Achieving Streaming ETL

Brian Likosar @liko9
liko@confluent.io

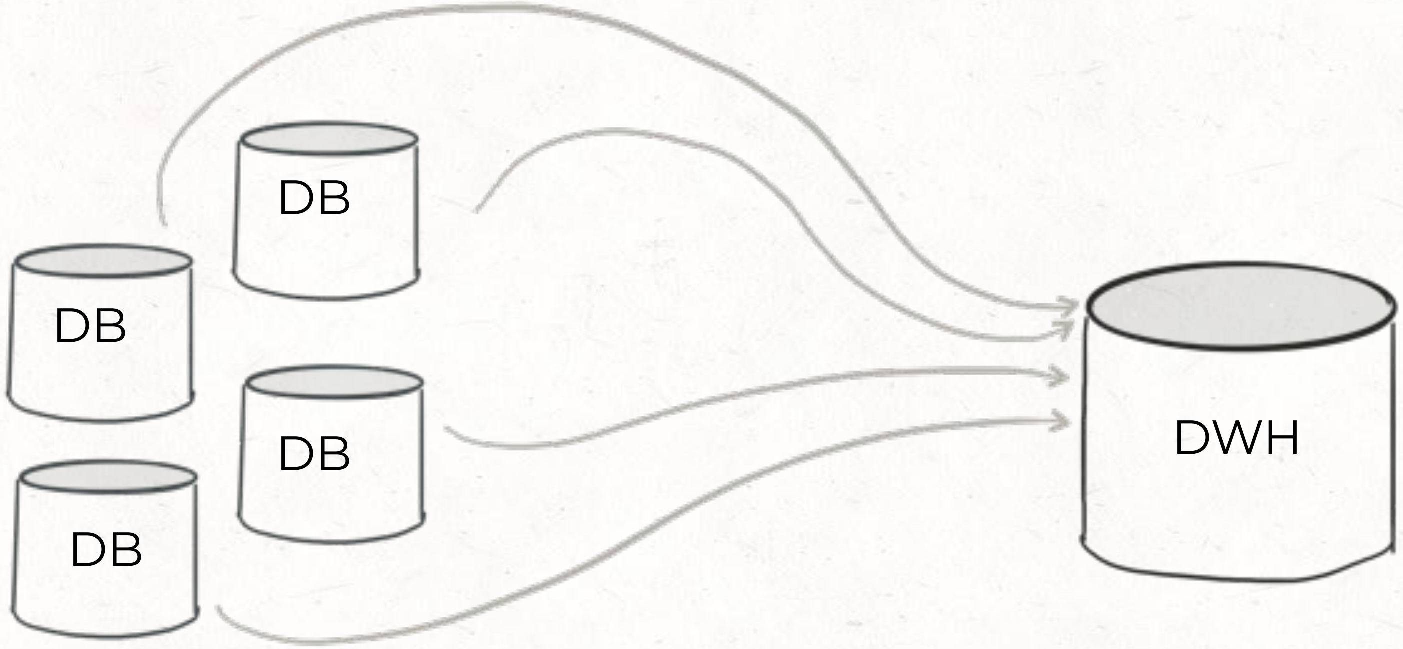
CONFLUENT

“

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**

CONFLUENT

"

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

"

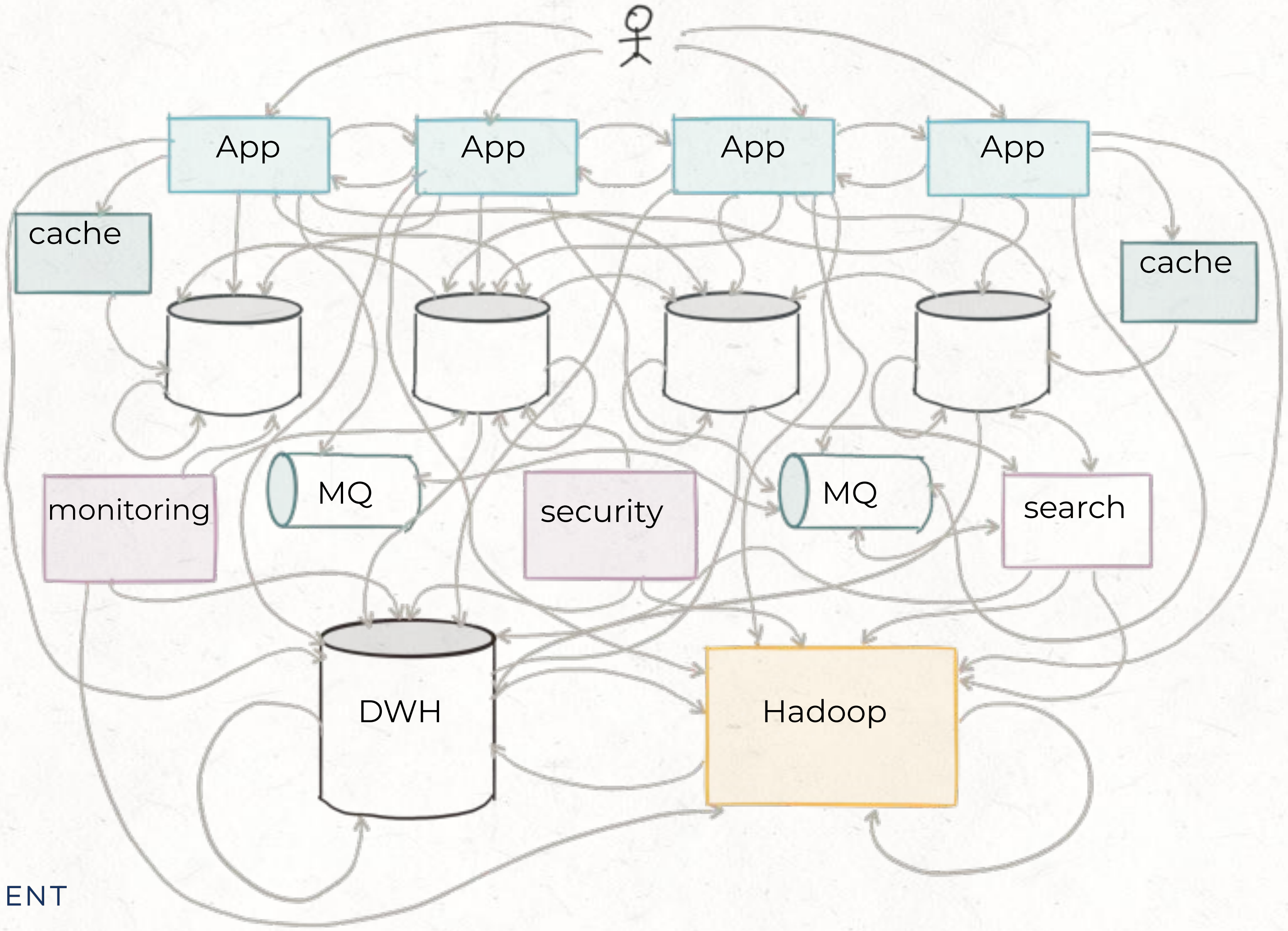
#3: Stream data is increasingly ubiquitous; need for faster processing than daily

CONFLUENT

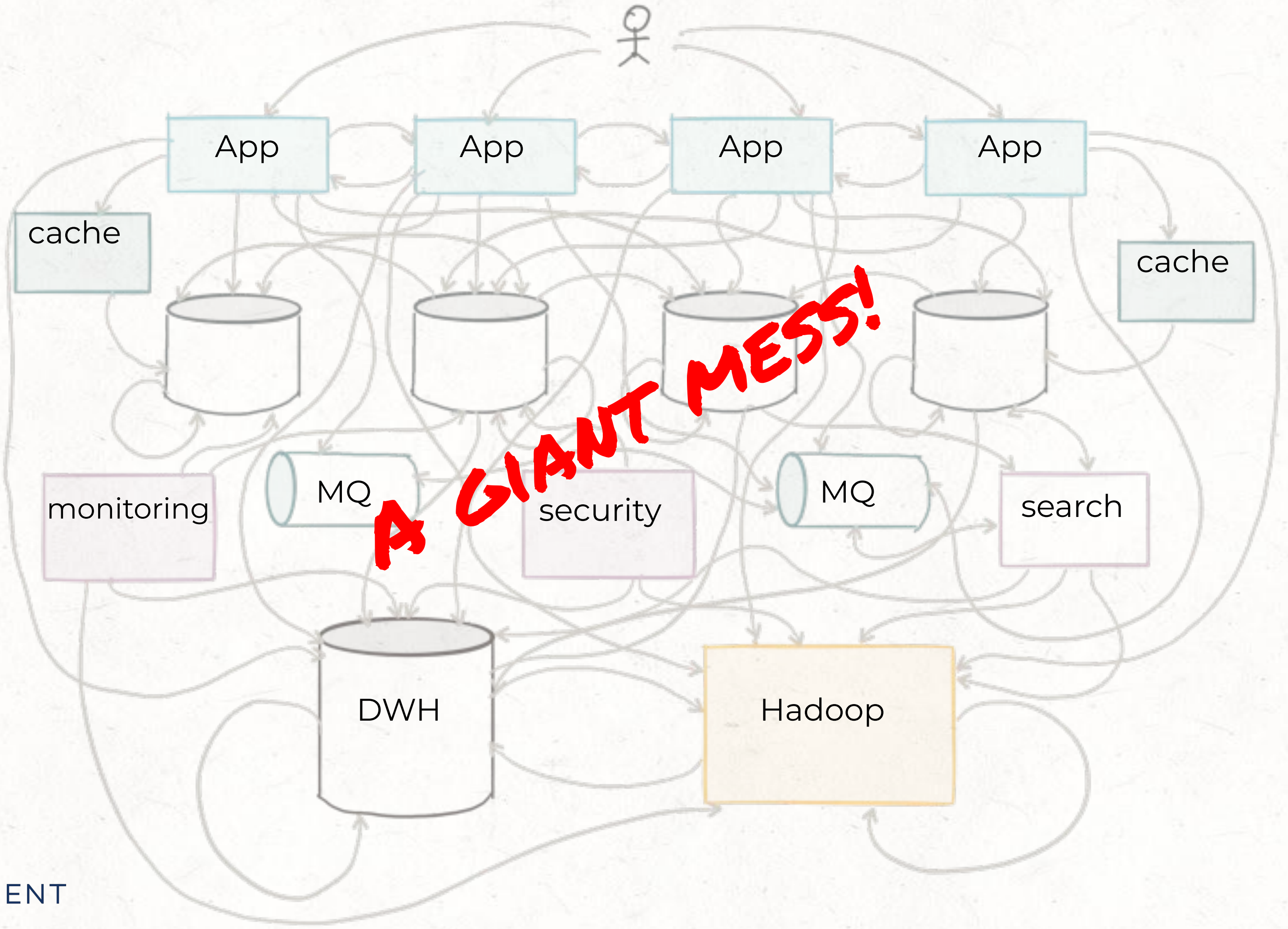
"

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

CONFLUENT


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

CONFLUENT

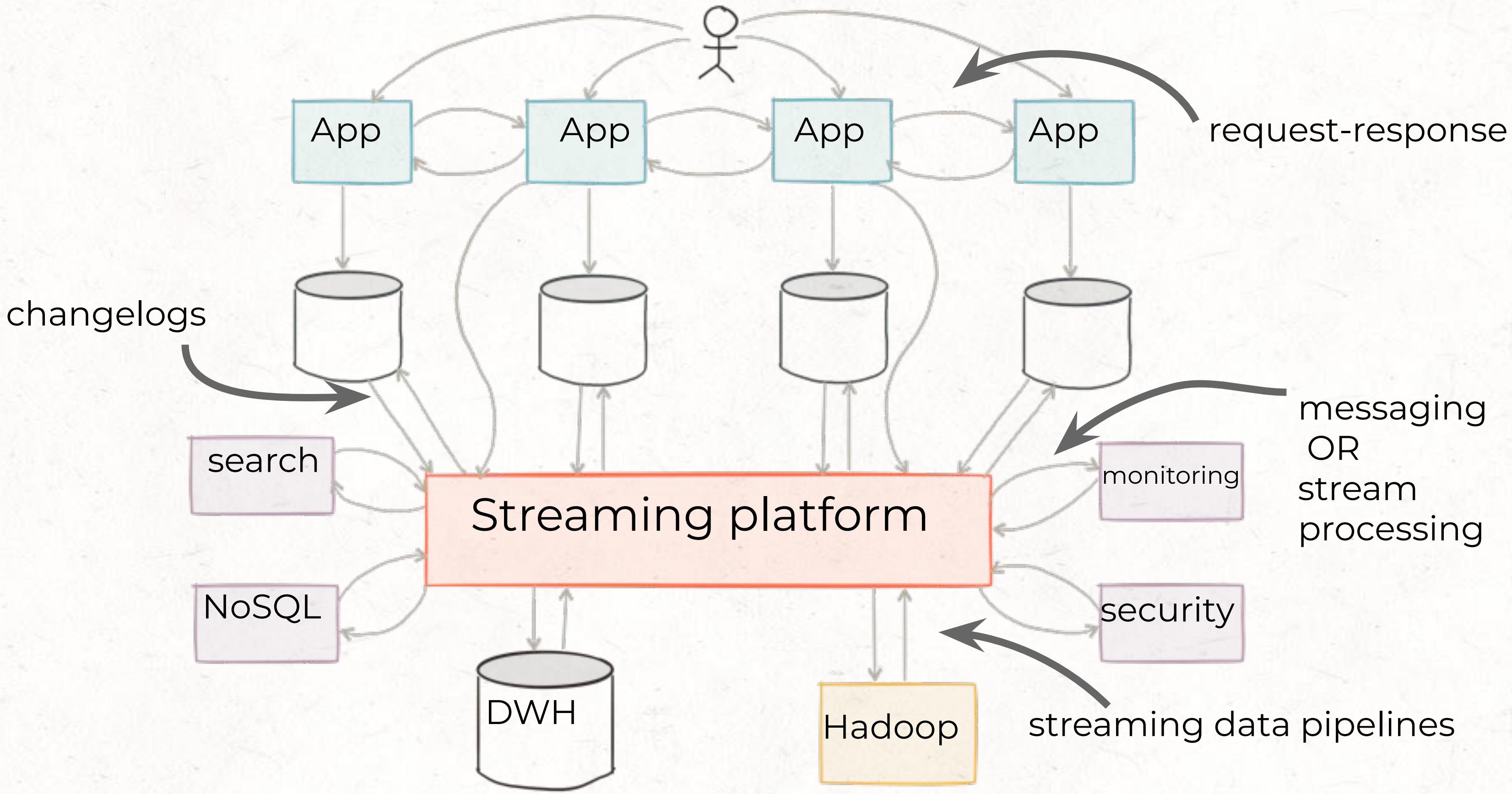

App
App
App
App
cache
cache
monitoring
MQ
security
MQ
search
DWH
Hadoop
A GIANT MESS!

CONFLUENT

"

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

CONFLUENT


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

CONFLUENT

A short history of data integration

CONFLUENT

"

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?

CONFLUENT

ETL has drawbacks

CONFLUENT

“

#1: The need for a **global schema**

CONFLUENT

"

#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***

CONFLUENT

"

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

CONFLUENT

"

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*

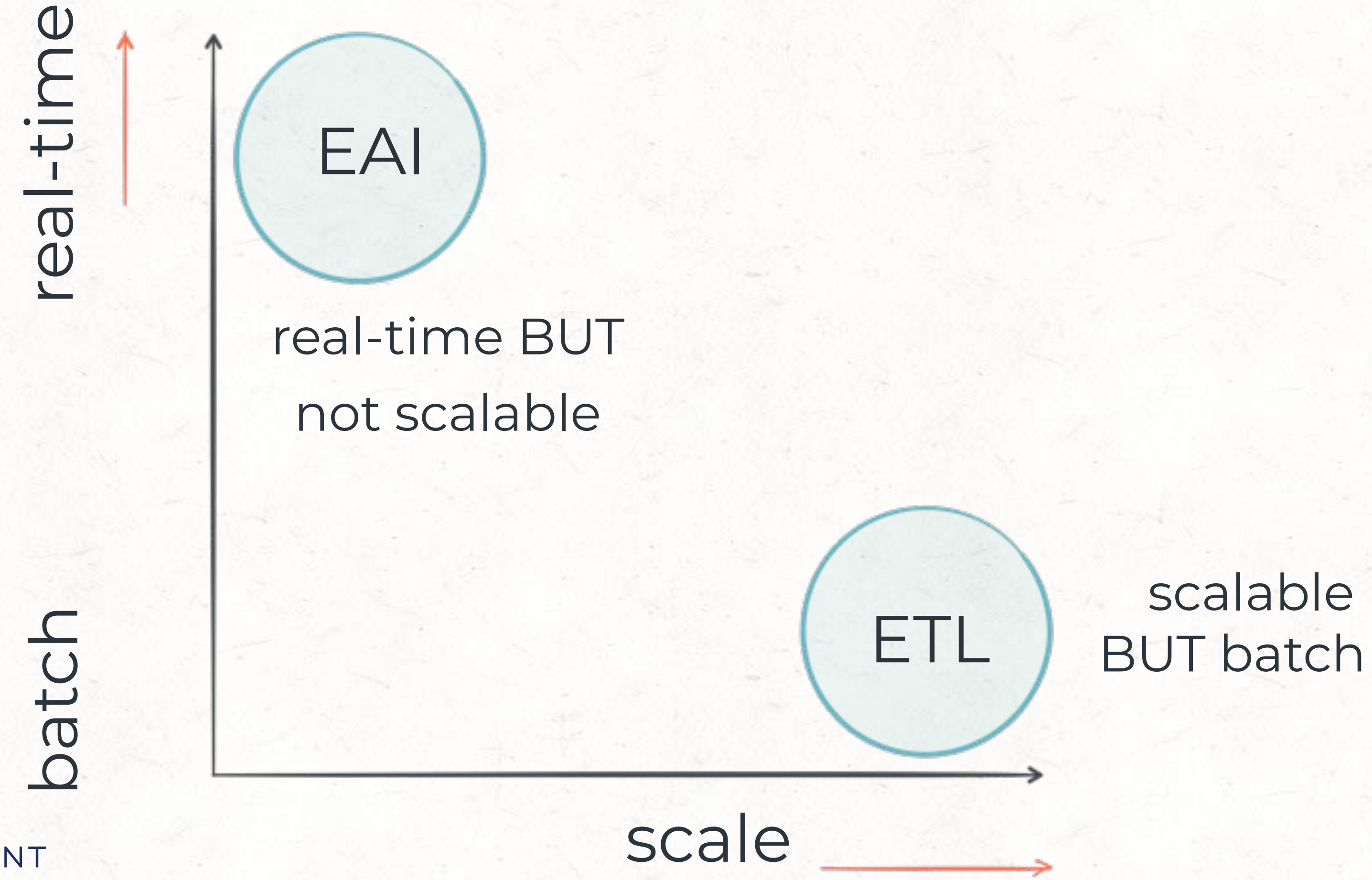
"

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

CONFLUENT

ETL and EAI are outdated!

Old world: scale or timely data, pick one



CONFLUENT

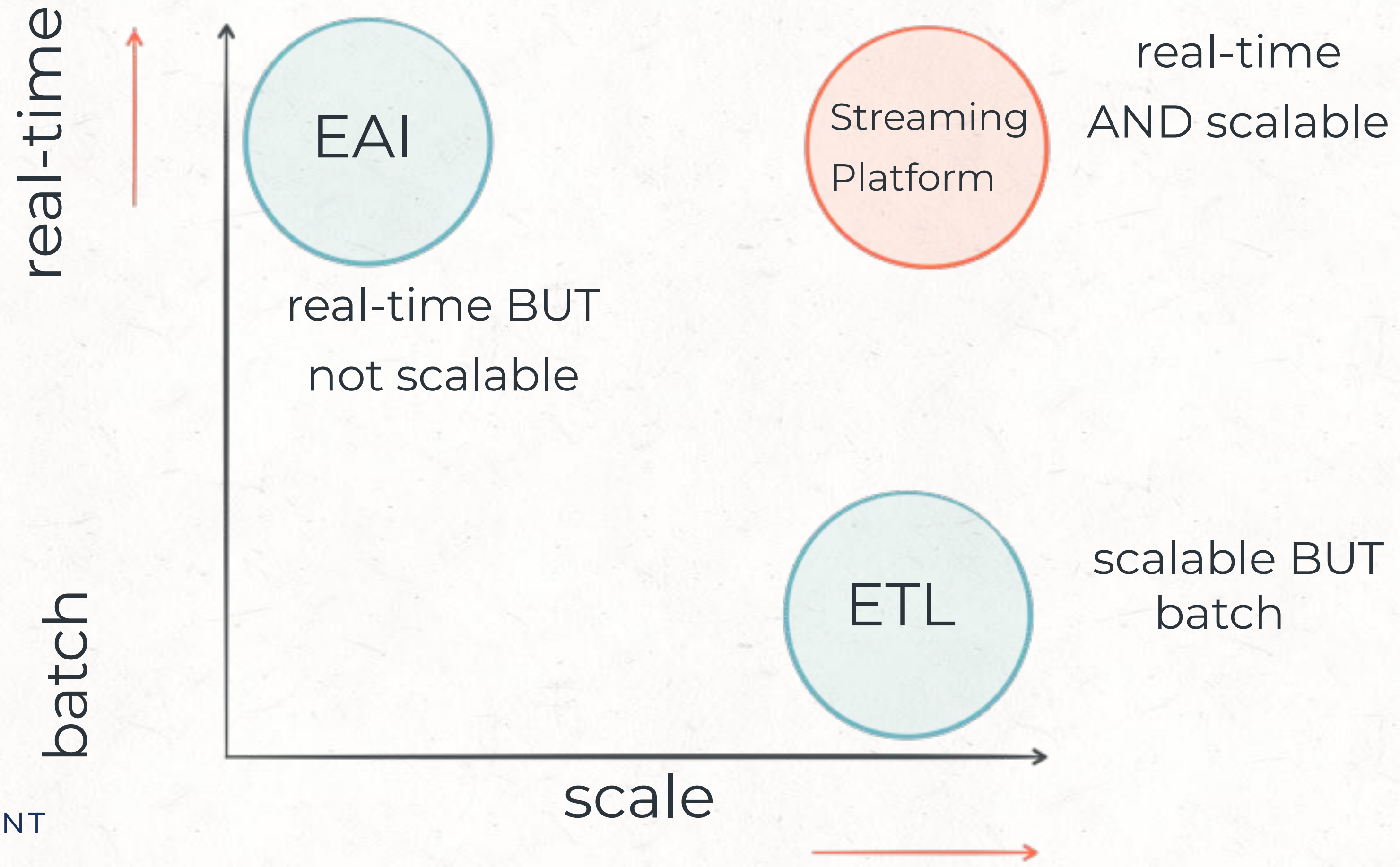
"

Data integration and ETL in the modern world need a

complete revamp

CONFLUENT

new world: streaming, real-time and scalable



CONFLUENT

"

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

CONFLUENT

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

CONFLUENT

"

#2 A streaming *platform from the grounds up; a fundamental transition to* event-centric *thinking*

CONFLUENT

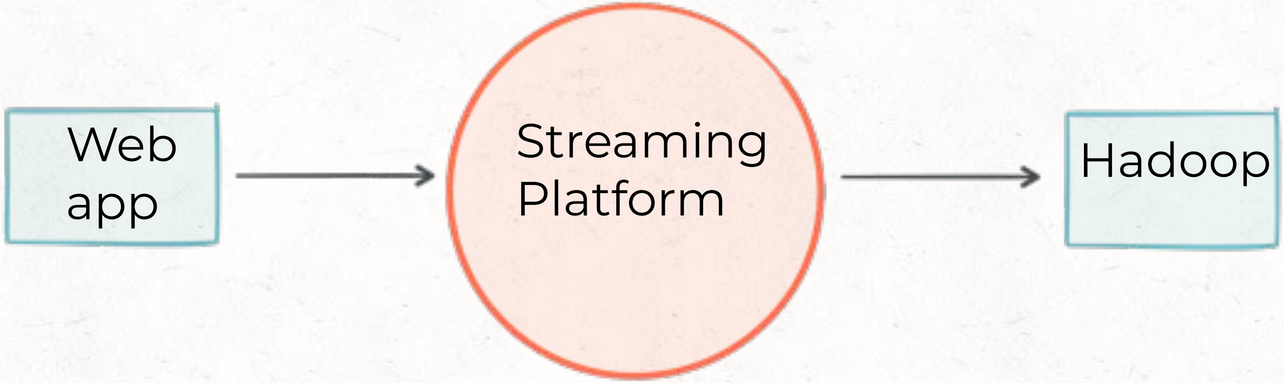
"

Streaming

~~*Batch*~~ *ETL*

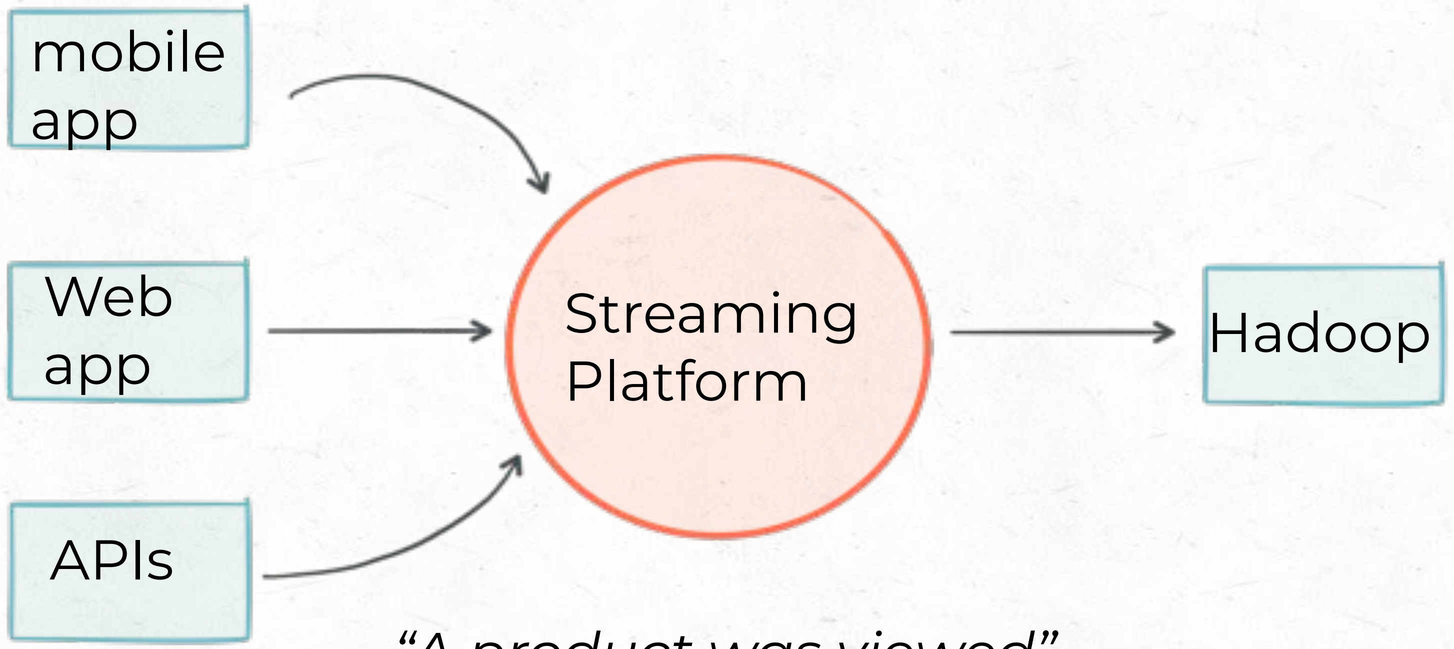
CONFLUENT

Event-Centric Thinking



"A product was viewed"

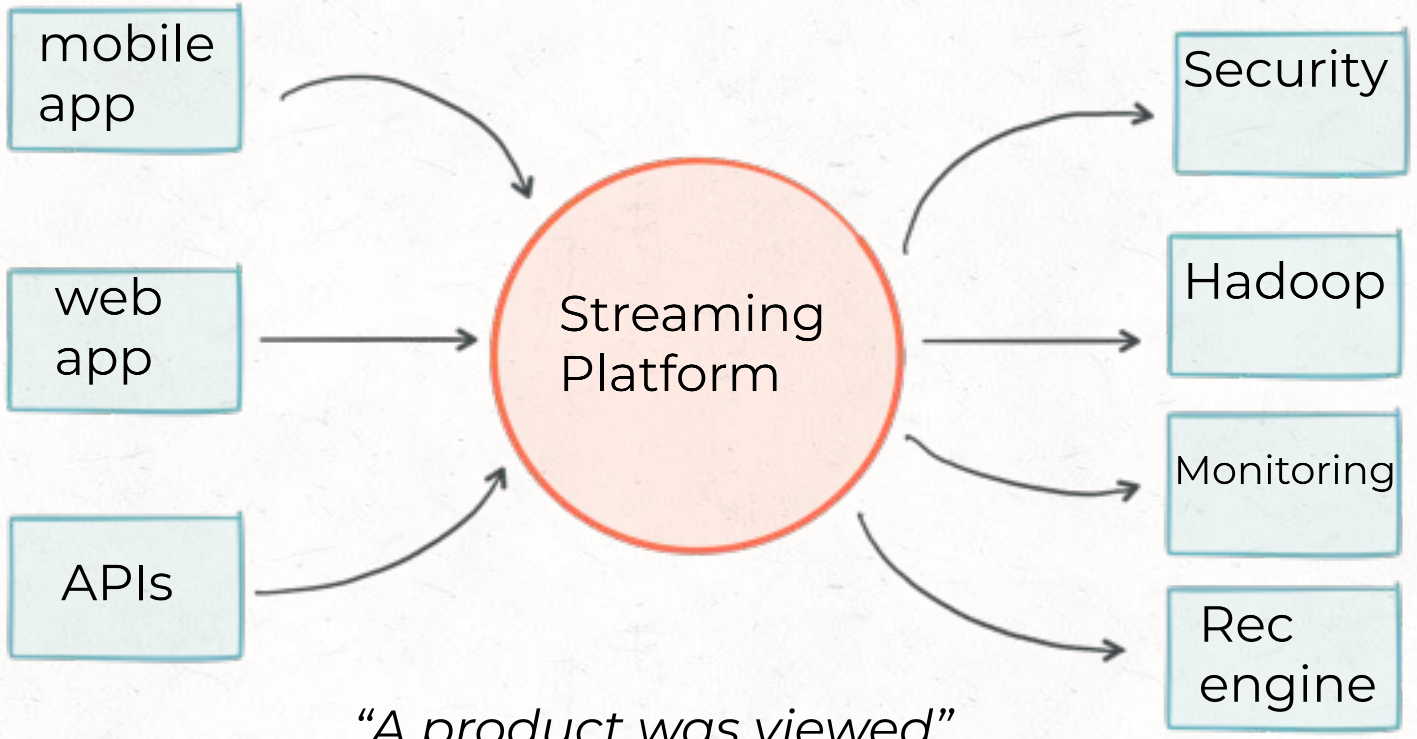
Event-Centric Thinking



"A product was viewed"

CONFLUENT

Event-Centric Thinking

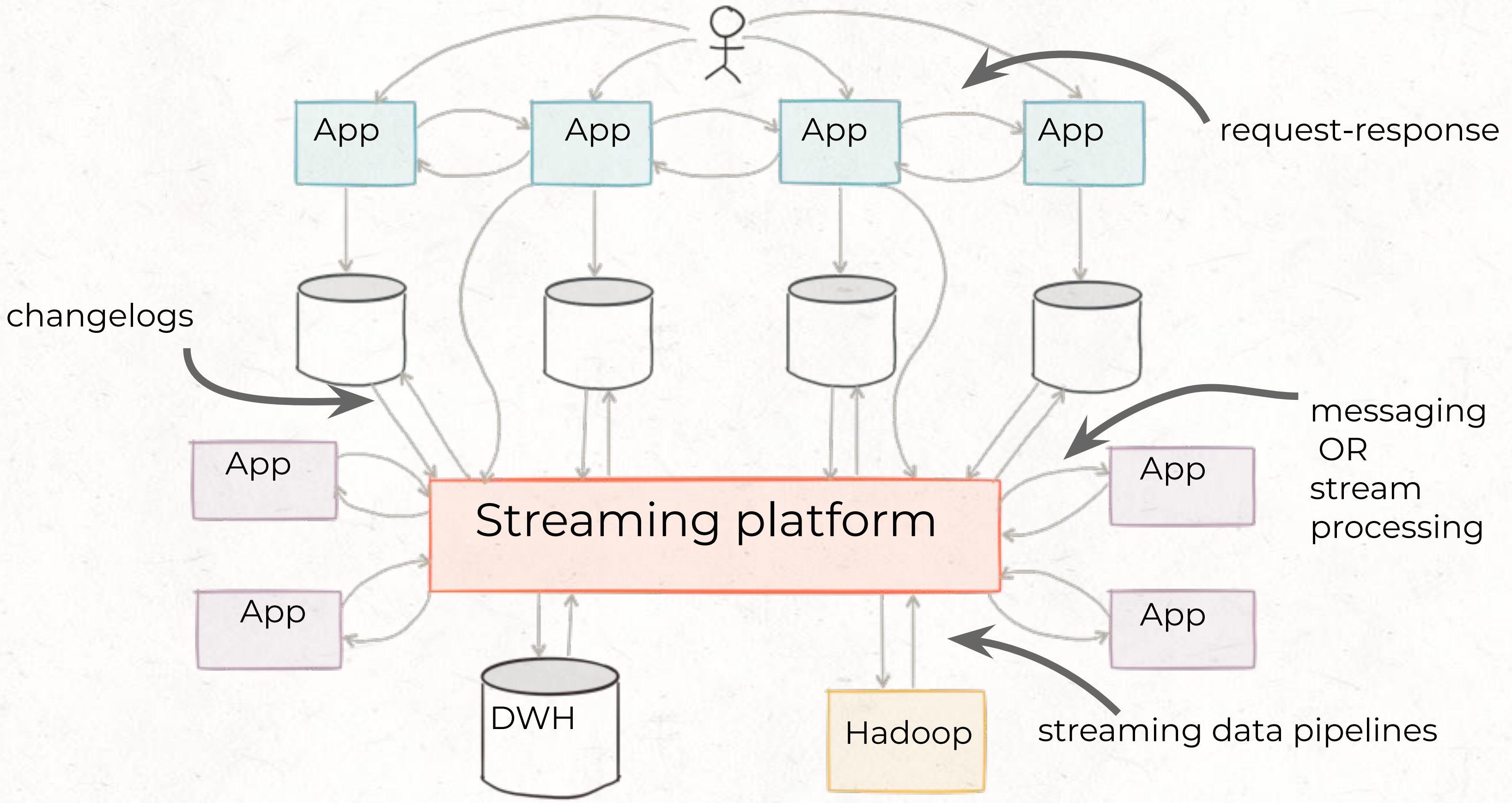


"A product was viewed"

CONFLUENT

"

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



request-response
App
App
App
App
changelogs
messaging OR stream processing
App
App
Streaming platform
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

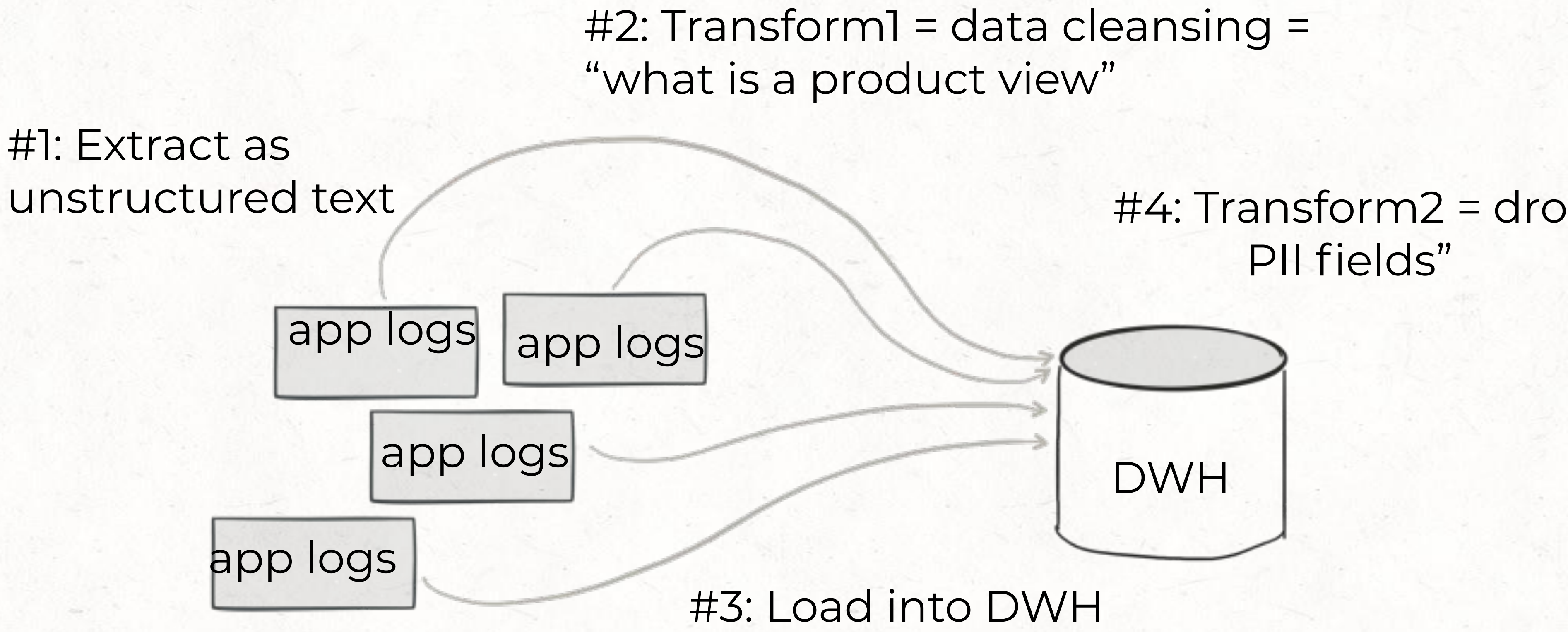
"

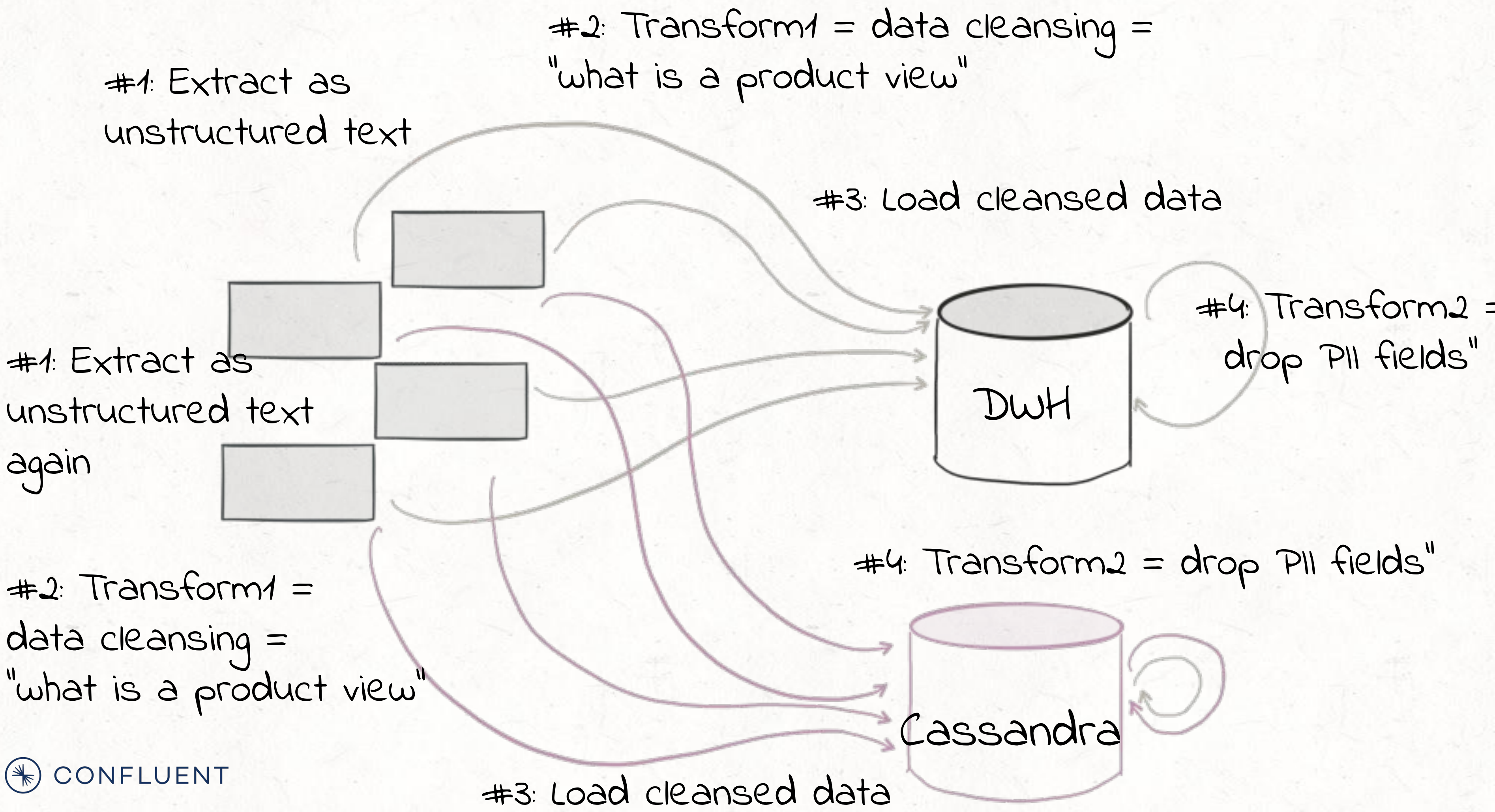
To enable forward compatibility,

redefine the T *in ETL:*

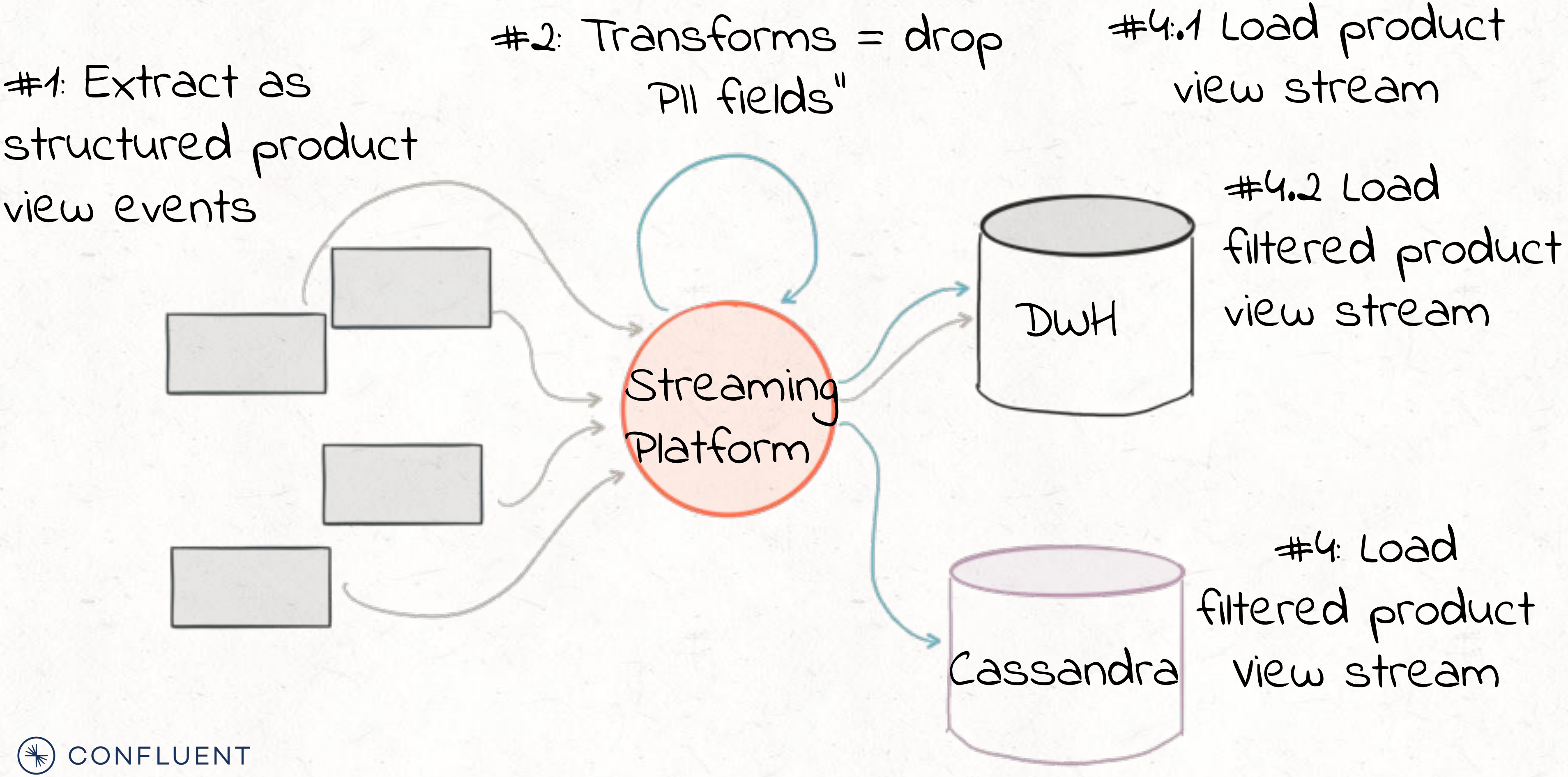
Clean data *in; Clean data out*

CONFLUENT


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


#1: Extract as unstructured text
#2: Transform1 = data cleansing = "what is a product view"
#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
#4: Load filtered product view stream
Streaming Platform
DWH
Cassandra

CONFLUENT

"

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

CONFLUENT

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

CONFLUENT

Requirements for a modern streaming data integration solution

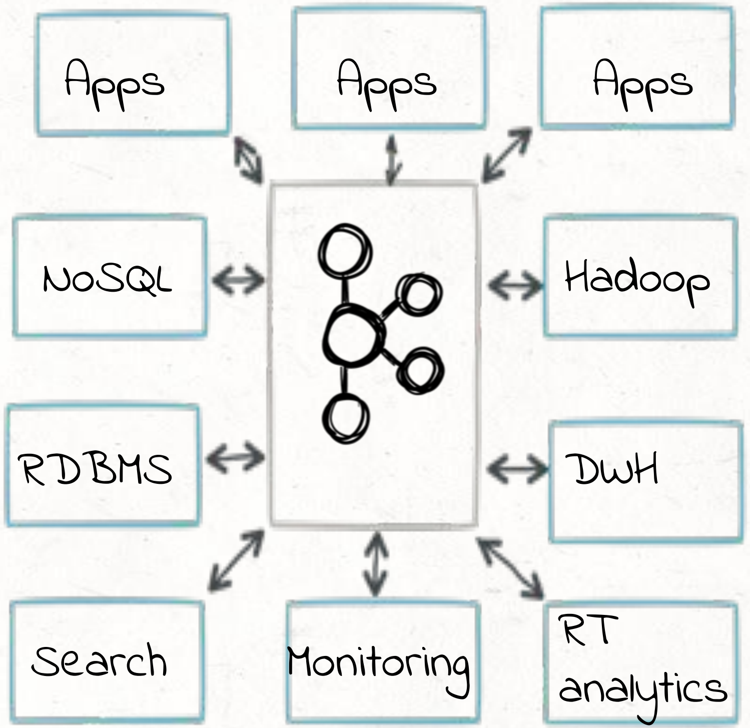
- Fault tolerance
- Parallelism
- Latency
- Delivery semantics
- Operations and monitoring
- Schema management

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 …*

CONFLUENT

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

CONFLUENT

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

CONFLUENT

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

CONFLUENT

"

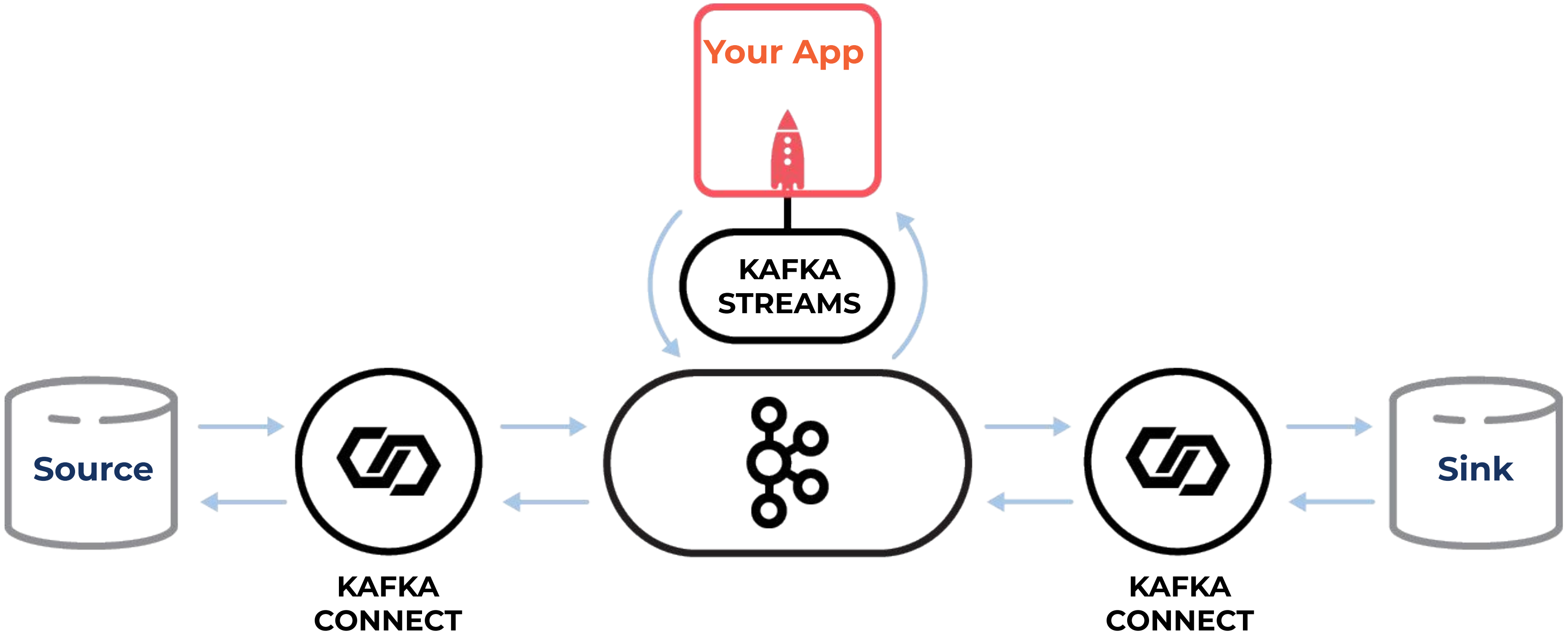
Streaming

~~Batch~~ data integration

CONFLUENT

What does a streaming platform look like and how does it enable Streaming ETL?

Kafka Connect and Kafka Streams



Instantly Connect Popular Data Sources & Sinks

Apache ACTIVEMQ
amazon KINESIS
amazon S3
Microsoft Azure Blob Storage
Azure Data Lake
cassandra
Data Diode
elastic
Google BigQuery
Google Cloud Storage
hadoop
APACHE HBASE
HTTP
IBM MQ
JDBC
JMS
MAPR
Microsoft SQL Server
MQTT .ORG
omni·sci
PostgreSQL
RabbitMQ
salesforce
splunk>
syslog

100+
pre-built connectors

ATTUNITY
camunda
Couchbase
DATASTAX
GridGain
HUMIO
HVR
IBM
imply
kinetica
XENON
mongoDB.
neo4j
ORACLE GOLDENGATE
Pivotal Greenplum
PUSH TECHNOLOGY
ROCKSET
SAP HANA
snowflake
VERTICA

80+ Confluent Supported

20+ Partner Supported, Confluent Verified

Confluent Platform

DEVELOPER

Unrestricted Developer Productivity

Multi-language Development
Non-Java Clients ● | REST Proxy ●

Rich Pre-built Ecosystem
Connectors | Hub ● | Schema Registry ●

Event Streaming Database
ksqlDB ●

OPERATOR

Efficient Operations at Scale

GUI-driven Mgmt & Monitoring
Control Center

Flexible DevOps Automation
Operator | Ansible ●

Dynamic Performance & Elasticity
Auto Data Balancer | Tiered Storage

ARCHITECT

Production-stage Prerequisites

Enterprise-grade Security
RBAC | Secrets | Audit Logs

Data Compatibility
Schema Registry ● | Schema Validation

Global Resilience
Multi-region Clusters | Replicator

Apache Kafka

● Open Source | Community licensed

Self-managed Software — **Freedom of Choice** — **Fully Managed Cloud Service**

Enterprise Support — **Professional Services** — **Committer-driven Expertise** — **Training** — **Partners**

Two Ways to Deploy Confluent

SELF-MANAGED SOFTWARE

Confluent Platform

The Enterprise Distribution of Apache Kafka

Deploy on any platform, on-prem or cloud

FULLY-MANAGED SOFTWARE

Confluent Cloud

Apache Kafka Re-Engineered for the Cloud

Available on the leading public clouds

aws Google Cloud Platform Microsoft Azure

CONFLUENT