



# Change Data Streaming Patterns for Microservices With Debezium

Gunnar Morling  
Software Engineer



# Gunnar Morling

- Open source software engineer at Red Hat
  - **Debezium**
  - Hibernate
- **Spec Lead** for Bean Validation 2.0
- Other projects: **Deptective**, MapStruct
- Java Champion

✉️ [gunnar@hibernate.org](mailto:gunnar@hibernate.org)

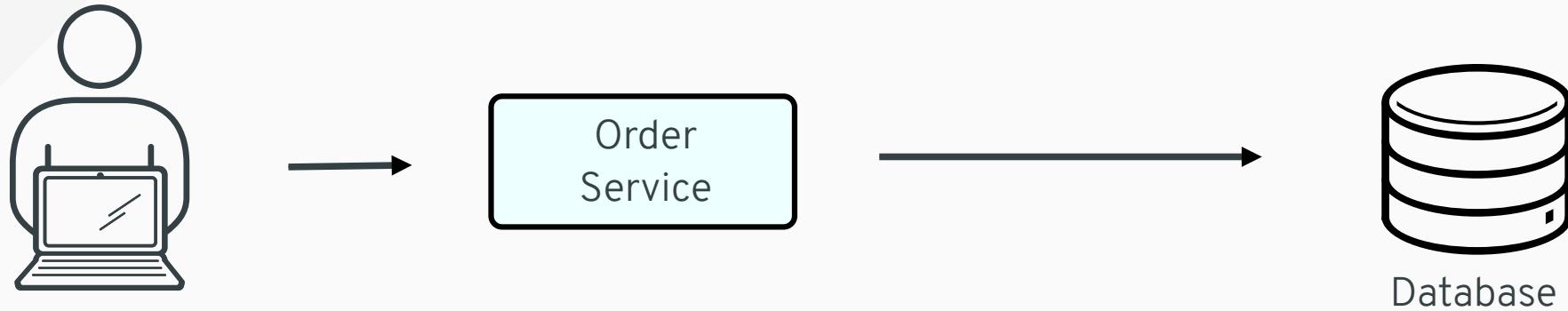
🐦 [@gunnarmorling](https://twitter.com/gunnarmorling)

🌐 <http://in.relation.to/gunnar-morling/>



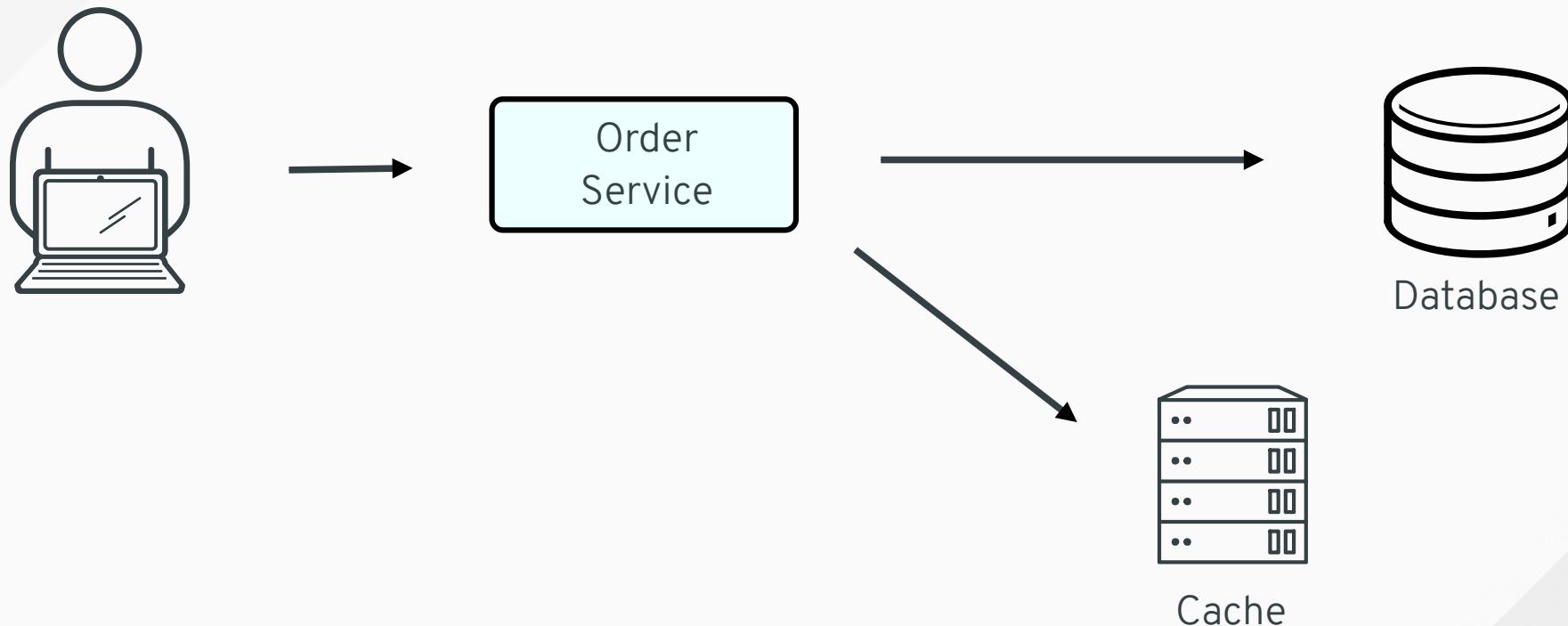
# A Common Problem

Updating Multiple Resources



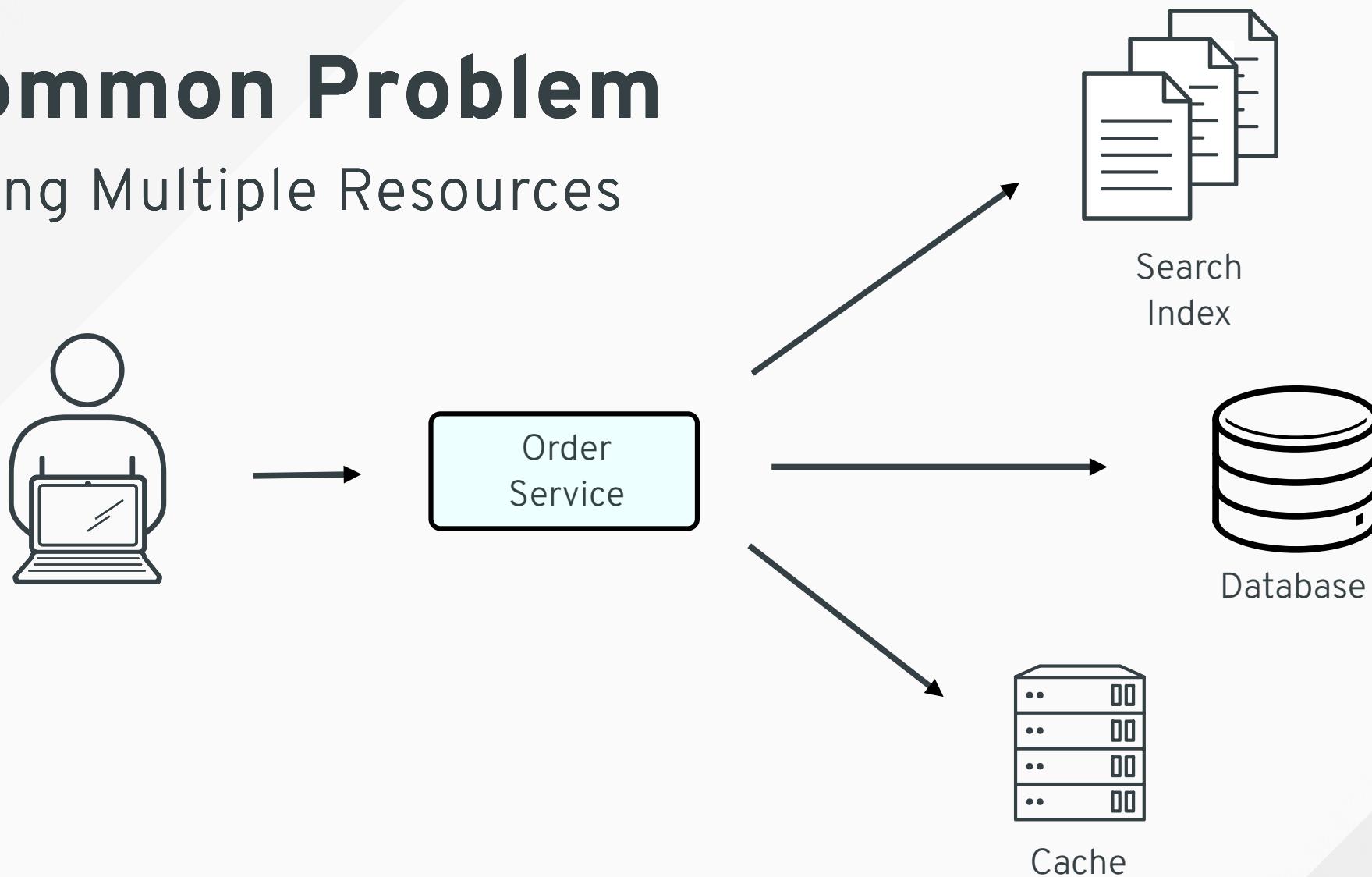
# A Common Problem

Updating Multiple Resources



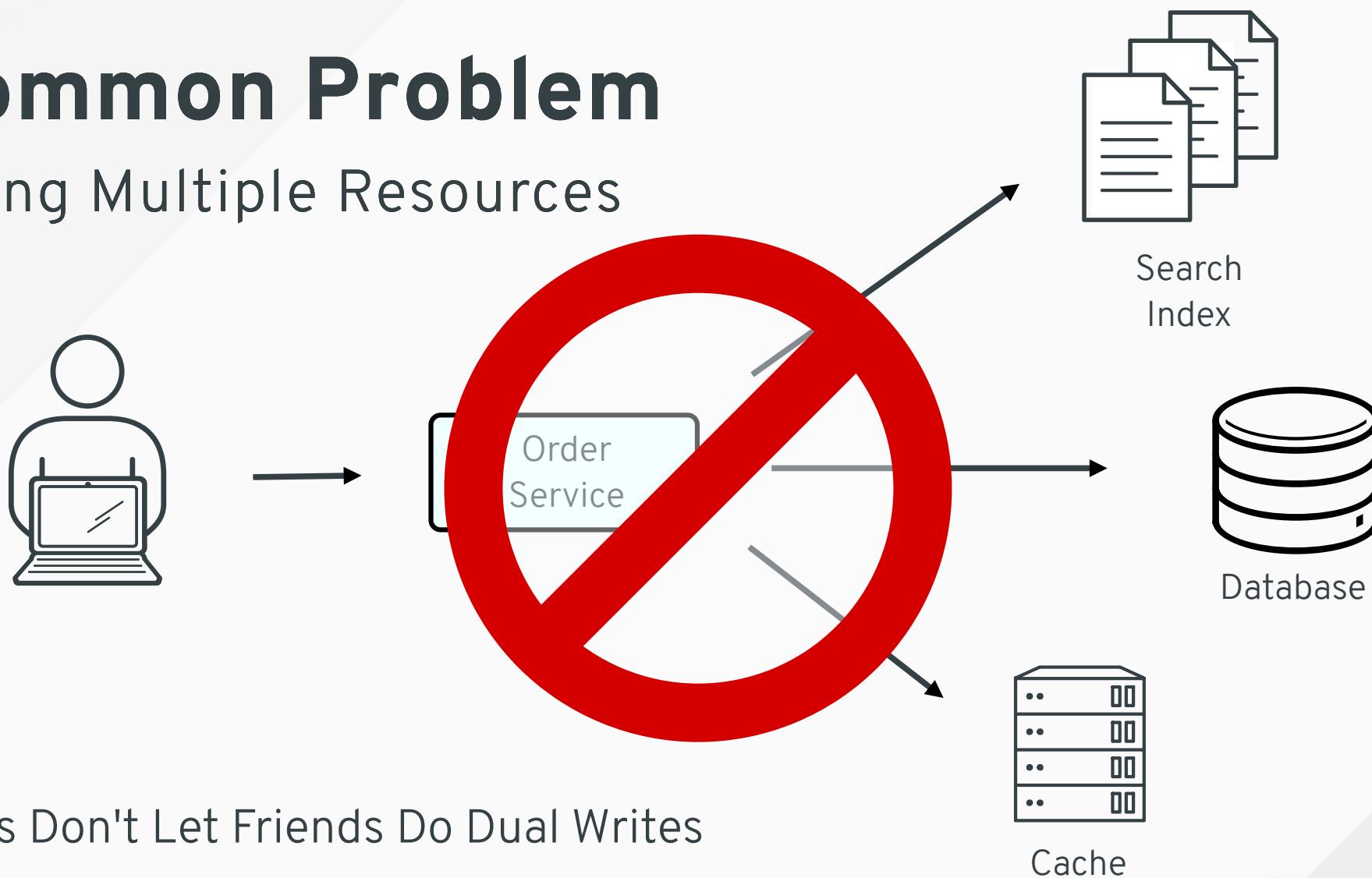
# A Common Problem

Updating Multiple Resources



# A Common Problem

Updating Multiple Resources



“ Friends Don't Let Friends Do Dual Writes

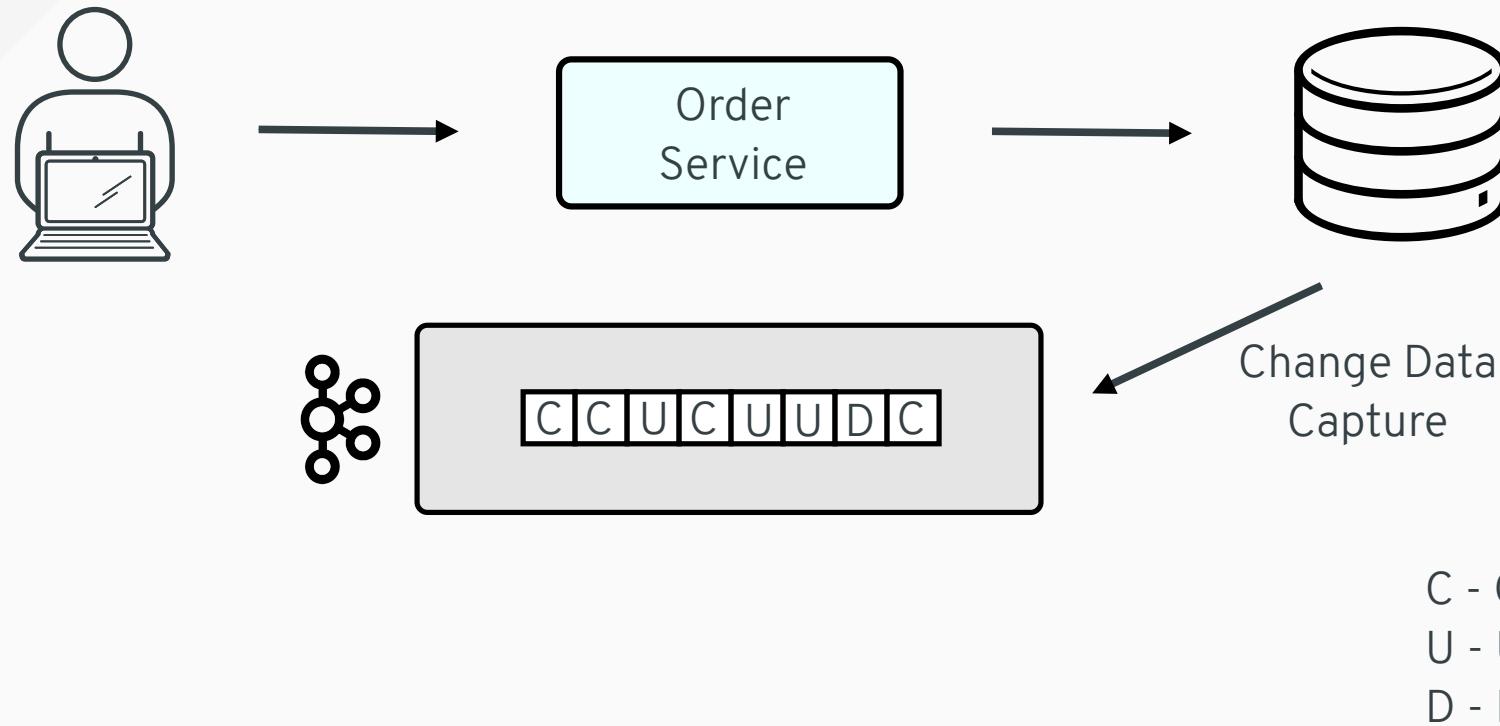
# A Better Solution

Streaming Change Events From the Database



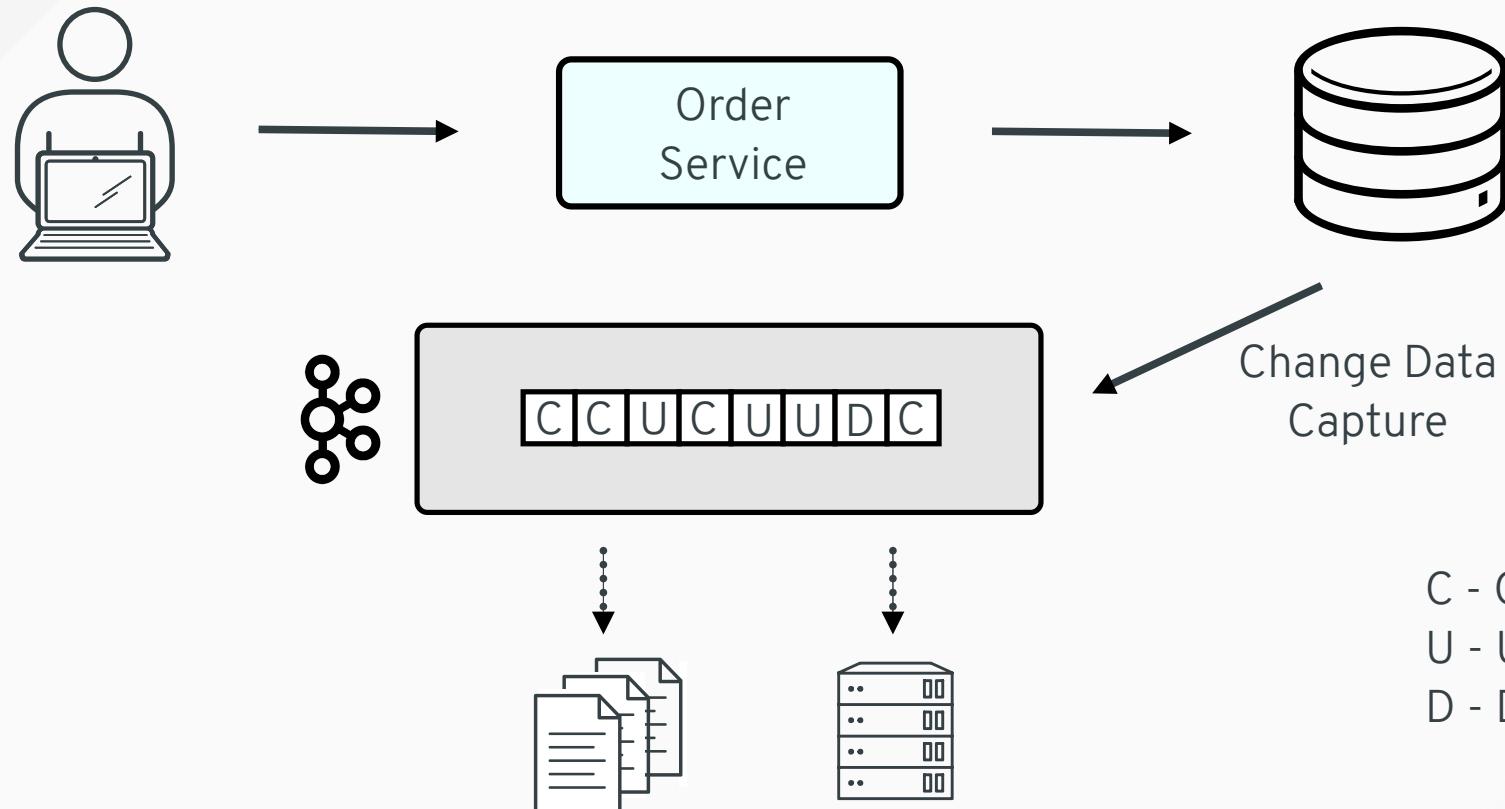
# A Better Solution

Streaming Change Events From the Database



# A Better Solution

Streaming Change Events From the Database



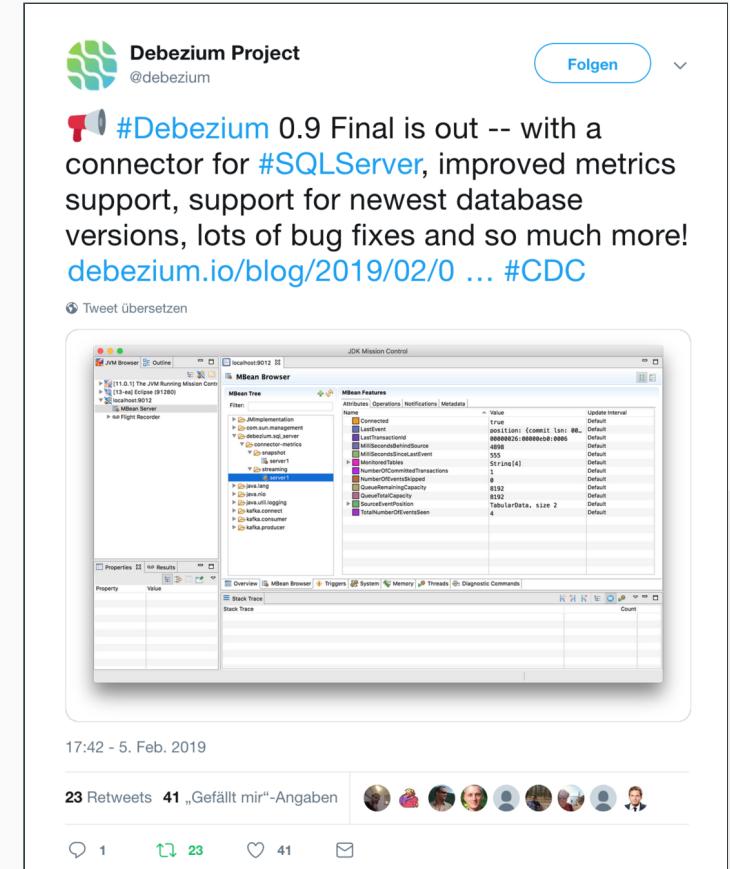
A perspective view of a long, modern escalator system in a subway station. The escalators are dark grey with white railings, moving upwards towards a bright, modern ceiling with recessed lighting. The station walls are light-colored and feature vertical architectural details. The overall atmosphere is clean and futuristic.

# Change Data Capture With Debezium

# Debezium

## Change Data Capture Platform

- Retrieves change events from **TX logs** from different DBs
  - Transparent to writing apps
  - Comprehensive **type support** (PostGIS etc.)
  - **Snapshotting**, Filtering etc.
- Fully open-source, very **active community**
- Latest version: 0.9 (based on **Kafka 2.2**)
- Production deployments at multiple companies (e.g. WePay, Trivago, BlaBlaCar etc.)



# Advantages of Log-based CDC

Tailing the transaction log

- **All data changes** are captured
- **No polling delay** or overhead
- **Transparent** to writing applications and models
- Can **capture deletes**
- Can capture **old record state** and further meta data
- Different formats/APIs, but Debezium deals with this

# Debezium

## CDC Use Cases

- Update or invalidate **caches**
- Enable **full-text search** via Elasticsearch, Solr etc.
- Data **replication**
- **Microservices** data exchange
- **Auditing/historization**
- Update **CQRS** read models
- Enable **streaming queries**

# Change Event Structure

- Key: PK of table
- Value: Describing the change event
  - **Before** state,
  - **After** state,
  - **Source** info
- Serialization formats:
  - JSON
  - Avro

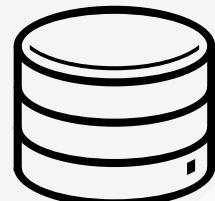
```
{  
    "before": null,  
    "after": {  
        "id": 1004,  
        "first_name": "Anne",  
        "last_name": "Kretchmar",  
        "email": "annek@noanswer.org"  
    },  
    "source": {  
        "name": "dbserver1",  
        "server_id": 0,  
        "ts_sec": 0,  
        "file": "mysql-bin.000003",  
        "pos": 154,  
        "row": 0,  
        "snapshot": true,  
        "db": "inventory",  
        "table": "customers"  
    },  
    "op": "c",  
    "ts_ms": 1486500577691  
}
```

# Debezium Connectors

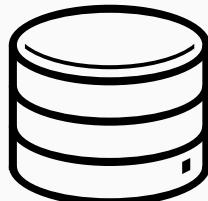
- MySQL
- Postgres
- MongoDB
- SQL Server
- Oracle (Tech Preview, based on XStream)
- Possible future additions
  - Cassandra?
  - MariaDB?



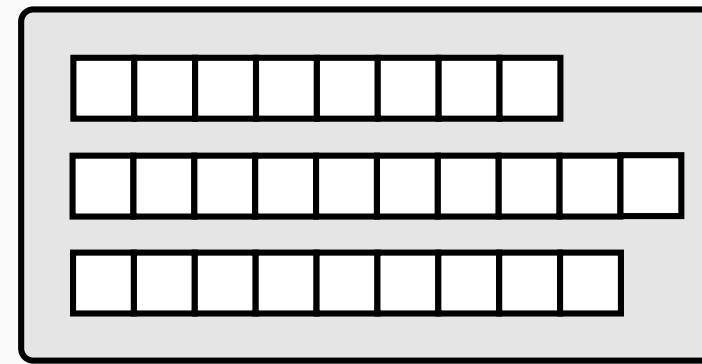
# CDC with Debezium and Kafka Connect



MySQL

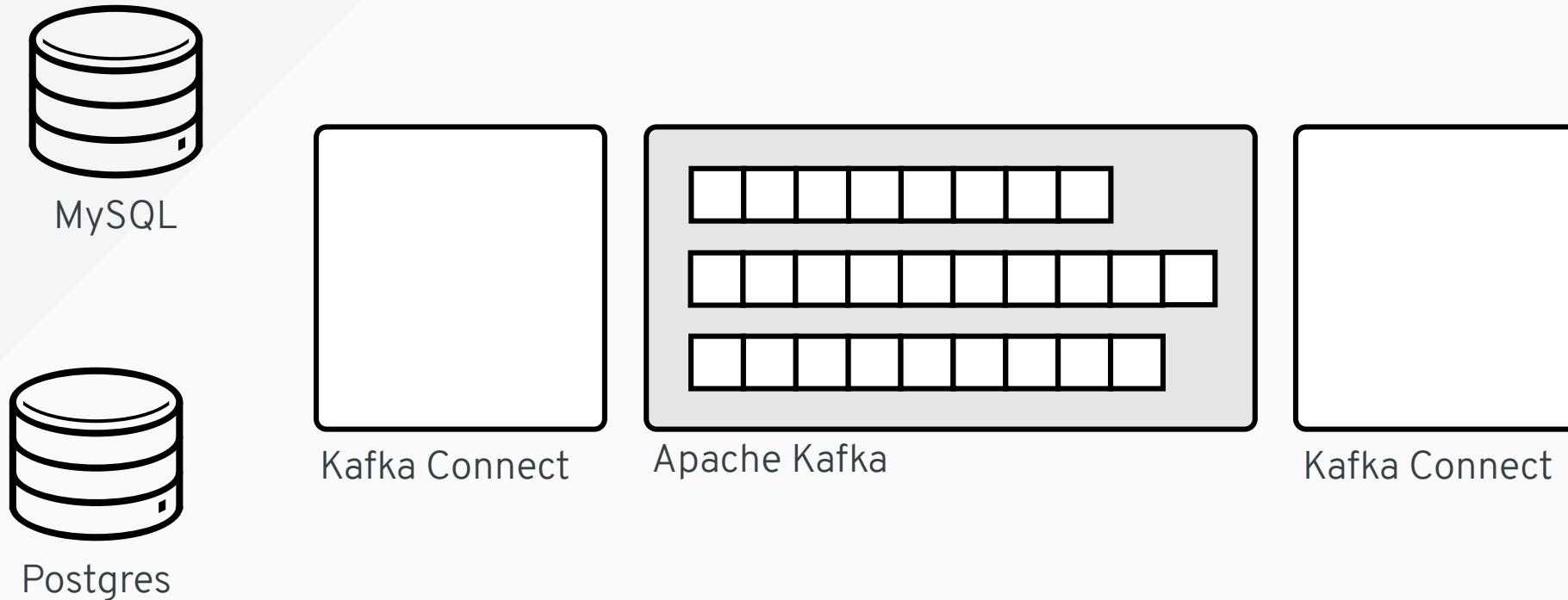


Postgres

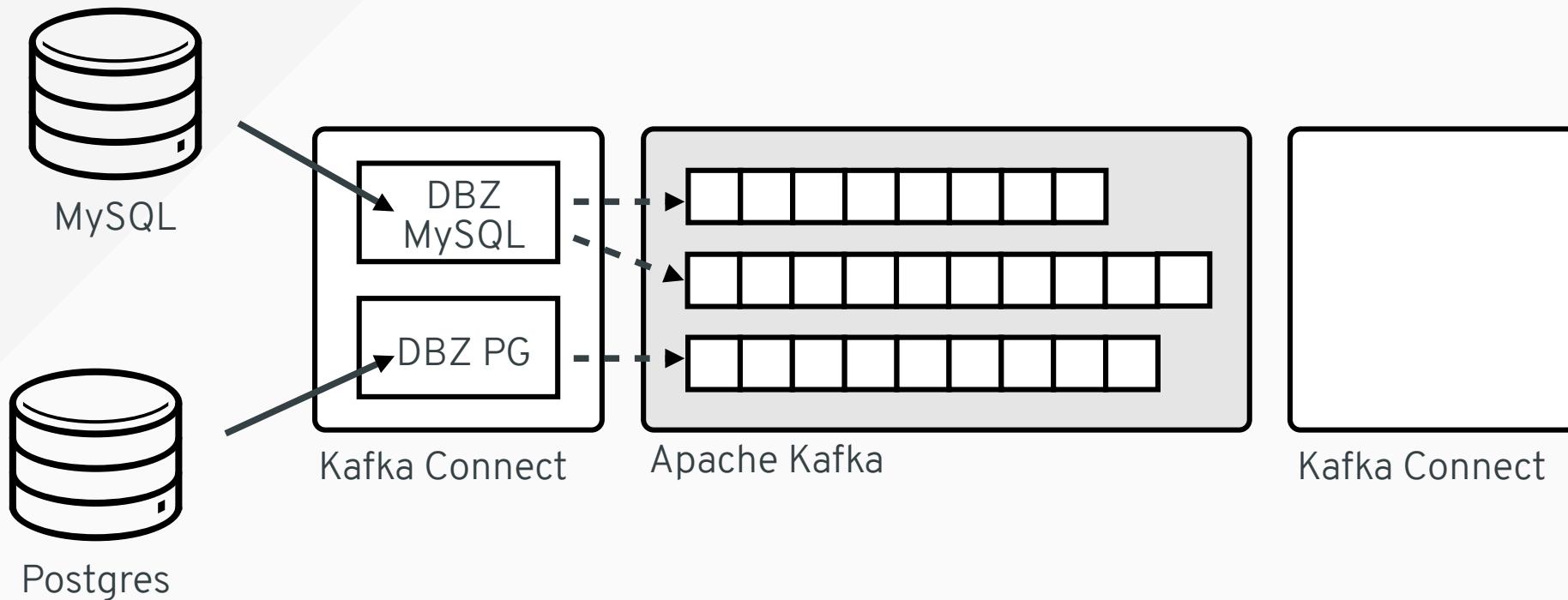


Apache Kafka

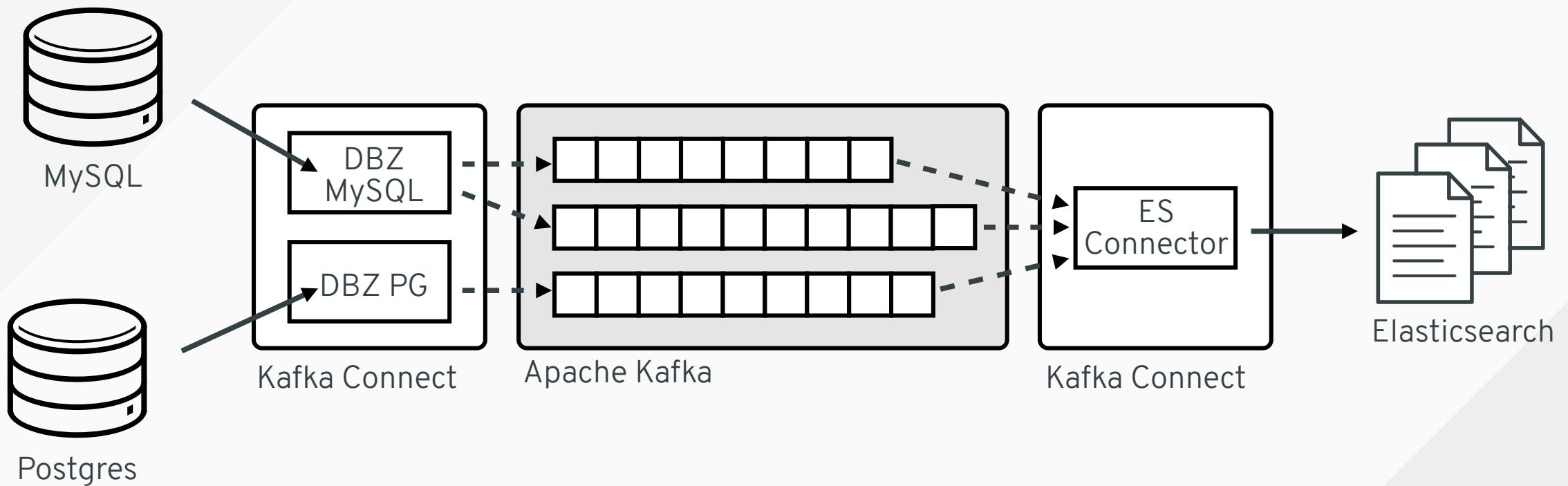
# CDC with Debezium and Kafka Connect



# CDC with Debezium and Kafka Connect



# CDC with Debezium and Kafka Connect



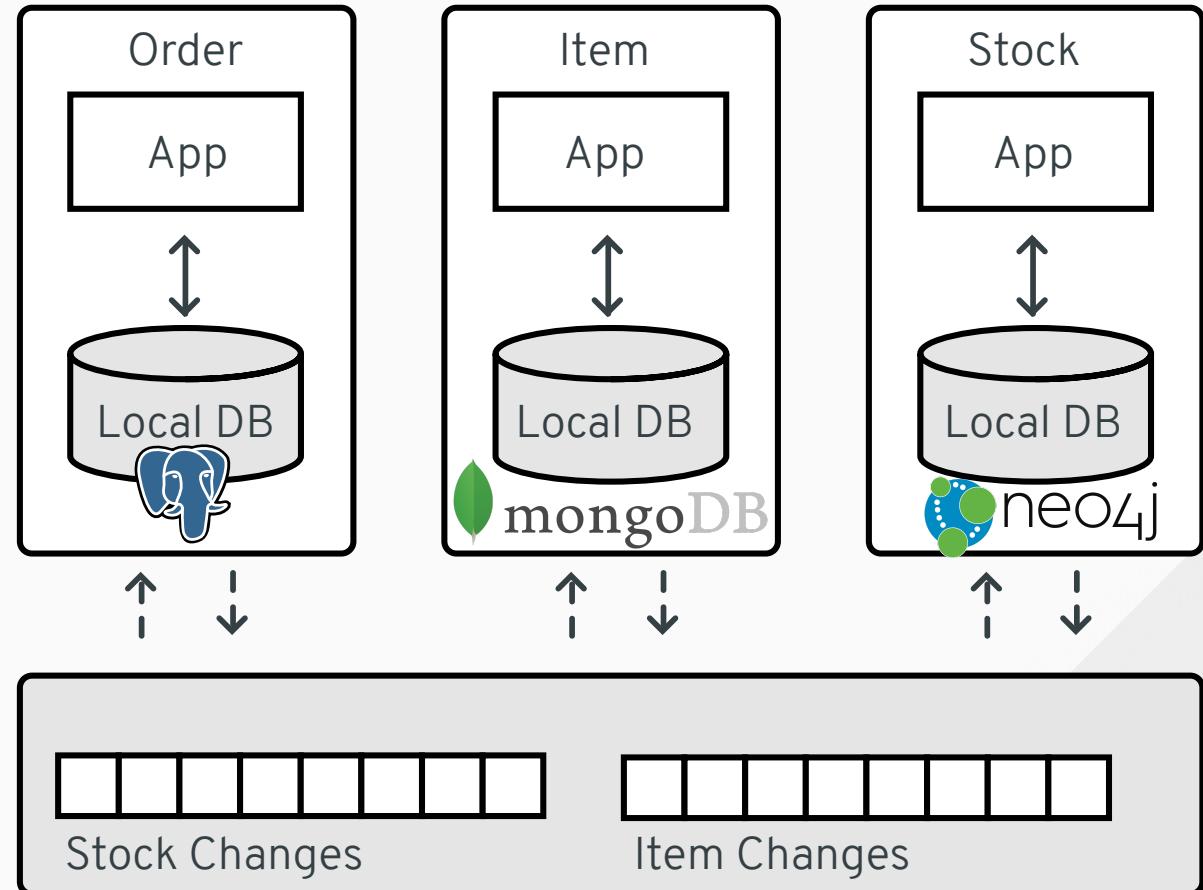
The background of the image is a night sky filled with stars. In the foreground, there are dark silhouettes of mountains. A winding road is visible at the bottom, with its lights creating long, streaky lines that curve through the darkness.

# Microservice CDC Patterns

# Pattern: Microservice Data Synchronization

## Microservice Architectures

- Propagate data between different services **without coupling**
- Each service keeps **optimised views locally**

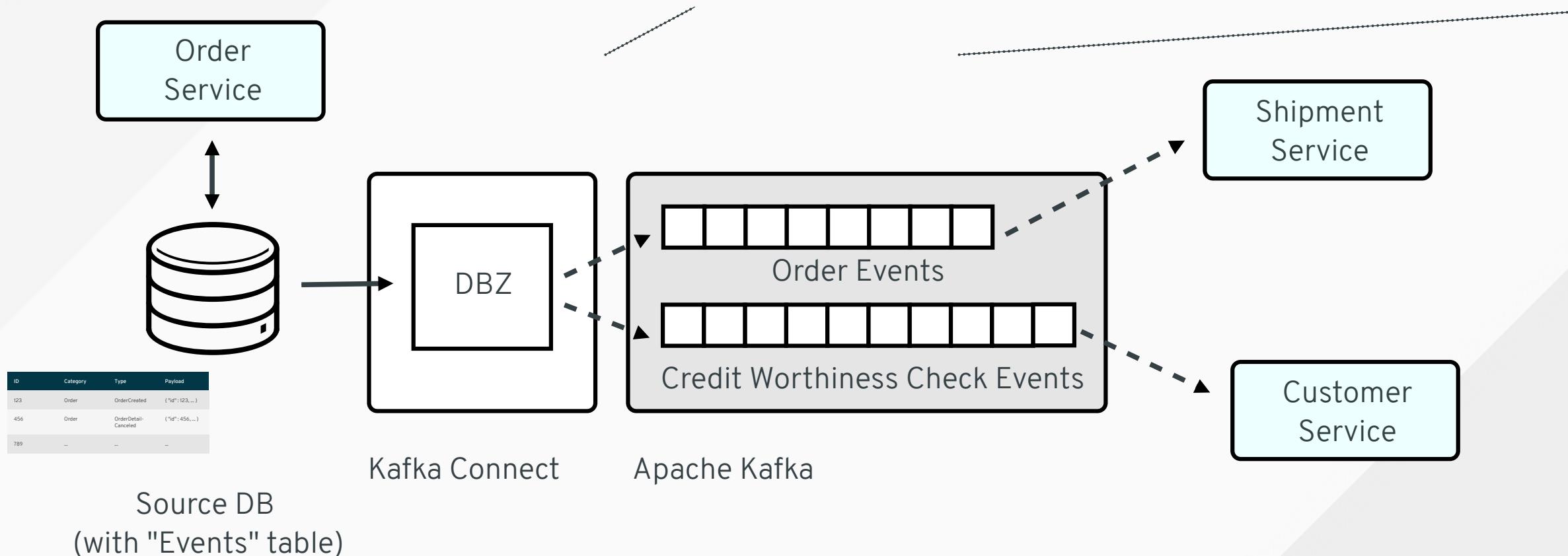


# Pattern: Outbox

## Separate Events Table

"Outbox" table

Id	AggregateType	AggregateId	Type	Payload
ec6e	Order	123	OrderCreated	{ "id": 123, ... }
8af8	Order	456	OrderDetailCanceled	{ "id": 456, ... }
890b	Customer	789	InvoiceCreated	{ "id": 789, ... }



# Pattern: Microservice Extraction

Migrating from Monoliths to Microservices

- **Extract microservice** for single component(s)
- Keep write requests **against running monolith**
- **Stream changes** to extracted microservice
- Test new functionality
- **Switch over**, evolve schema only afterwards

# Pattern: Leverage the Powers of SMTs

## Single Message Transformations

- **Aggregate** sharded tables to single topic
- **Keep compatibility** with existing consumers
- **Format conversions**, e.g. for dates
- Ensure compatibility with sink connectors
  - Extracting "after" state only
  - Expand MongoDB's JSON structures

# Pattern: Ensuring Data Quality

## Detecting Missing or Wrong Data

- Constantly **compare record counts** on source and sink side
  - Raise alert if threshold is reached
- Compare every n-th record **field by field**
  - E.g. have all records compared within one week

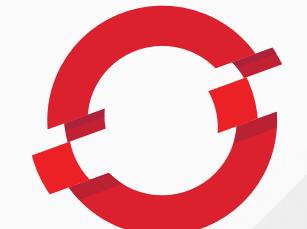
The background of the image is a wide-angle photograph of a rural landscape. It features rolling green hills that stretch across the horizon. In the middle ground, there is a long, white bridge or overpass spanning a valley. The sky above is a clear, pale blue with some wispy, white clouds. The overall scene is peaceful and suggests a natural environment.

**Demo**

# Running Debezium on Kubernetes

AMQ Streams: Enterprise Distribution of Apache Kafka

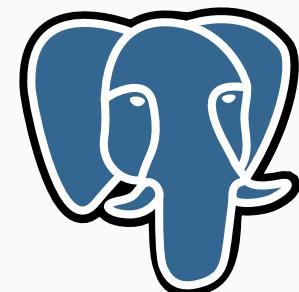
- Provides
  - **Container images** for Apache Kafka, Connect, Zookeeper and MirrorMaker
  - **Operators** for managing/configuring Apache Kafka clusters, topics and users
  - Kafka Consumer, Producer and Admin clients, Kafka Streams
- Supported by Red Hat
- Upstream Community: **Strimzi**



# Support for Debezium

## Red Hat Integration

- Debezium is being **productized** as part of the **Red Hat Integration product**
- Initially **Microsoft SQL Server, MySQL, PostgreSQL, and MongoDB** connectors
- Integrated with **AMQ Streams**
- **Developer Preview** to be released soon; general availability (GA) planned for later this year



# Summary

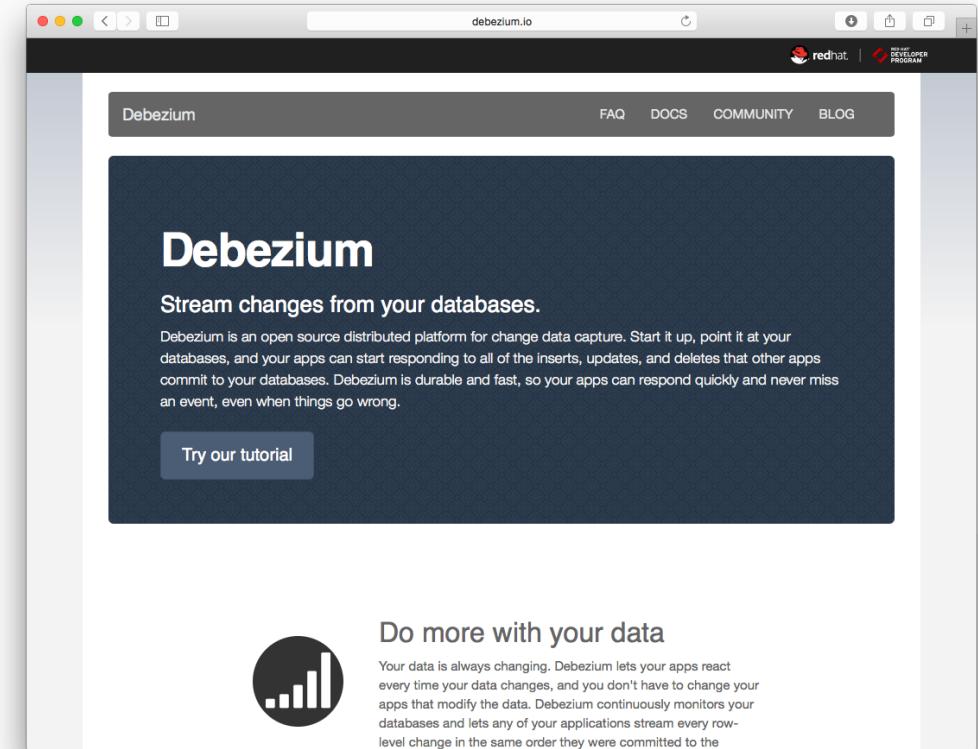
- **CDC** enables use cases such as replication, microservices data exchange and much more
- **Debezium:** CDC for a growing number of databases
- **Contributions welcome!**
- Tell us about your **feature requests and ideas!**



“ Friends Don't Let Friends Do Dual Writes

# Resources

- **Website:** <http://debezium.io/>
- **Source code**, examples, Compose files etc.  
<https://github.com/debezium>
- **Discussion group**  
<https://groups.google.com/forum/#!forum/debezium>
- **Strimzi** (Kafka on Kubernetes/OpenShift)  
<http://strimzi.io/>
- **Latest news:**  @debezium





**Red Hat**