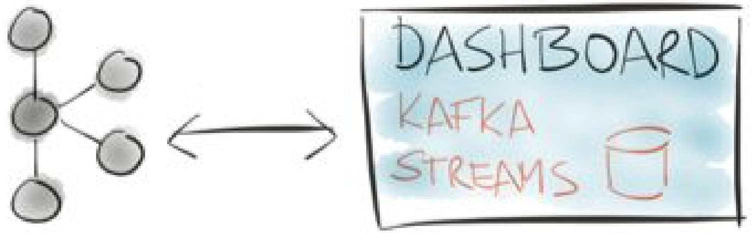


Kafka's Streams API: Simple is beautiful

Vision 1

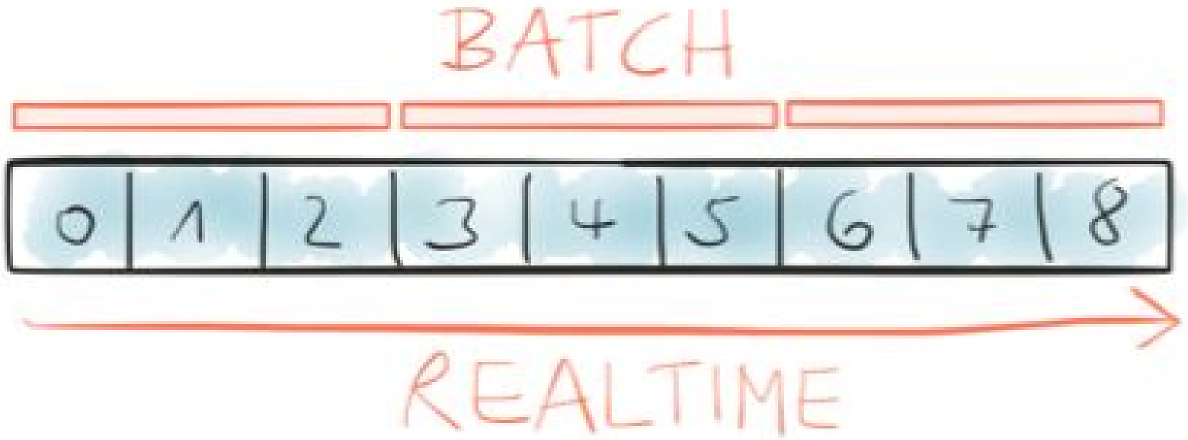


Vision 2



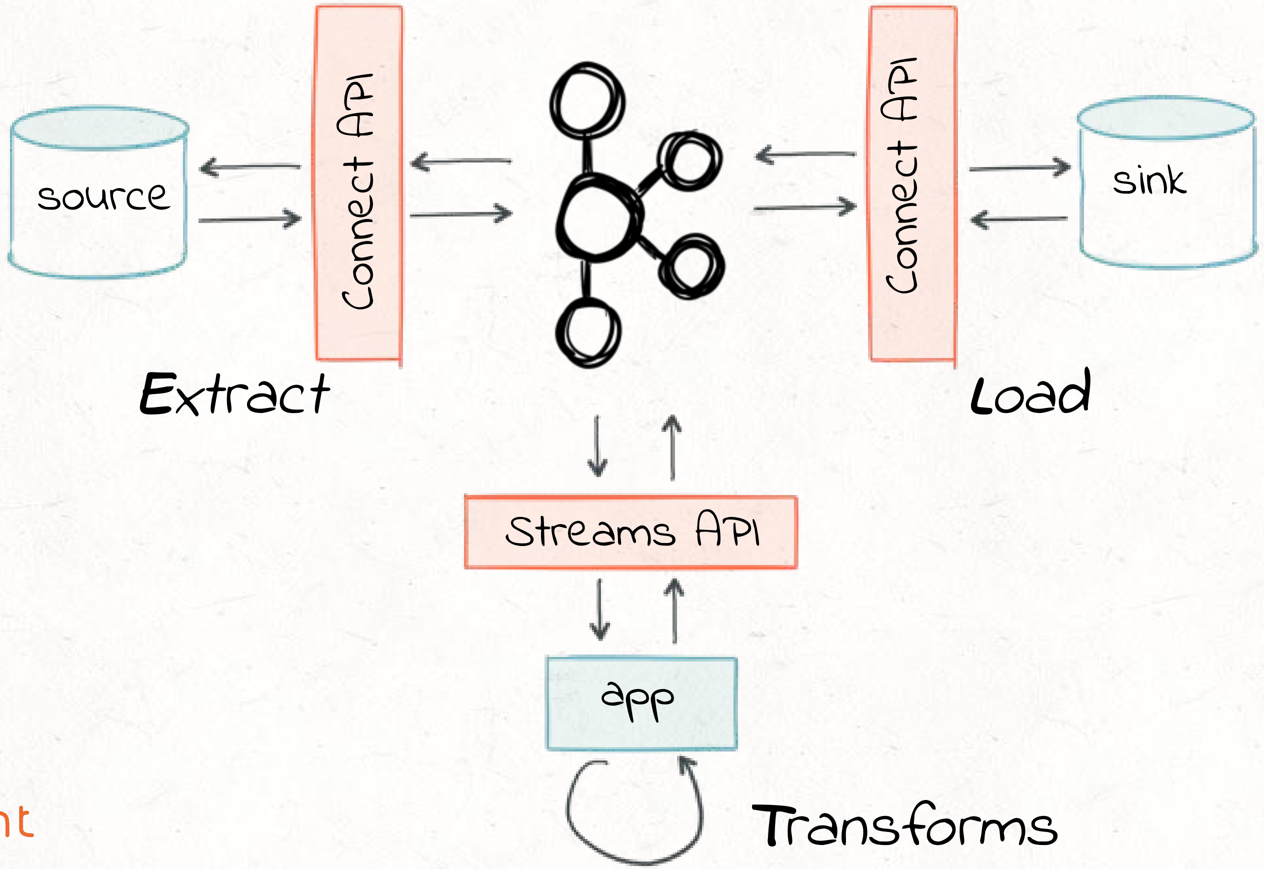
confluent

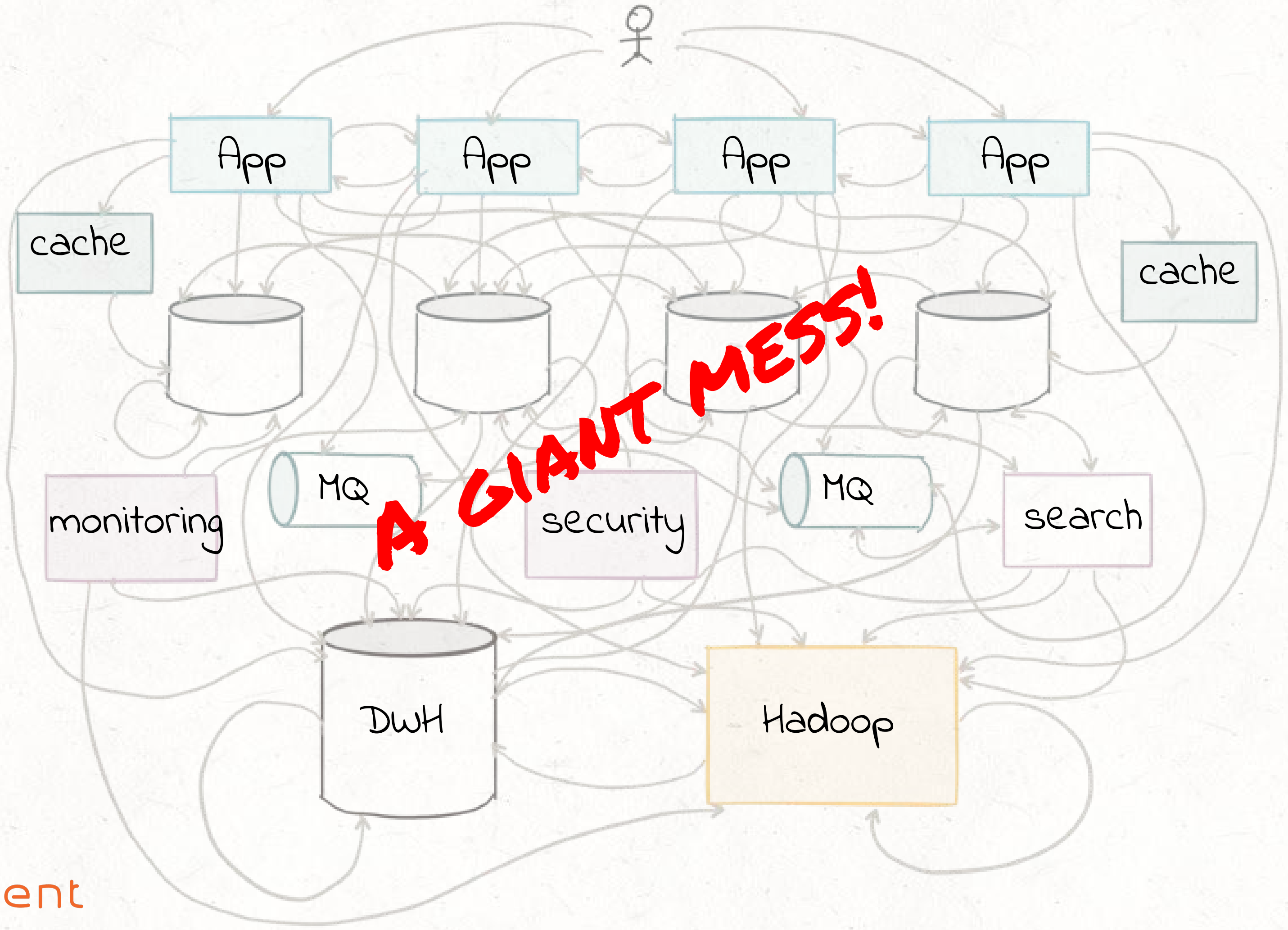
Logs unify batch and stream processing



confluent

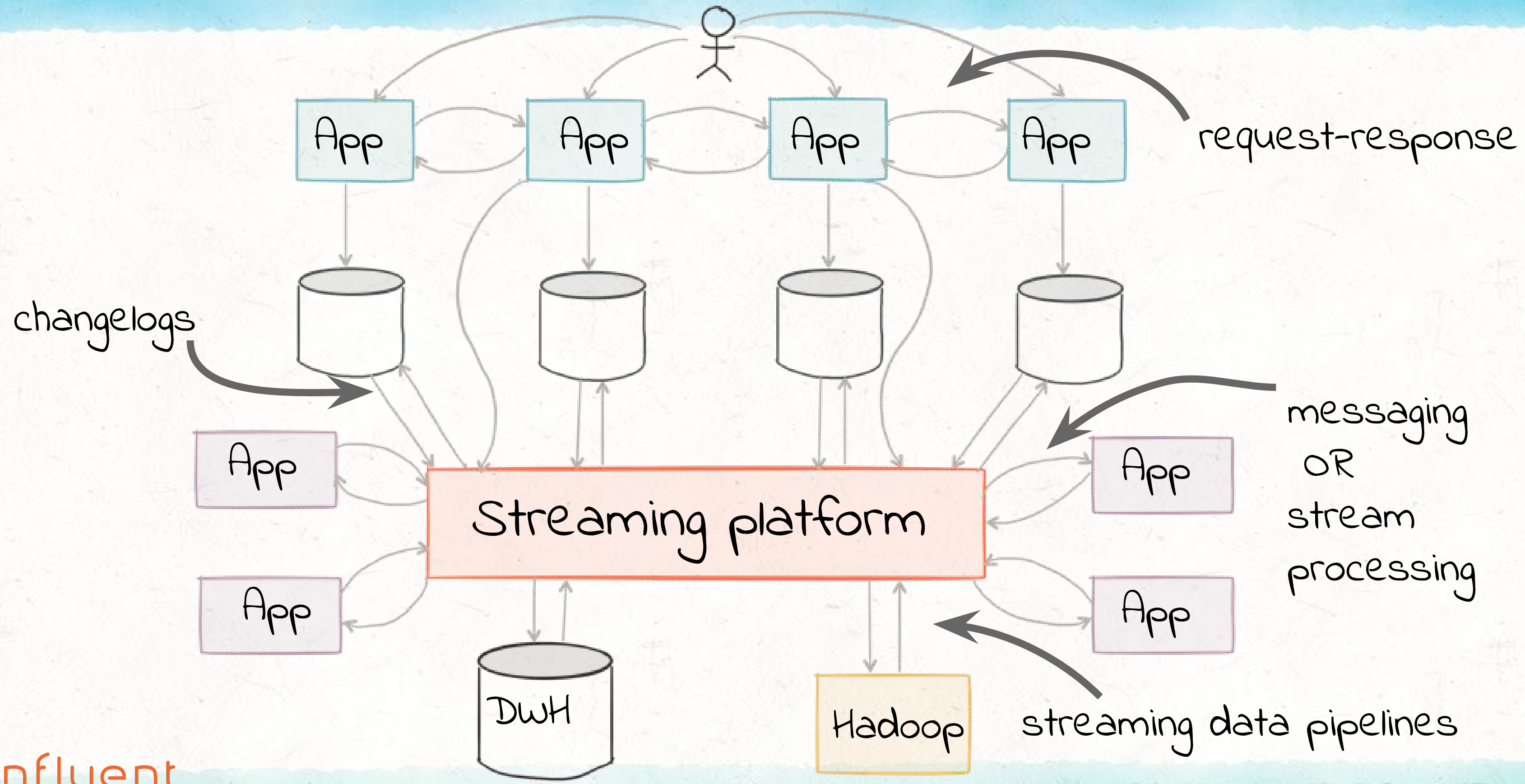
New shiny future of ETL: Kafka




App
App
App
App
cache
cache
MQ
MQ
monitoring
security
search
DWH
Hadoop
A GIANT MESS!

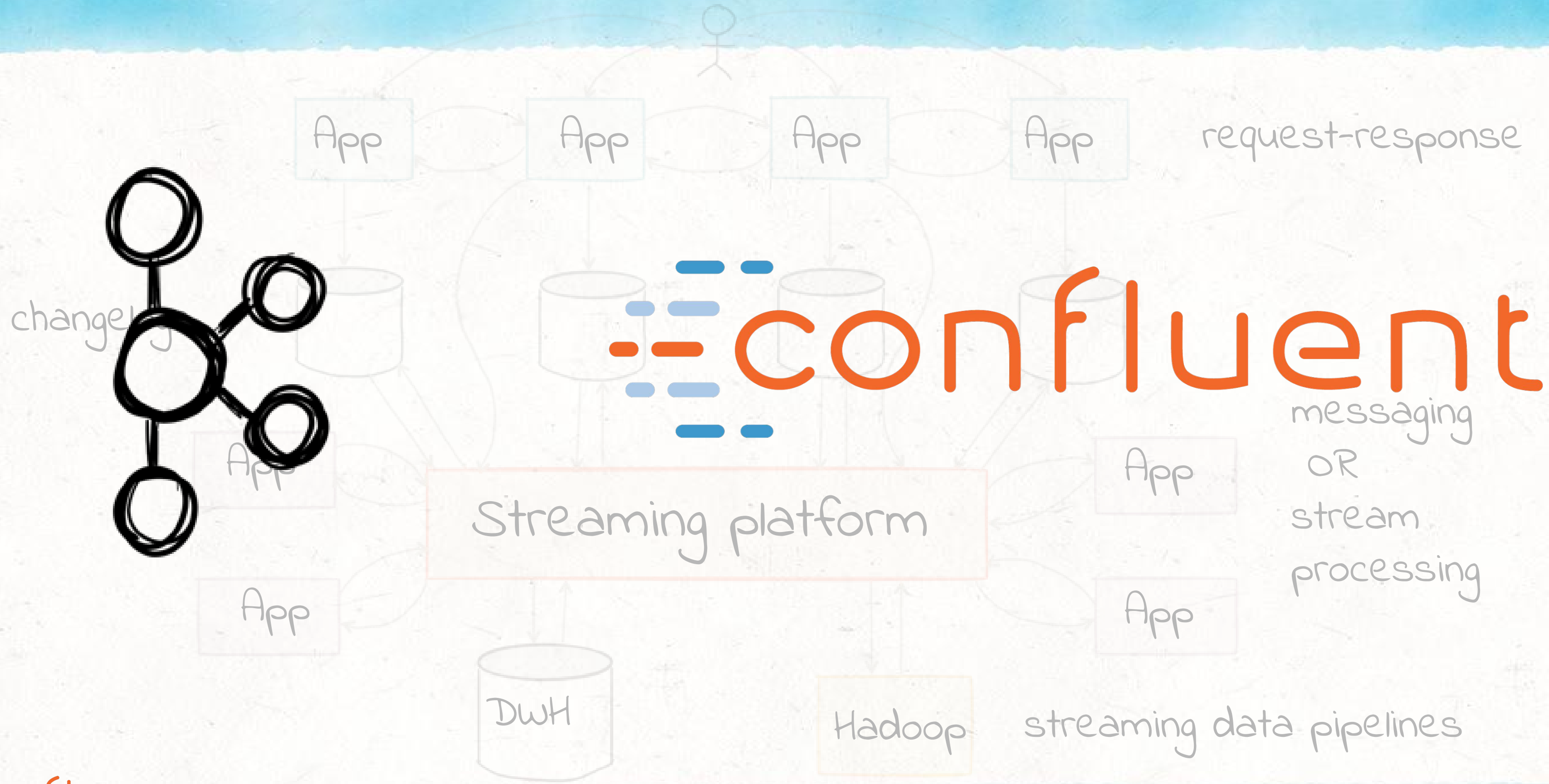
confluent

ALL YOUR DATA ... EVERYWHERE ... NOW



confluent

VISION: ALL YOUR DATA … EVERYWHERE … NOW



Thank you!

@nehanarkhede

confluent