LONDON

Debezium vs. the world

An overview of the CDC ecosystem

Marta Paes

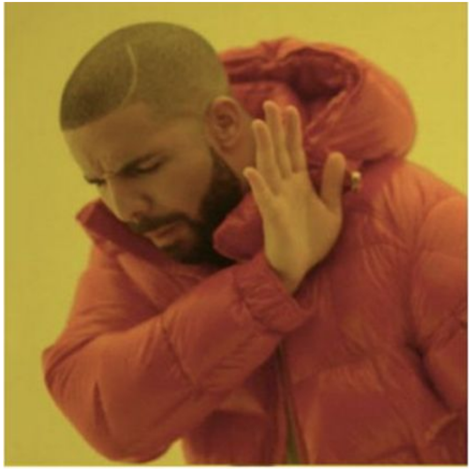
Sr. Product Manager @Materialize

This is not a 🌶️ talk.

Things move **fast**. If you notice inaccuracies, or are building a tool that could be featured in a future version of this talk, come around after the talk!

What we talk about when we talk about CDC

Query-based CDC



- ❌ Some data changes might get lost
- ❌ DELETE operations are not captured
- ❌ Trade-off: frequency vs. load on source DBs
- ❌ Can’t propagate schema changes

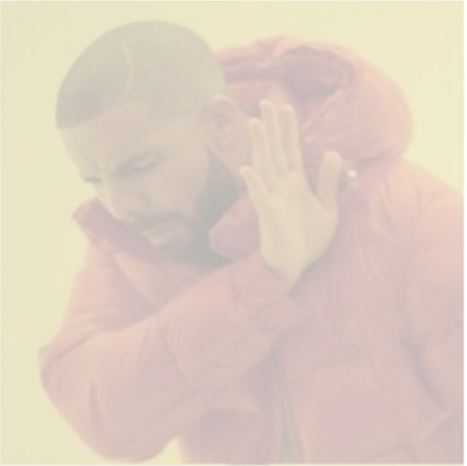
What we talk about when we talk about CDC

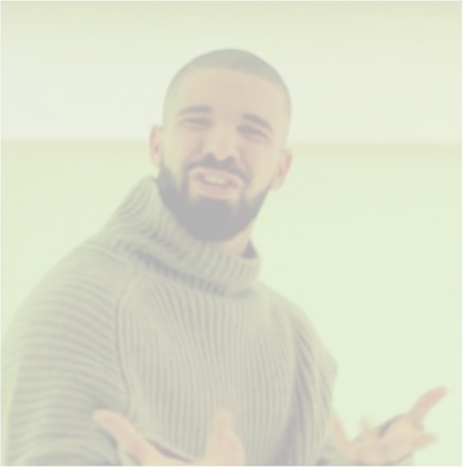
Query-based CDC

What if we just tapped into the transaction log?

What we talk about when we talk about CDC

Query-based CDC





Log-based CDC

- ✅ All data changes are captured
- ✅ More context on the actual changes
- ✅ Low propagation delay (*i.e.* near real time)
- ✅ Less taxing on the source database

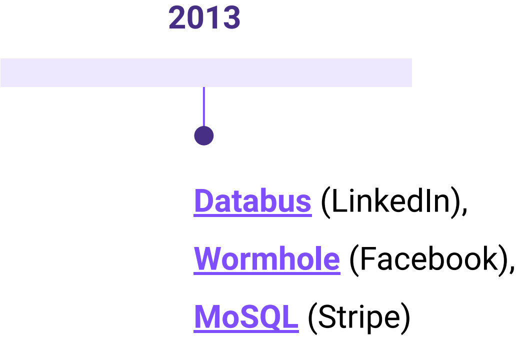
Tale of the tape

Or, how it all started.

Materialize

How it all started

Like most tools that are a commodity in streaming today, the first CDC systems were developed at internet-scale companies.



2013

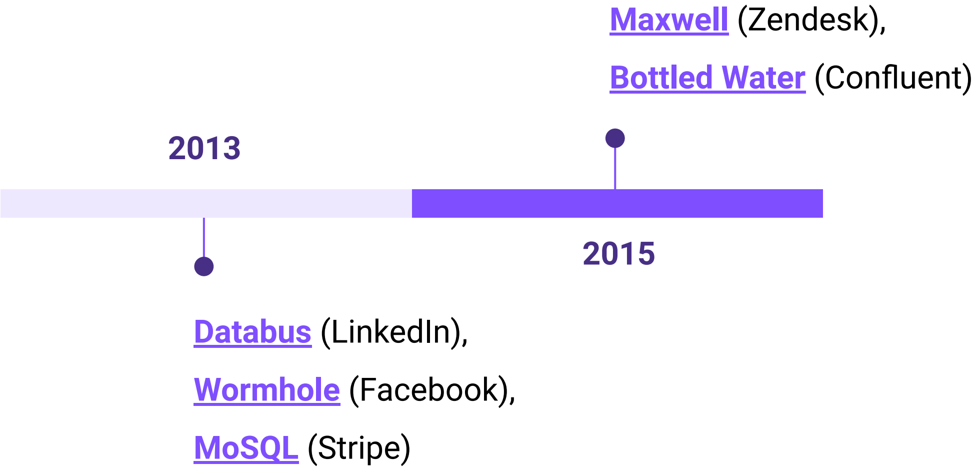
Databus (LinkedIn),

Wormhole (Facebook),

MoSQL (Stripe)

How it all started

Like most tools that are a commodity in streaming today, the first CDC systems were developed at internet-scale companies.



Maxwell (Zendesk),

Bottled Water (Confluent)

2013

2015

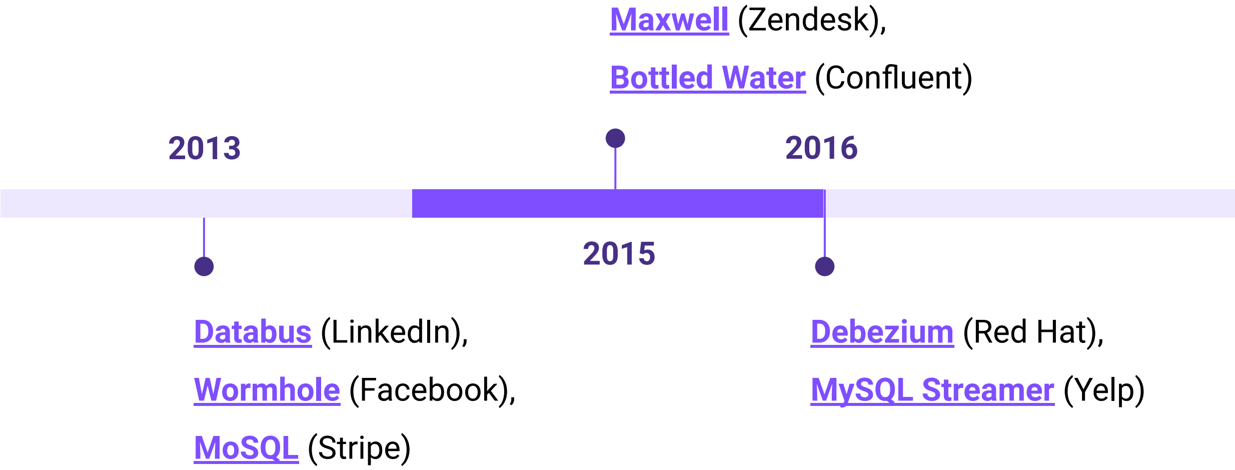
Databus (LinkedIn),

Wormhole (Facebook),

MoSQL (Stripe)

How it all started

Like most tools that are a commodity in streaming today, the first CDC systems were developed at internet-scale companies.



2013

Databus (LinkedIn),
Wormhole (Facebook),
MoSQL (Stripe)

2015

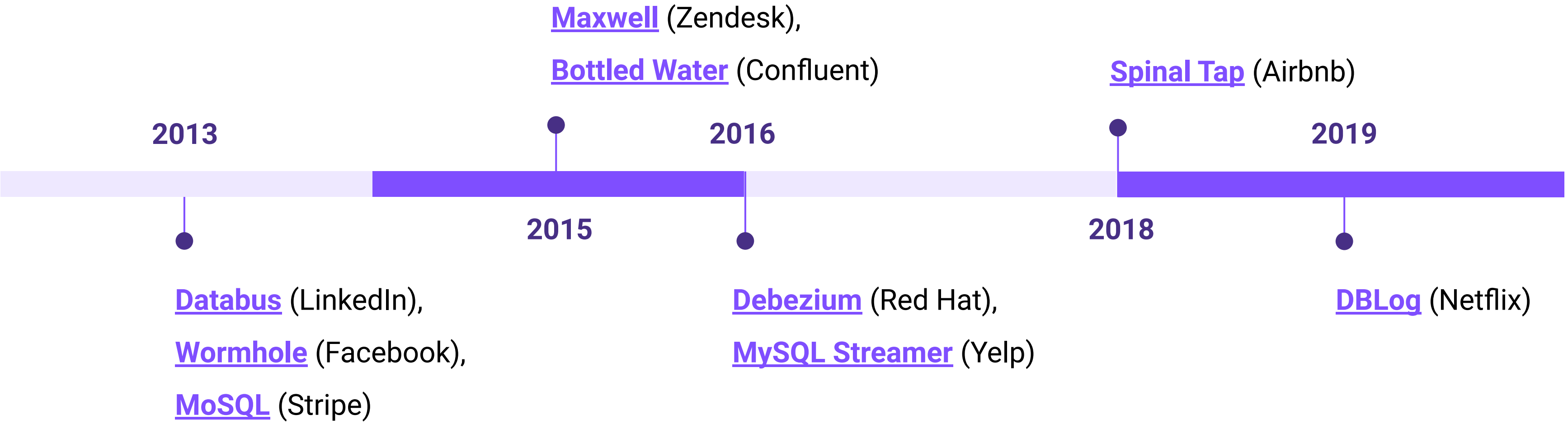
Maxwell (Zendesk),
Bottled Water (Confluent)

2016

Debezium (Red Hat),
MySQL Streamer (Yelp)

How it all started

Like most tools that are a commodity in streaming today, the first CDC systems were developed at internet-scale companies.



2013

Databus (LinkedIn),
Wormhole (Facebook),
MoSQL (Stripe)

2015

Maxwell (Zendesk),
Bottled Water (Confluent)

2016

Debezium (Red Hat),
MySQL Streamer (Yelp)

2018

Spinal Tap (Airbnb)

2019

DBLog (Netflix)

Where it landed

Debezium has become the standard CDC tool over time, with a **strong community** behind it. Like **any tool**, it has some good and some less good.

The good 😚

- Deployment via well-understood tools (Kafka + Kafka Connect).
- Standard schema for change events.
- Support for a large number of CDC connectors.

The less good 😕

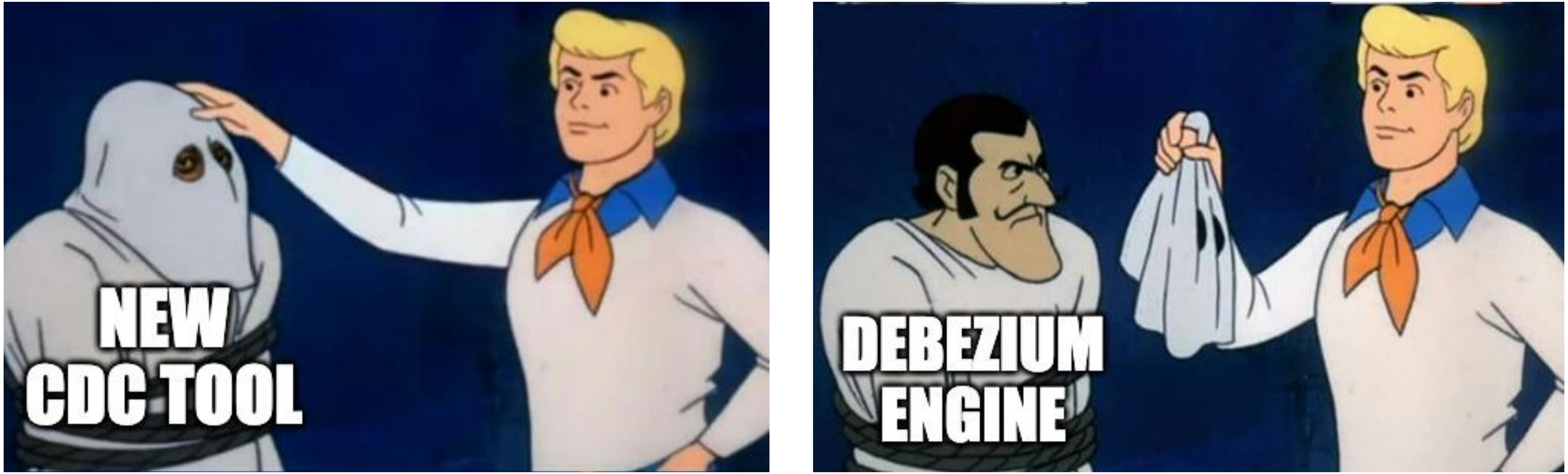
- At-least-once delivery guarantees*, no transactional consistency OOTB.
- No graceful schema evolution OOTB.

* Exactly-once support (KIP-618) will gradually roll out, starting with the PostgreSQL connector in 2.3.

Round 1 🔔

Same same, but different.

“Have you heard about this new CDC tool?”



Myth buster 👻: you **don’t need** Kafka and Kafka Connect to run Debezium! You can embed it in your applications using the Debezium Engine, or target other sink types (*e.g.* Amazon Kinesis, Google Pub/Sub) using the Debezium Server.

Running Debezium under the hood

Tools that leverage the Debezium Engine or the Debezium Server can:

- **Abstract** some **complexity** of operating Debezium et. al from the end user.
- Enable advanced features like **schema evolution** using existing primitives.

Debezium

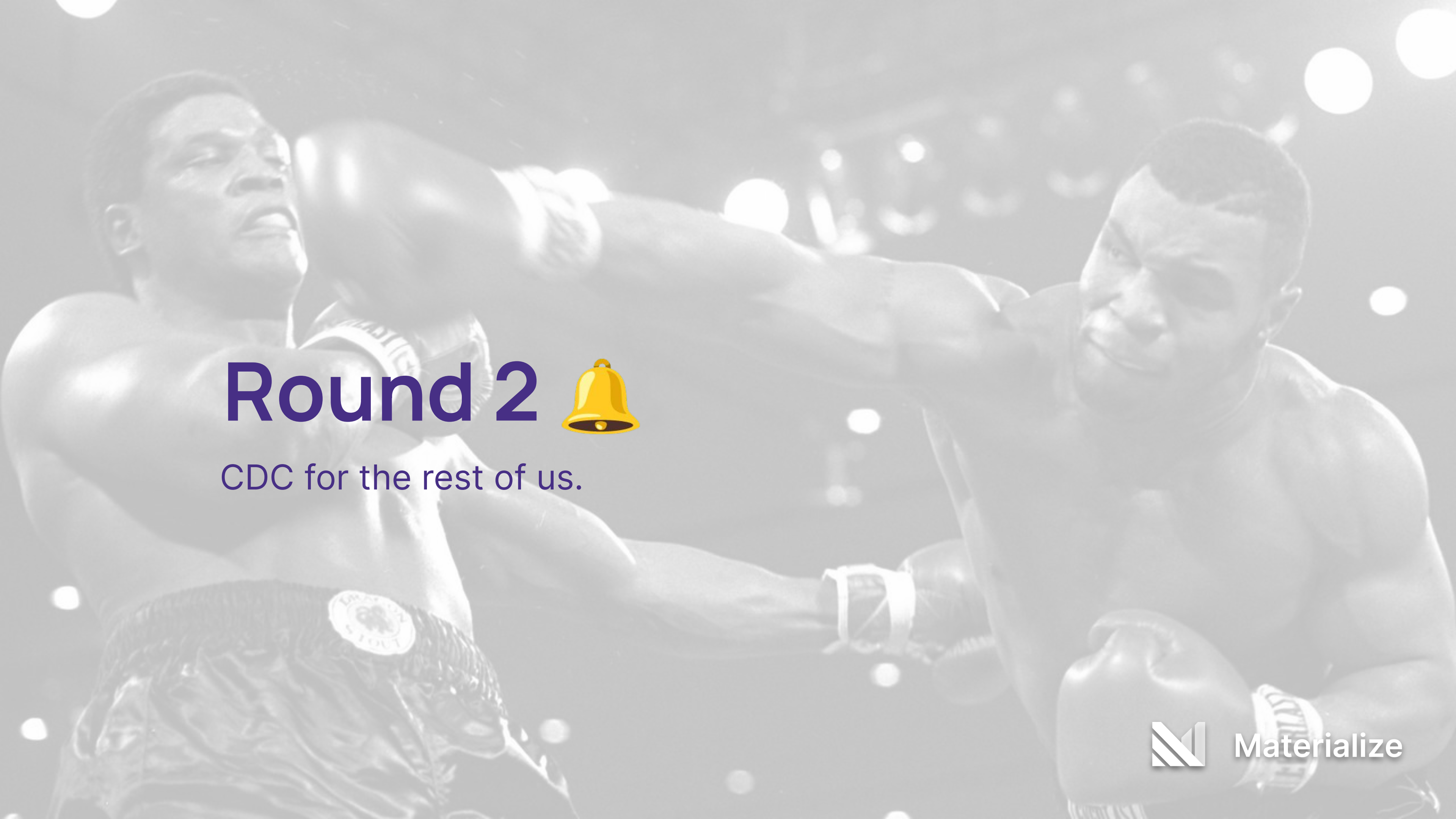
Examples

Streamkap

RisingWave CDC connectors

Flink CDC connectors

Confluent CDC connectors



Round 2 🔔

CDC for the rest of us.

Materialize

“Have you heard about streaming?”

Tools building support for CDC from scratch can:

- Create a **user experience** that is tailored to long-time **SQL** users.
- Have **more control** over **semantics**.

Examples

Artie

Estuary

Materialize

Acquisitions

HVR
(Fivetran)

Arcion
(Databricks)

Decision

Debezium isn't going anywhere...

Materialize

…but there's a whole world to explore!

Check out Materialize and our native PostgreSQL and MySQL CDC sources if you're considering streaming SQL!

Materialize