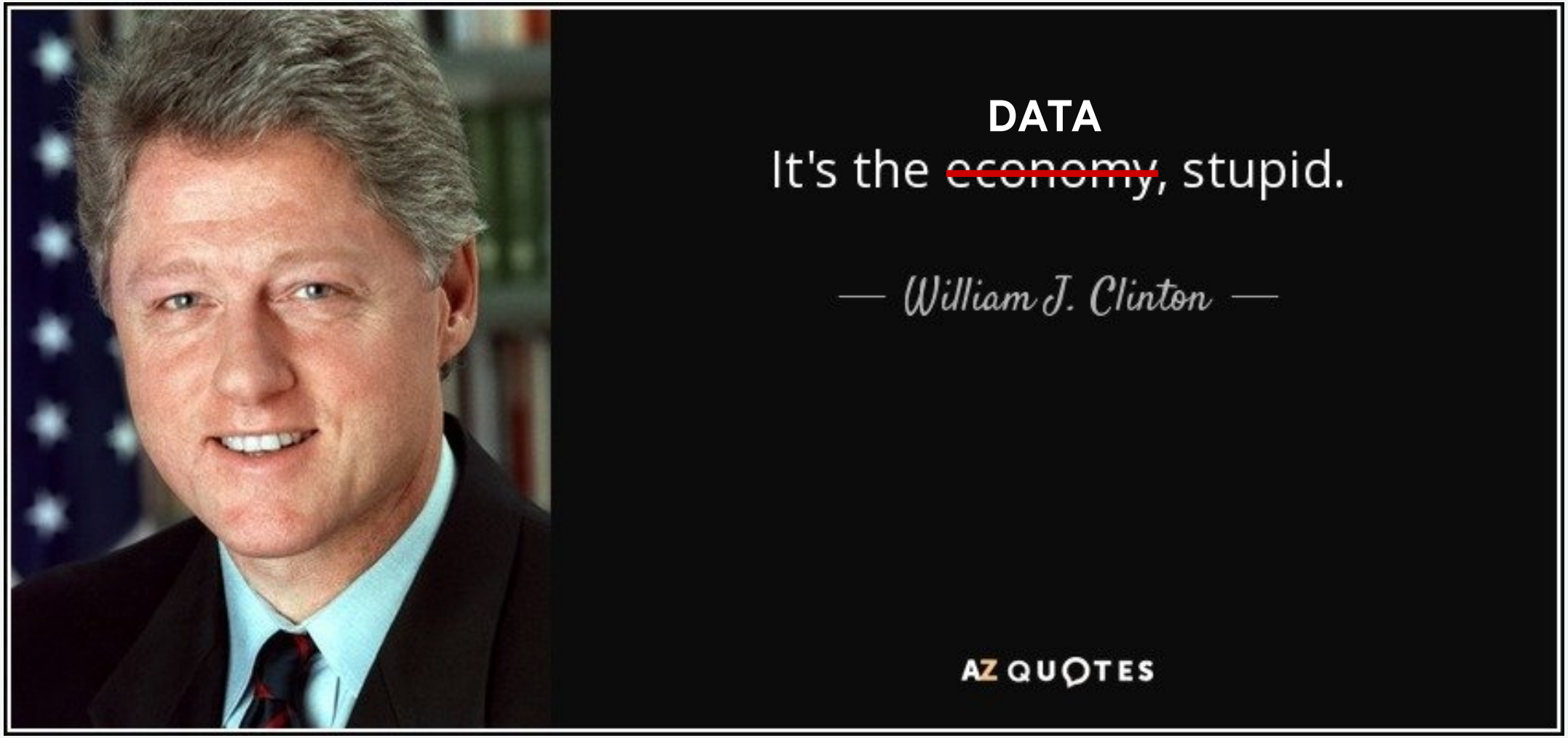
Joker<?>

Practical Change Data Streaming Use Cases With Apache Kafka and Debezium

Gunnar Morling

Software Engineer


DATA
It's the economy, stupid.
— William J. Clinton —
AZ QUOTES


DATA, DATA, DATA!


OOPS, MY BAD...
WRONG DATA
memegenerator.net

The Issue with Dual Writes

What's the problem?
Change data capture to the rescue!

1

CDC Use Cases & Patterns

2

Replication
Audit Logs
Microservices

Practical Matters

3

Deployment Topologies
Running on Kubernetes
Single Message Transforms

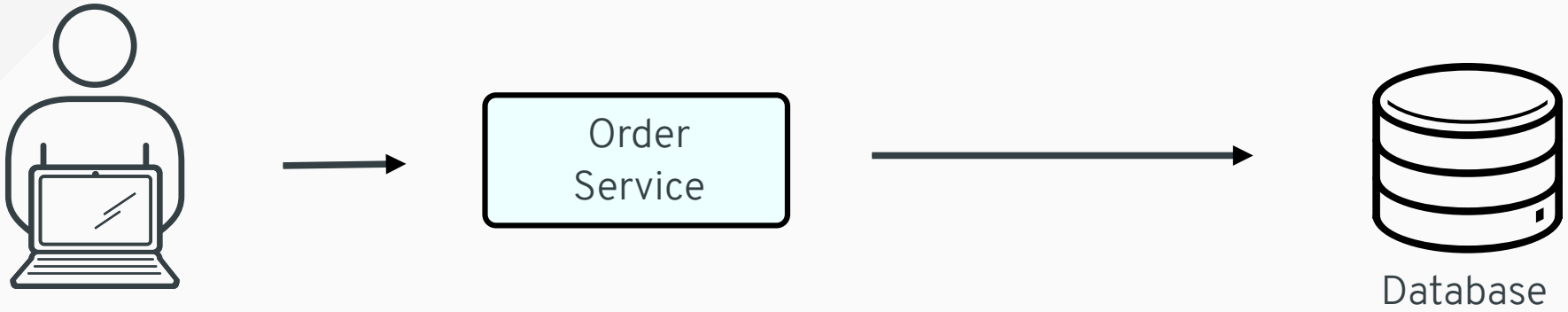
Gunnar Morling

- Open source software engineer at Red Hat
  - **Debezium**
  - Hibernate
- **Spec Lead** for Bean Validation 2.0 

- Other projects: **Deptective**, MapStruct
- Java Champion 

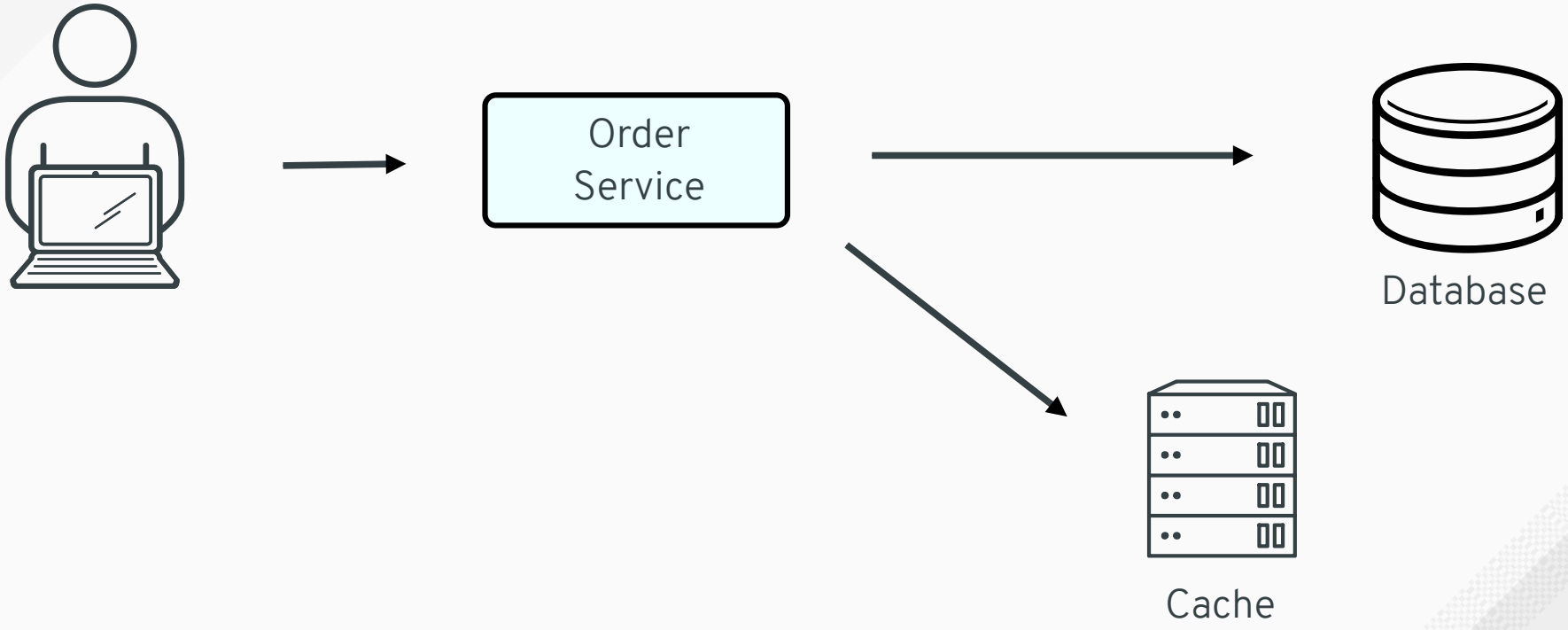
A Common Problem

Updating Multiple Resources



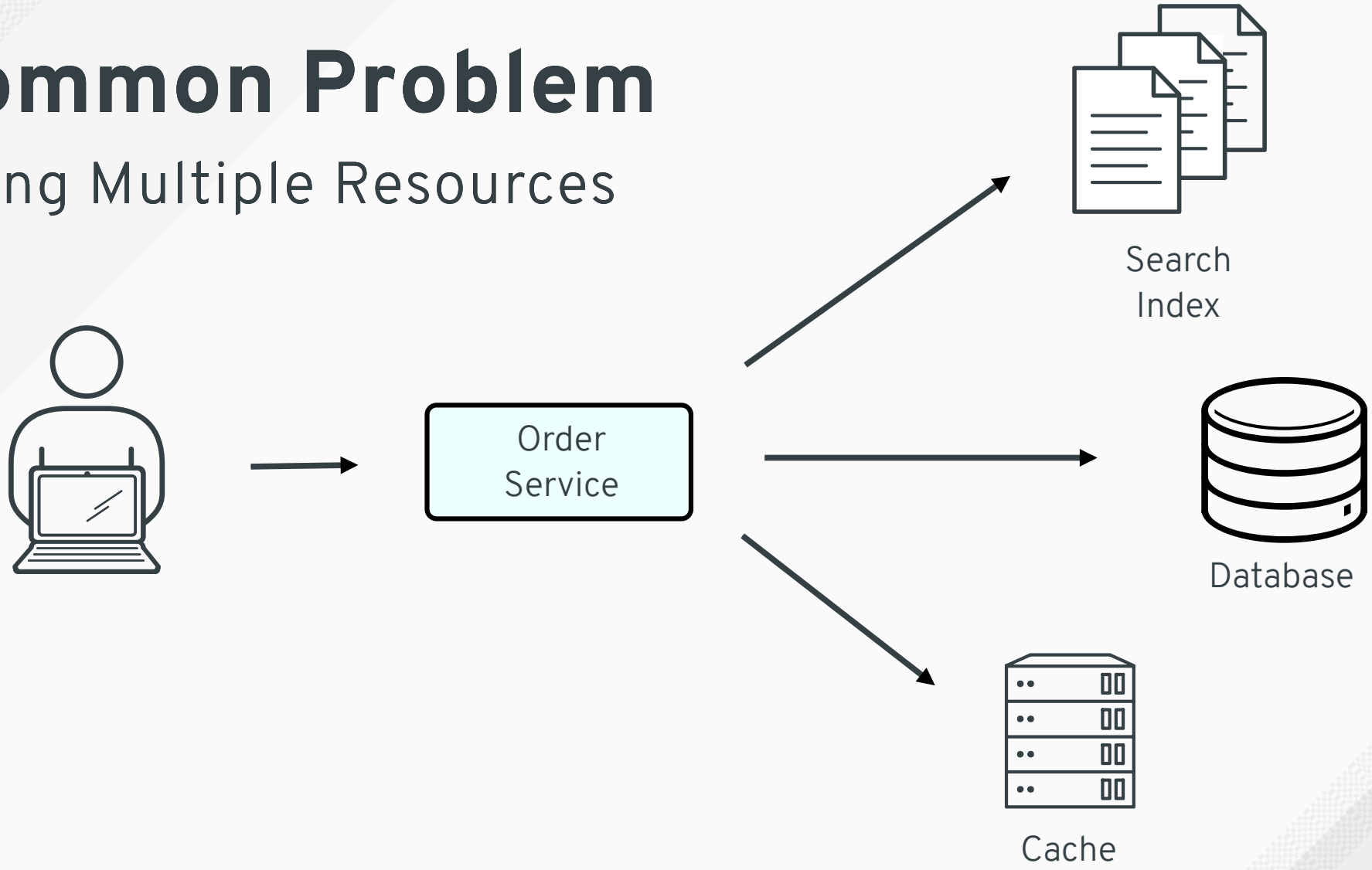
A Common Problem

Updating Multiple Resources



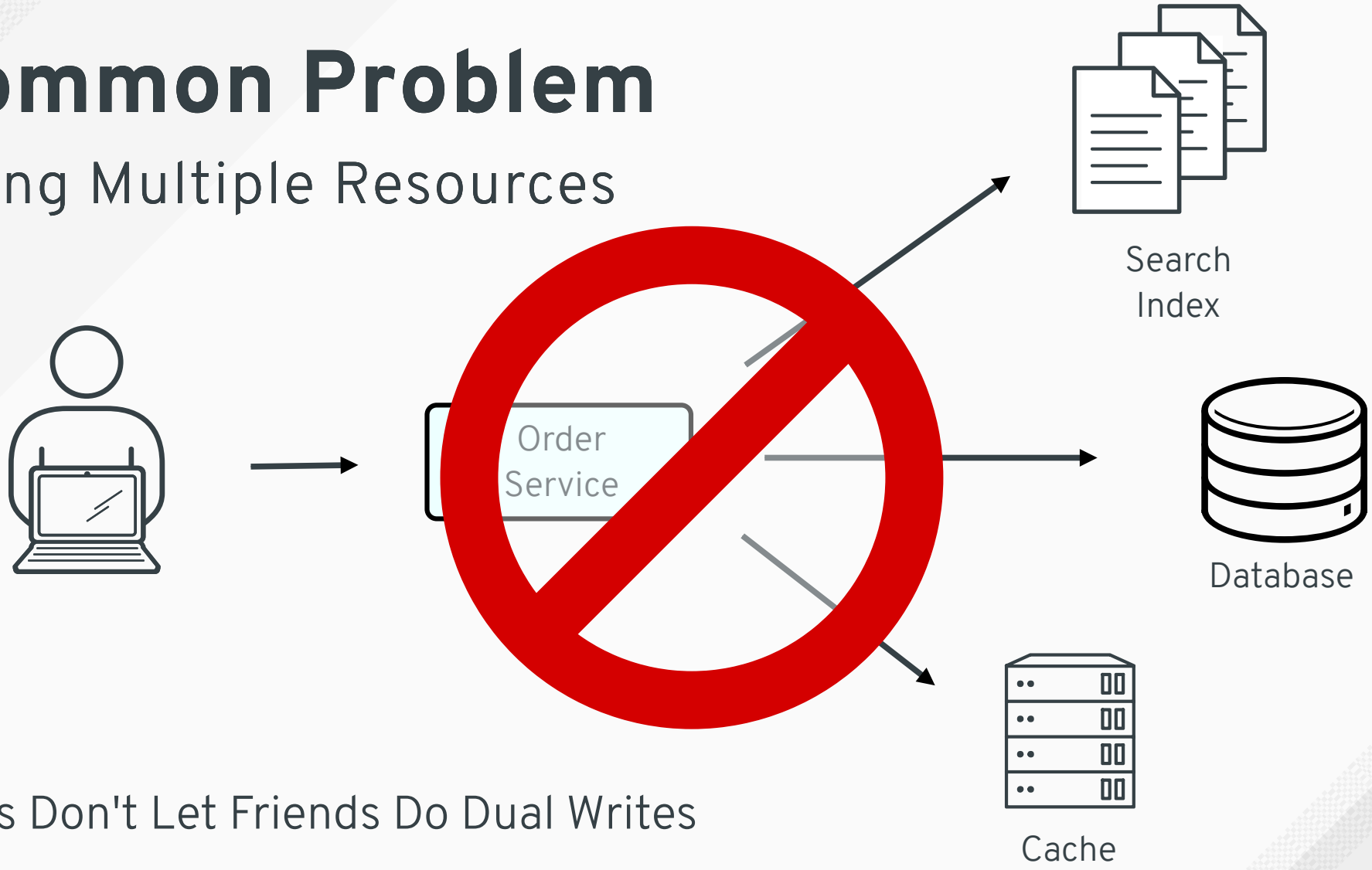
A Common Problem

Updating Multiple Resources



A Common Problem

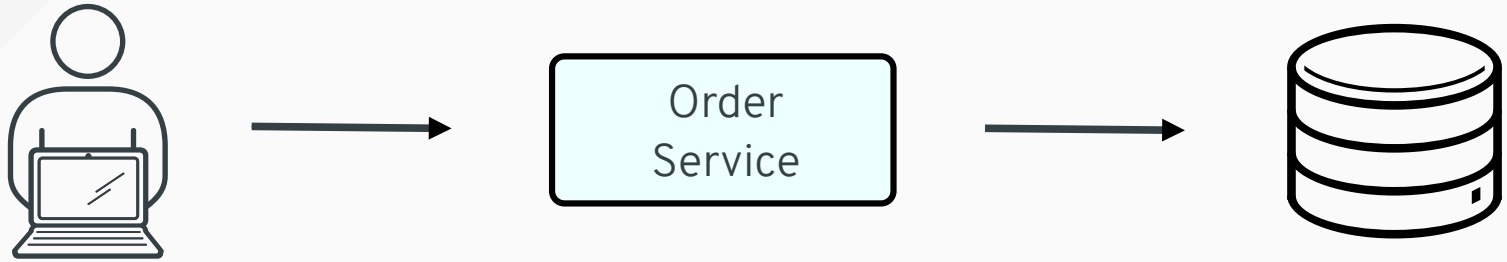
Updating Multiple Resources



“ Friends Don't Let Friends Do Dual Writes

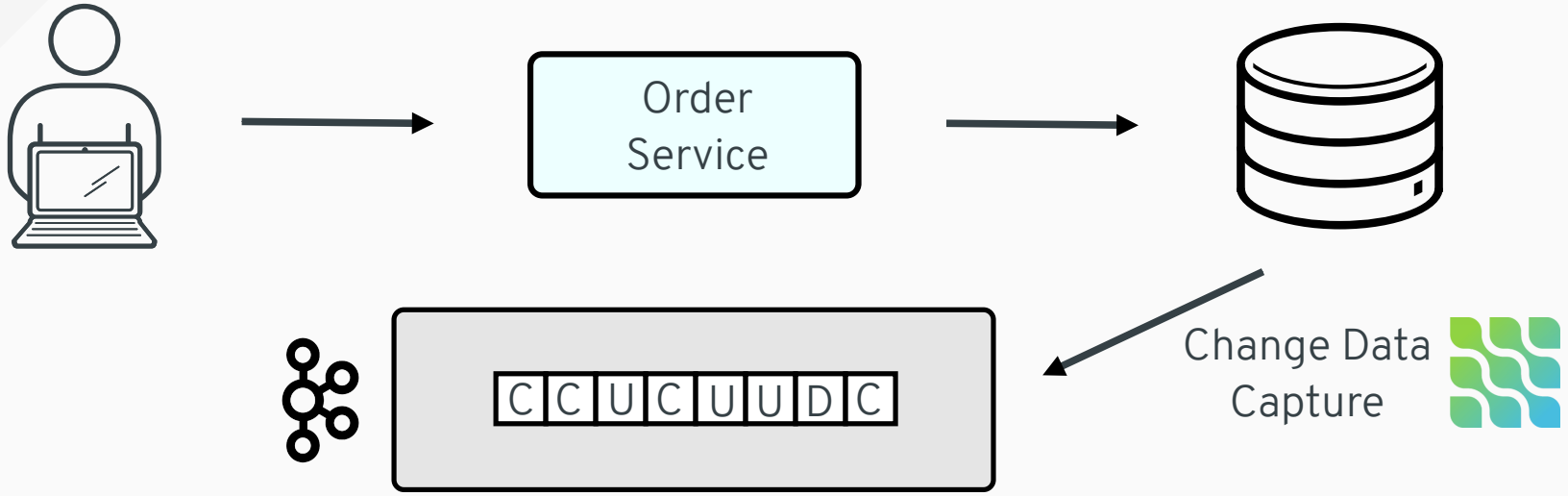
A Better Solution

Streaming Change Events From the Database



A Better Solution

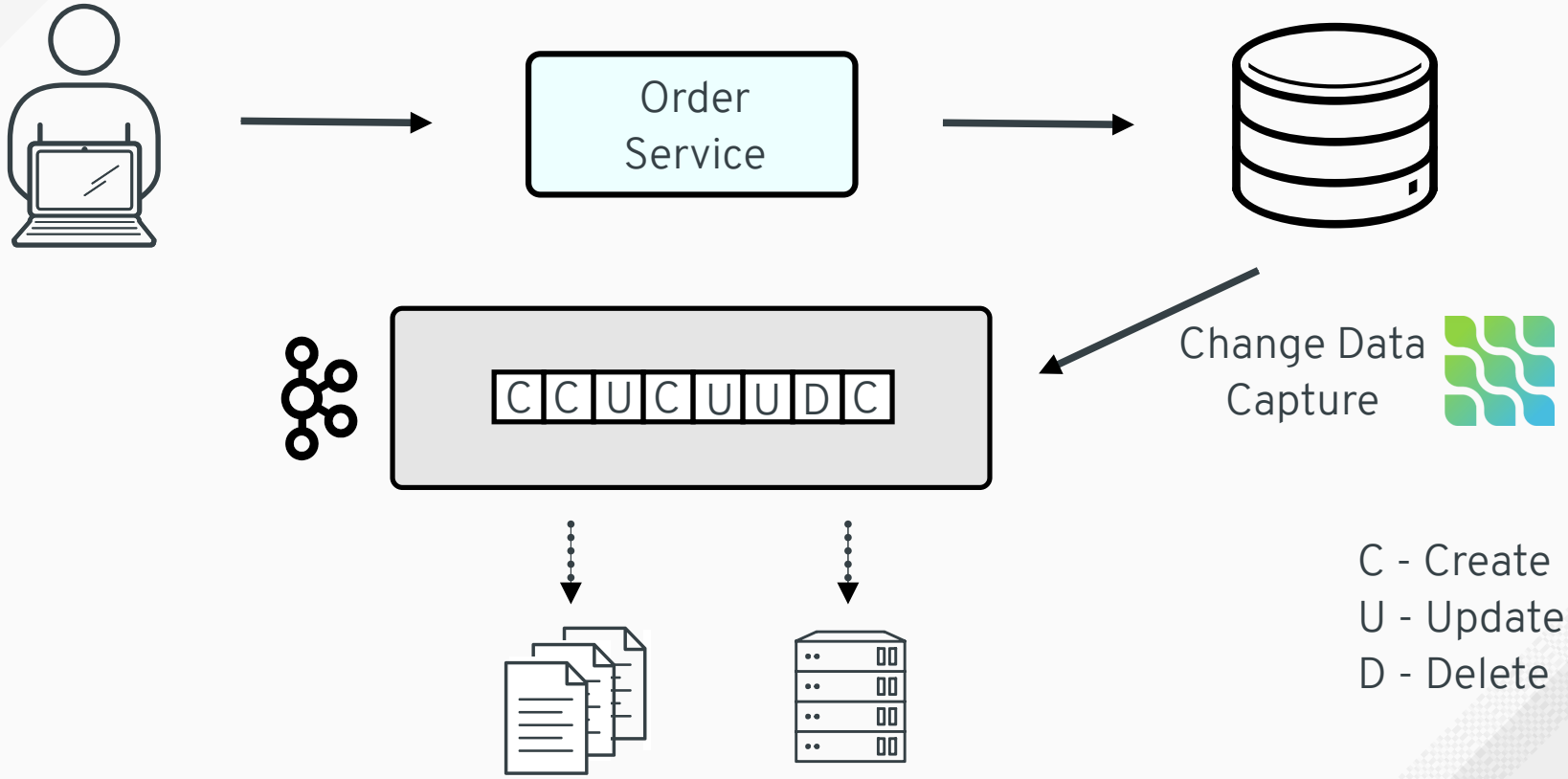
Streaming Change Events From the Database



C - Create
U - Update
D - Delete

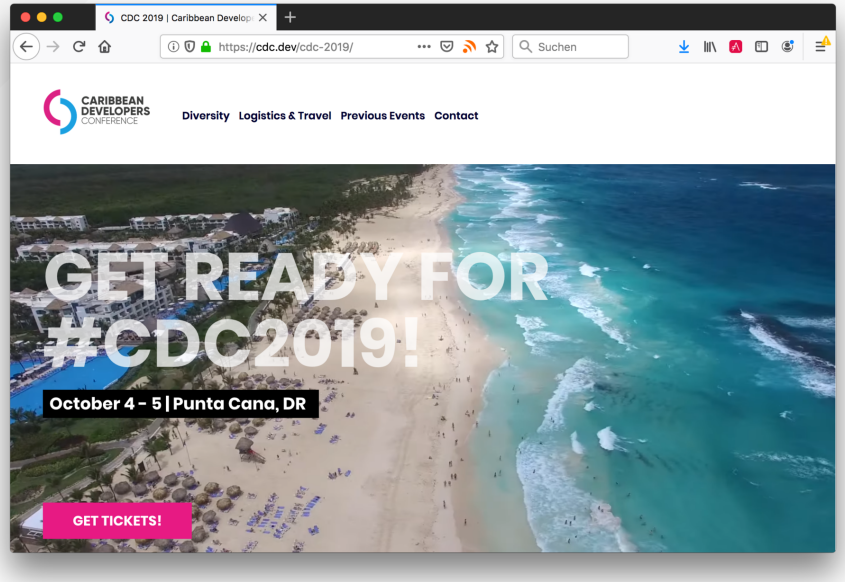
A Better Solution

Streaming Change Events From the Database



CDC = …

- **C**onsumer-**D**riven **C**ontracts?
- **C**enters for **D**isease **C**ontrol and Prevention?
- **C**aribbean **D**evelopers **C**onference?



© Brick Pics https://flic.kr/p/q5YgYD
© https://cdc.dev/

CDC = …

- **C**onsumer-**D**riven **C**ontracts?
- **C**enters for **D**isease **C**ontrol and Prevention?
- **C**aribbean **D**evelopers **Con**ference?

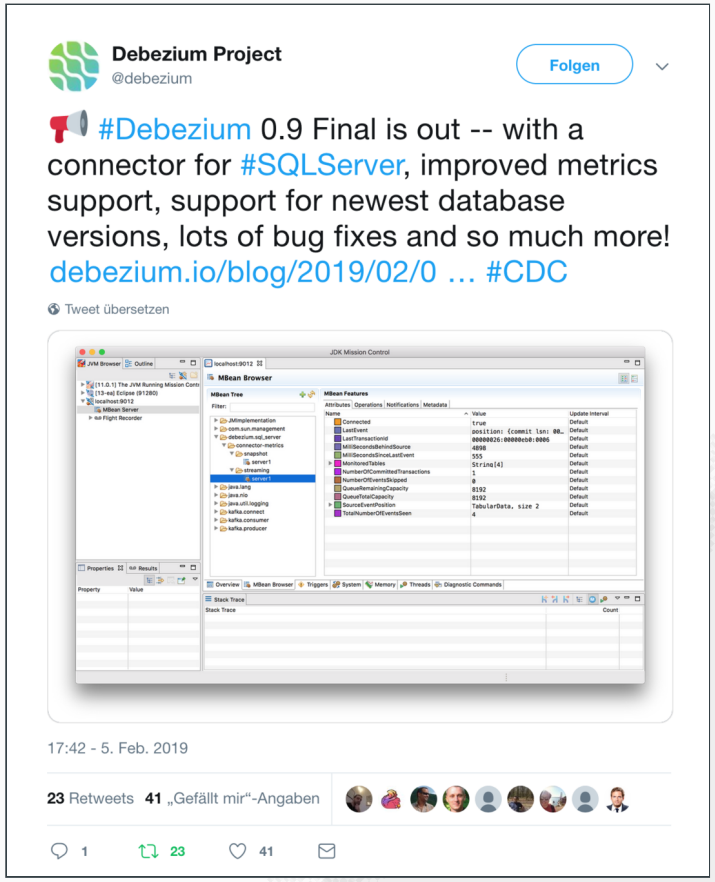
Change Data Capture!


Change Data Capture
With Debezium

Debezium

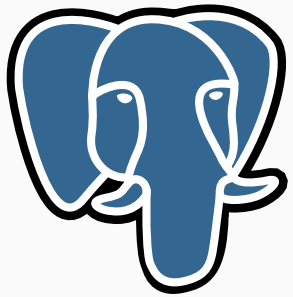
Change Data Capture Platform

- Log-based CDC for multiple databases
  - Comprehensive **type support** (PostGIS etc.)
  - **Snapshotting**, Filtering etc.
- Via Apache Kafka or embedded
- Fully open-source, very **active community**
- Production deployments at multiple companies (e.g. WePay, Convoy, JW Player, Usabilla, BlaBlaCar etc.)



Debezium Connectors

- **MySQL**
- **Postgres**
- **MongoDB**
- **SQL Server**
- Cassandra (Incubating)
- Oracle (Incubating, based on XStream)
- Possible future additions
  - DB2?
  - MariaDB?



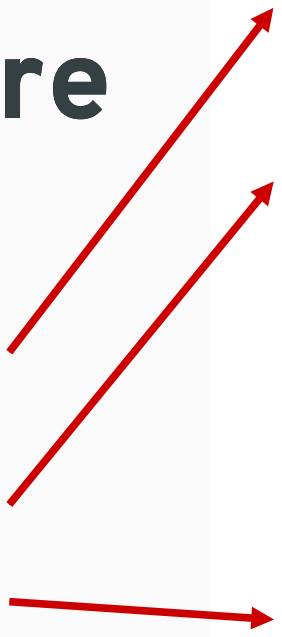
Meme idea: Robin Moffatt

Log- vs. Query-Based CDC

	Query-Based	Log-Based
All data changes are captured	-	+
No polling delay or overhead	-	+
Transparent to writing applications and models	-	+
Can **capture deletes** and **old record state**	-	+
Installation/Configuration	+	-

Change Event Structure

- Key: Primary key of table
- Value: Describing the change event
  - **Old** row state
  - **New** row state
  - **Metadata**
- 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
}
```

The Issue with Dual Writes

What's the problem?
Change data capture to the rescue!

1

CDC Use Cases & Patterns

2

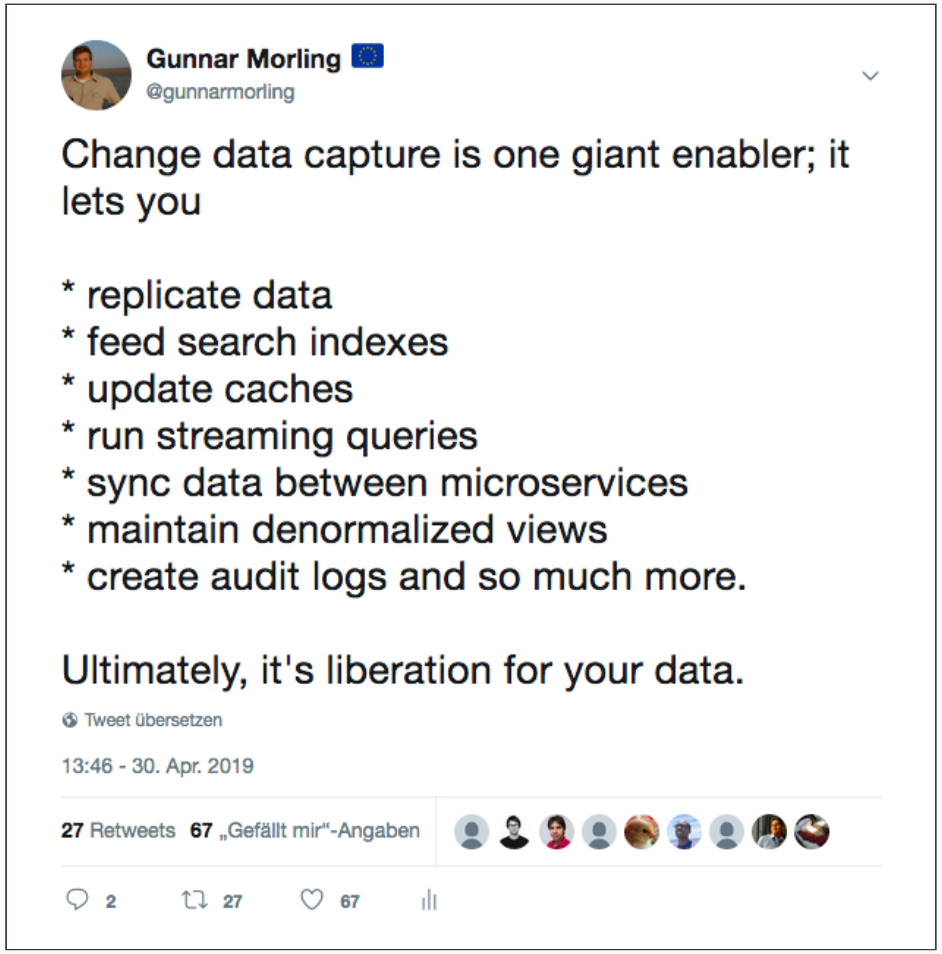
Replication
Audit Logs
Microservices

Practical Matters

3

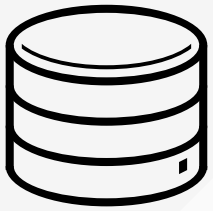
Deployment Topologies
Running on Kubernetes
Single Message Transforms

CDC – "Liberation for Your Data"

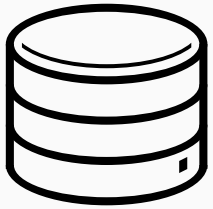


Data Replication

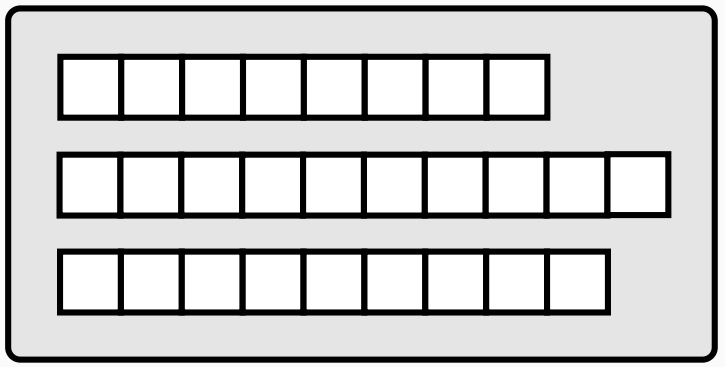
Zero-Code Streaming Pipelines



MySQL



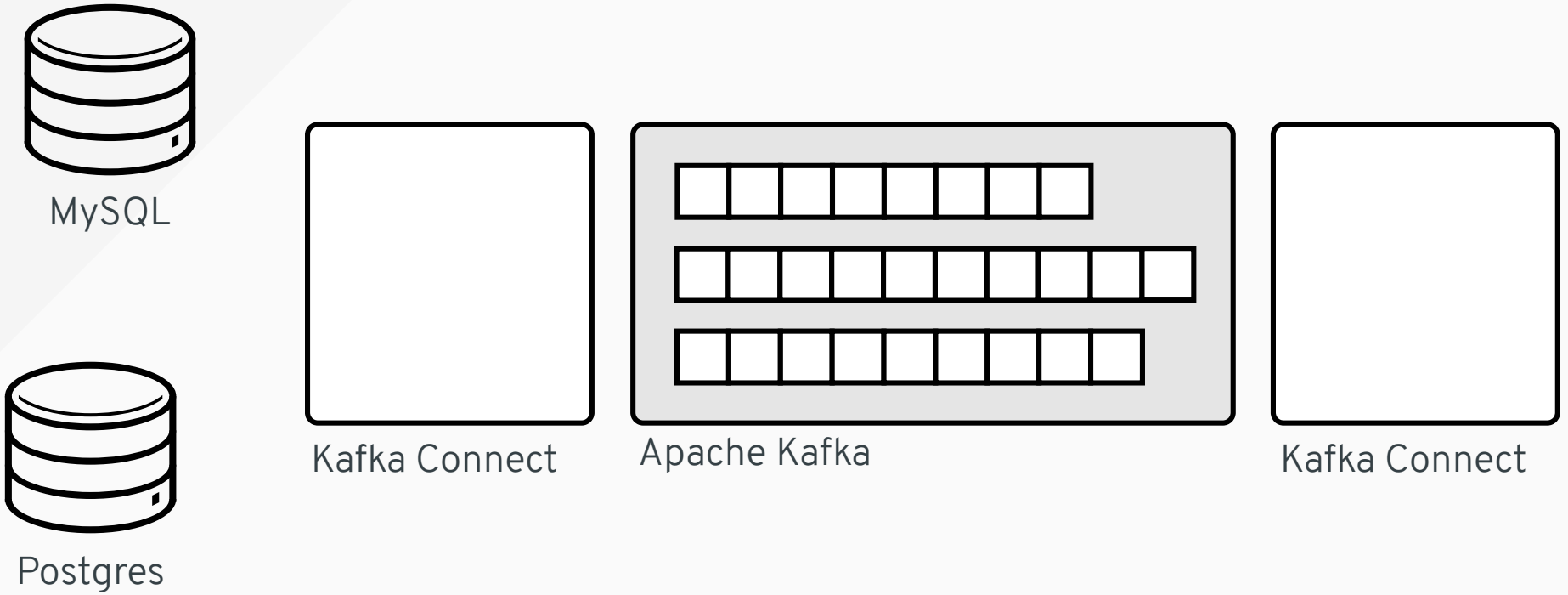
Postgres

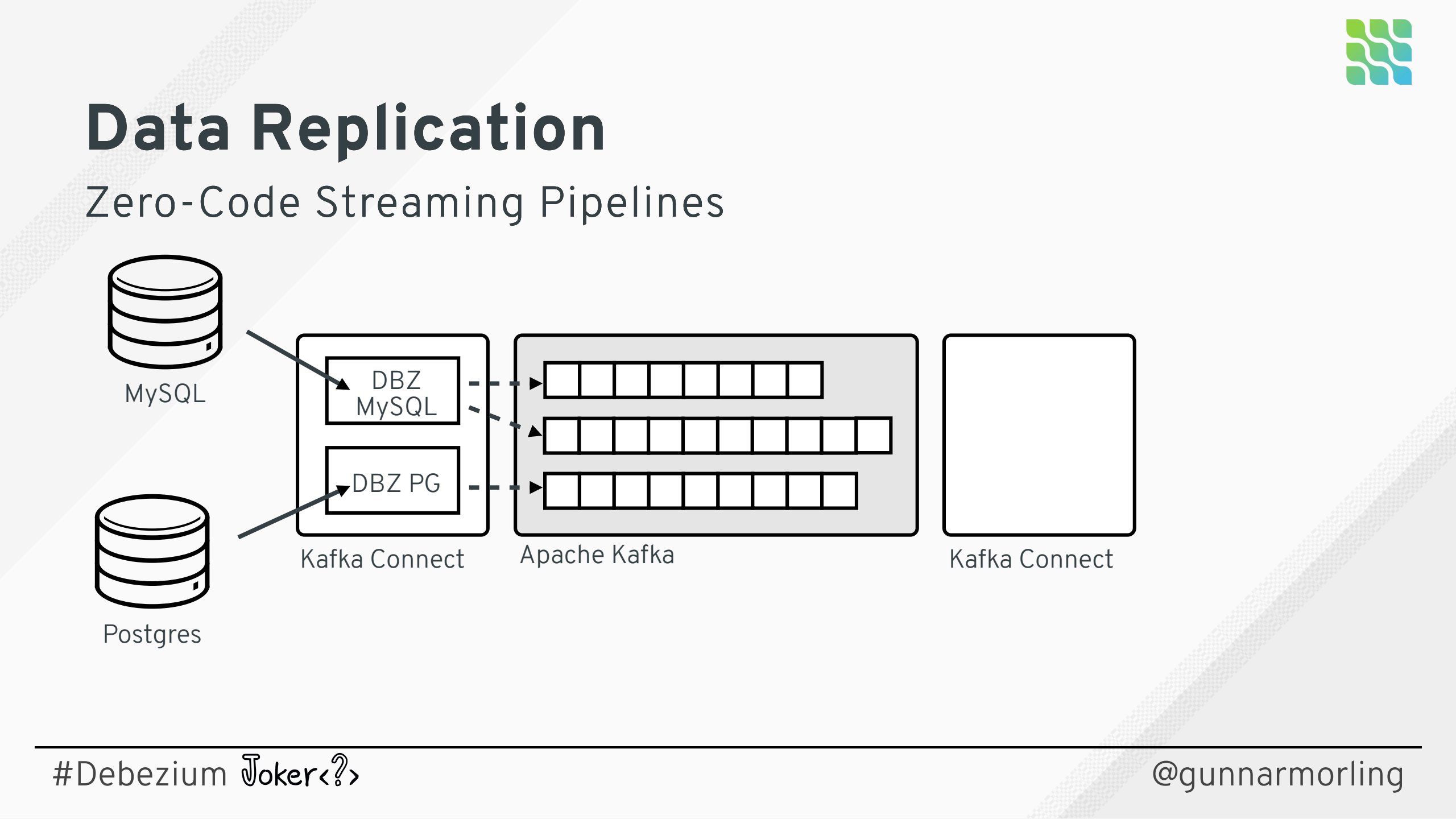


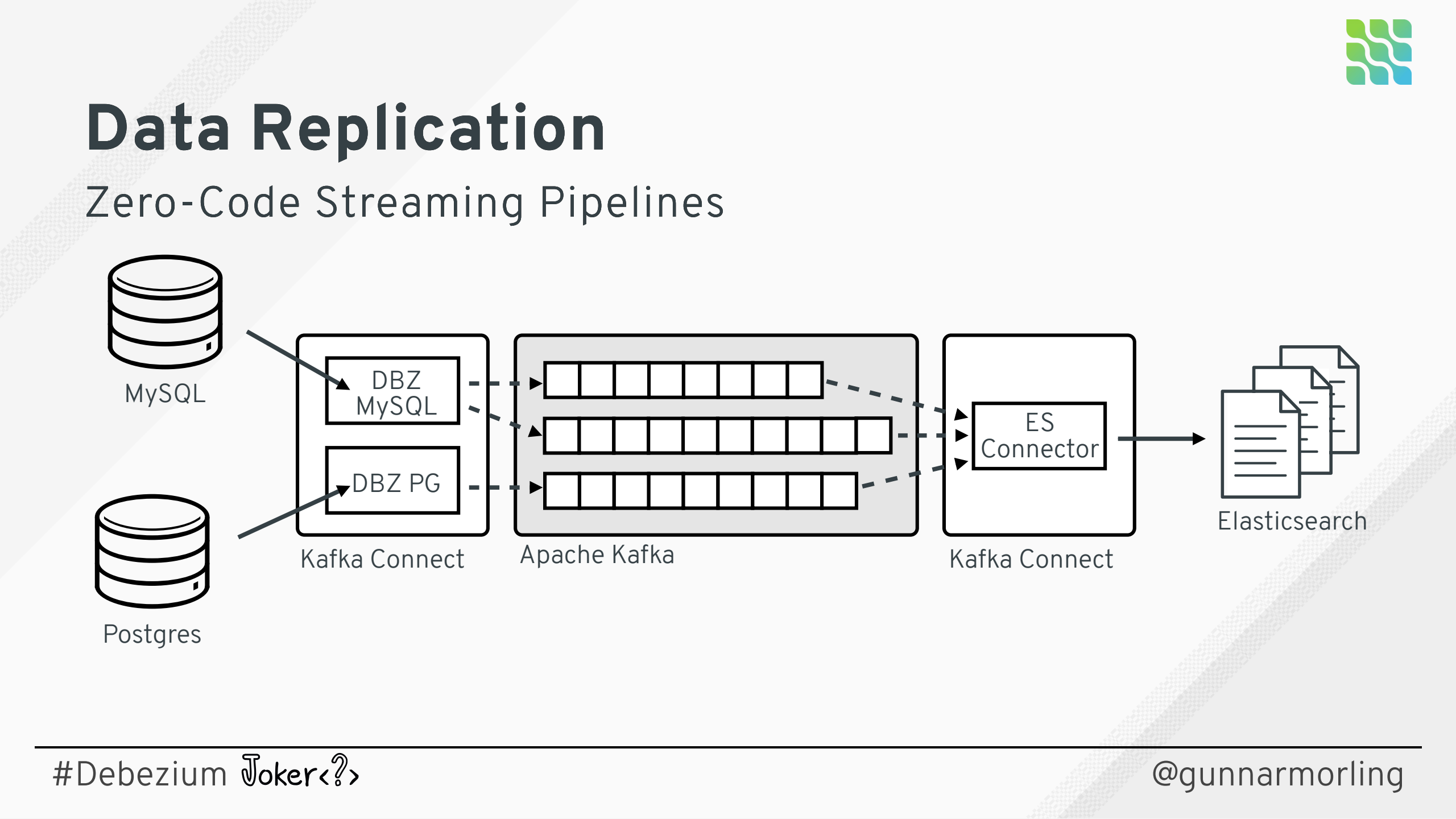
Apache Kafka

Data Replication

Zero-Code Streaming Pipelines

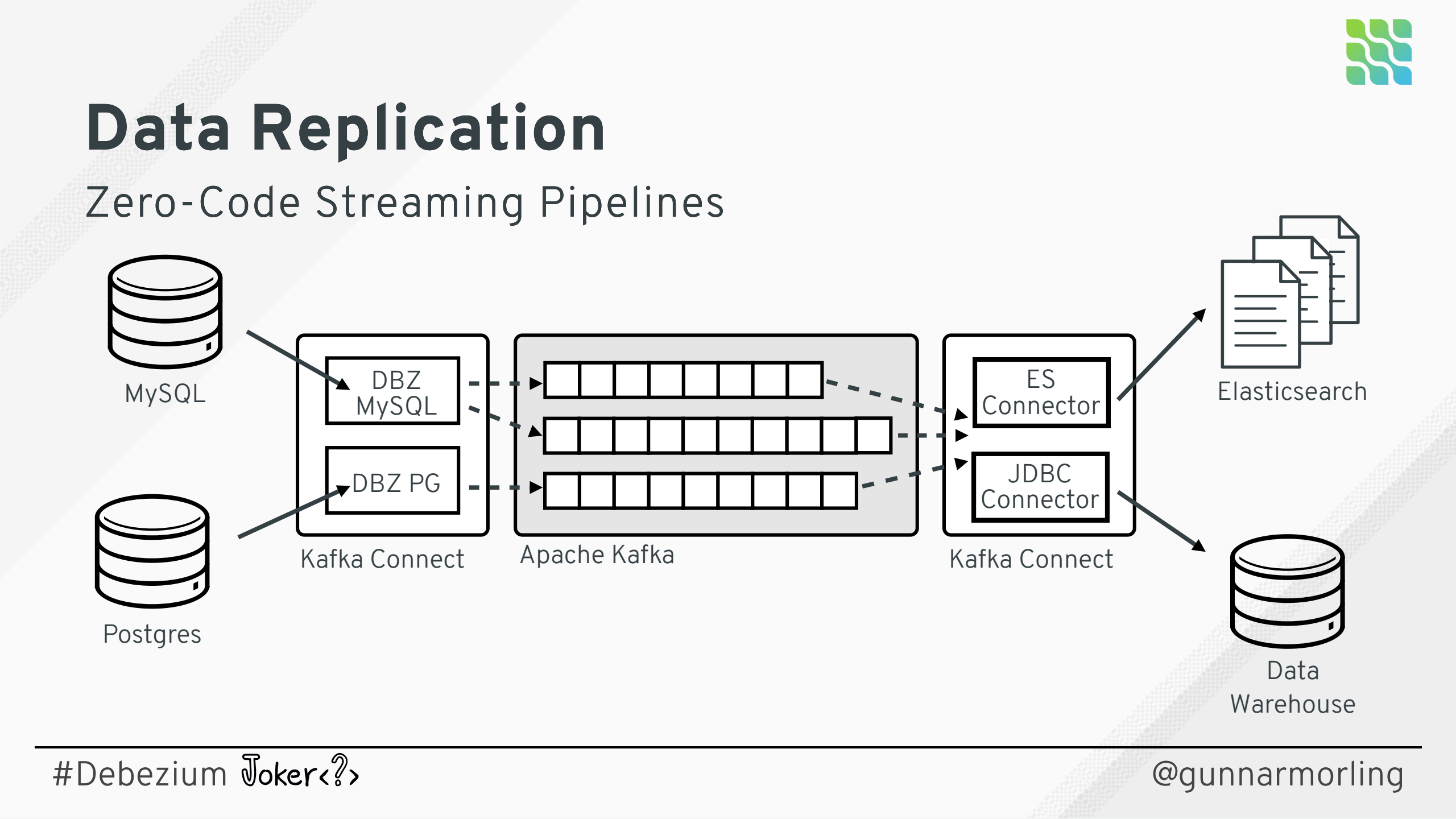



Data Replication
Zero-Code Streaming Pipelines
MySQL
Postgres
DBZ MySQL
DBZ PG
Kafka Connect
Apache Kafka
Kafka Connect
#Debezium Joker<?>
@gunnarmorling



Data Replication

Zero-Code Streaming Pipelines

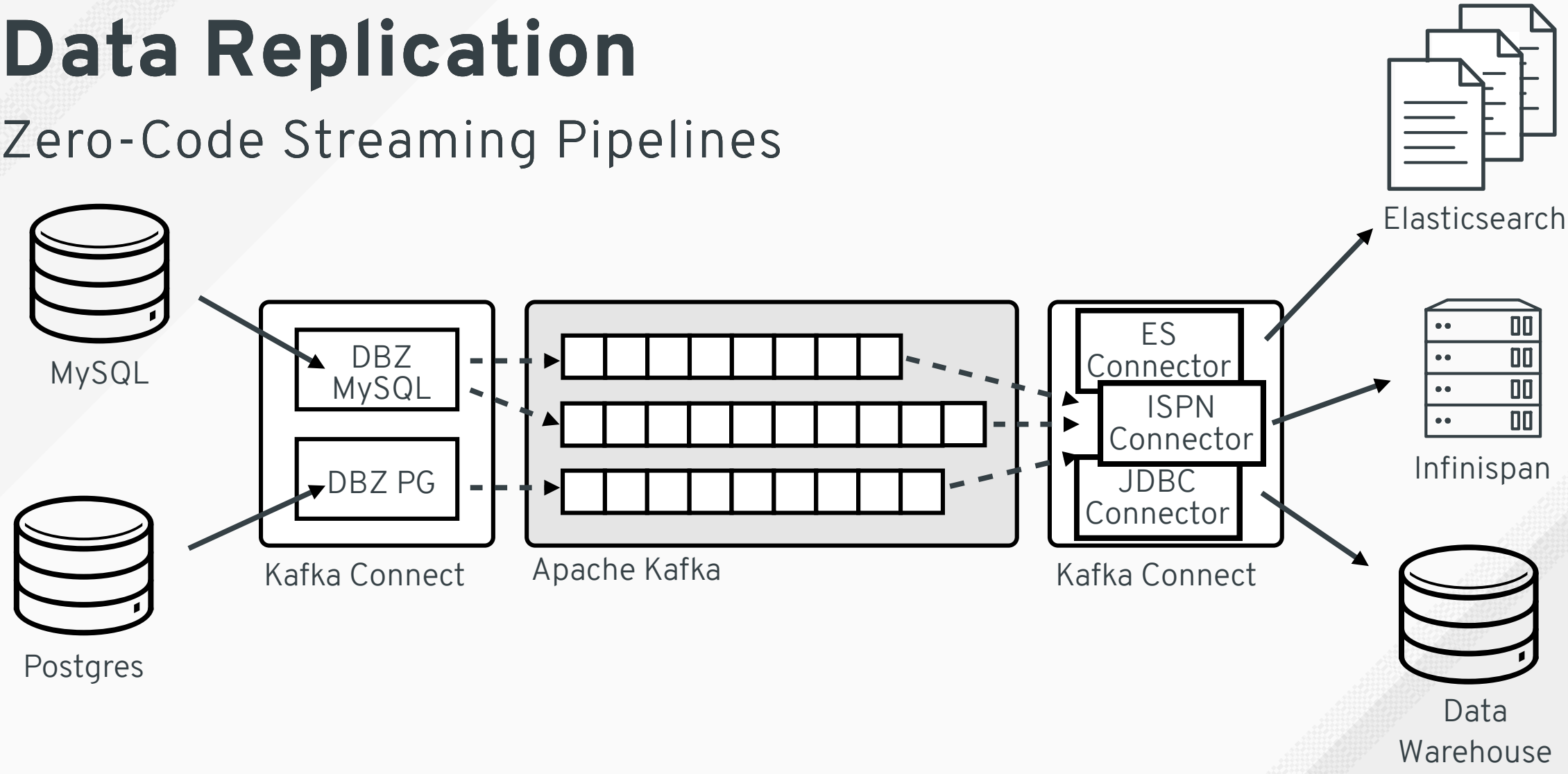


Data Replication

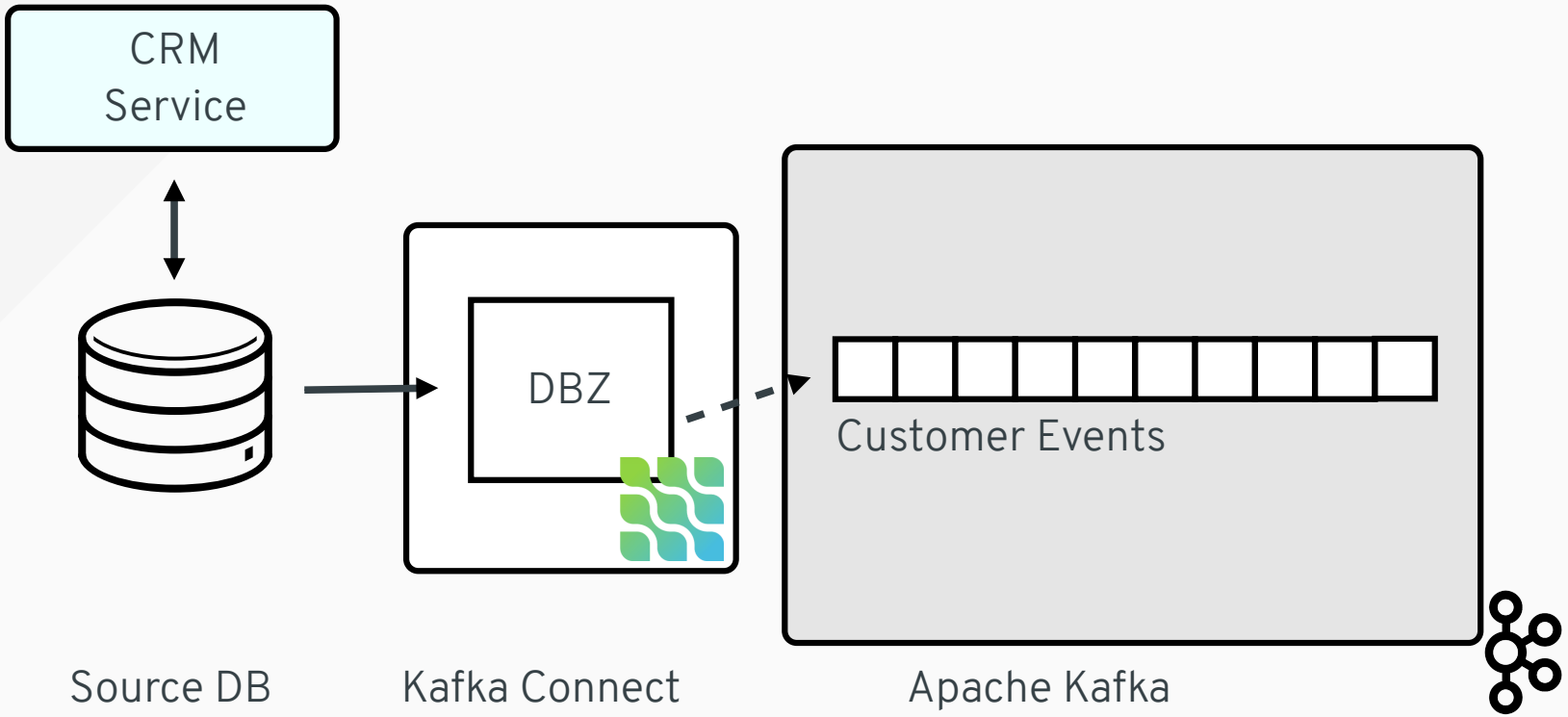
Zero-Code Streaming Pipelines

Data Replication

Zero-Code Streaming Pipelines



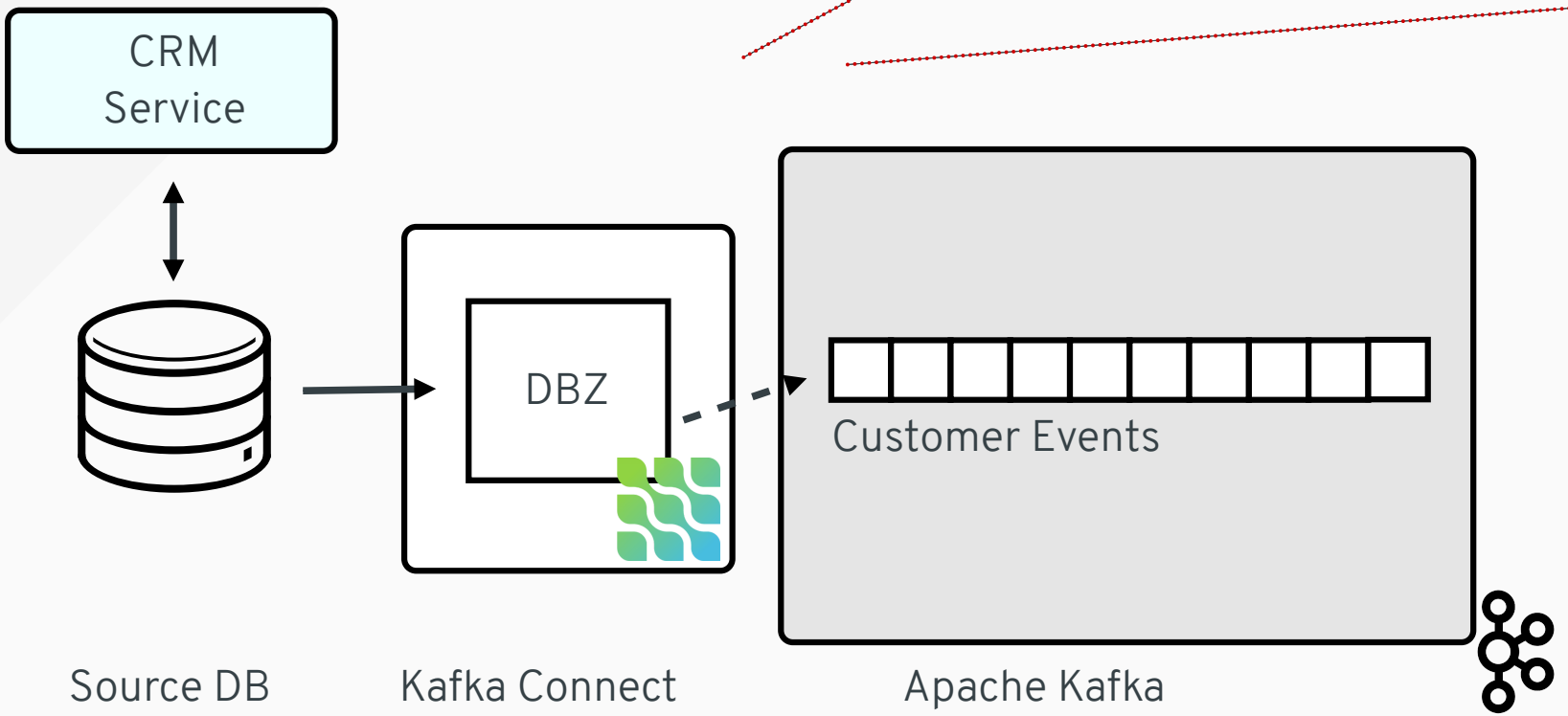
Auditing


CRM
Service
DBZ
Customer Events
Source DB
Kafka Connect
Apache Kafka

Auditing

"Transactions" table

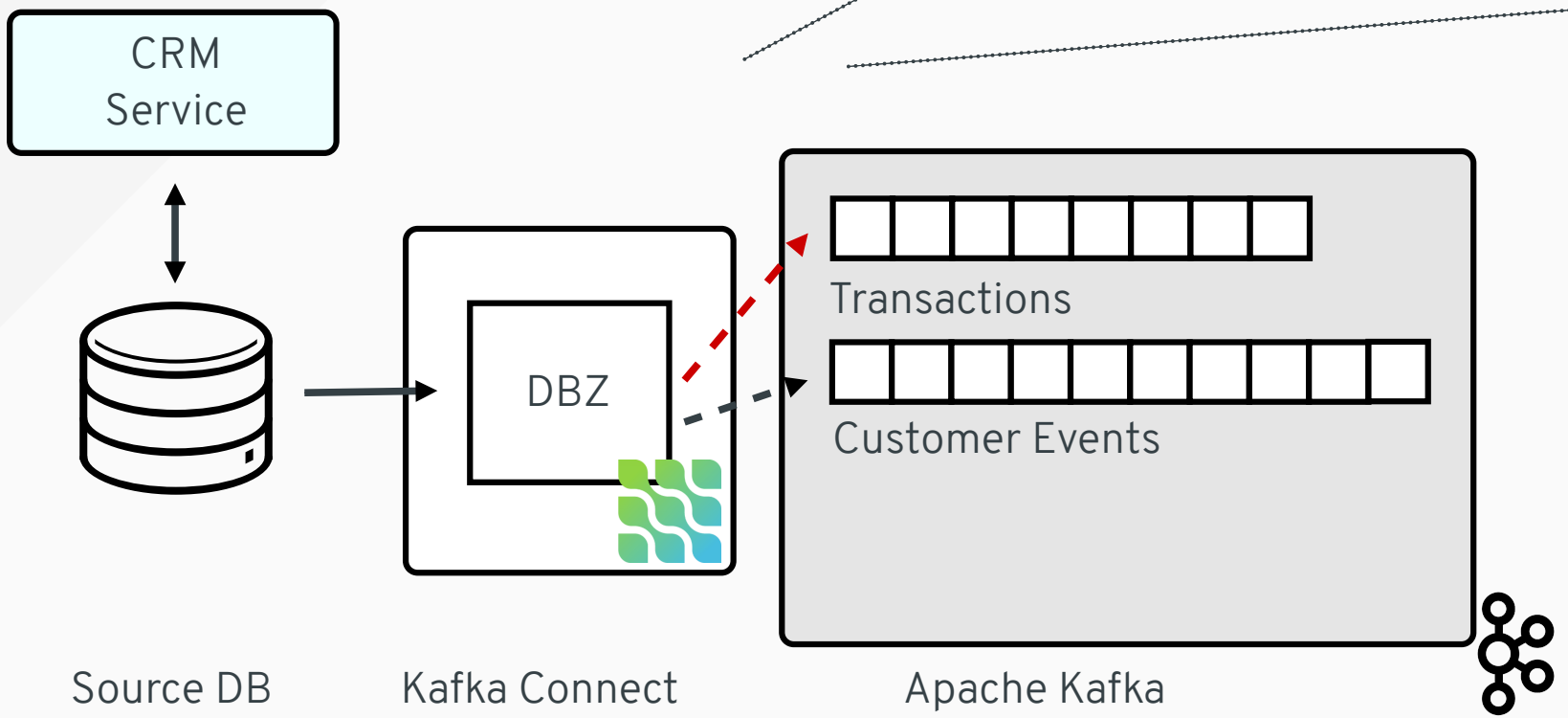
Id	User	Use Case
tx-1	Bob	Create Customer
tx-2	Sarah	Delete Customer
tx-3	Rebecca	Update Customer



Auditing

"Transactions" table

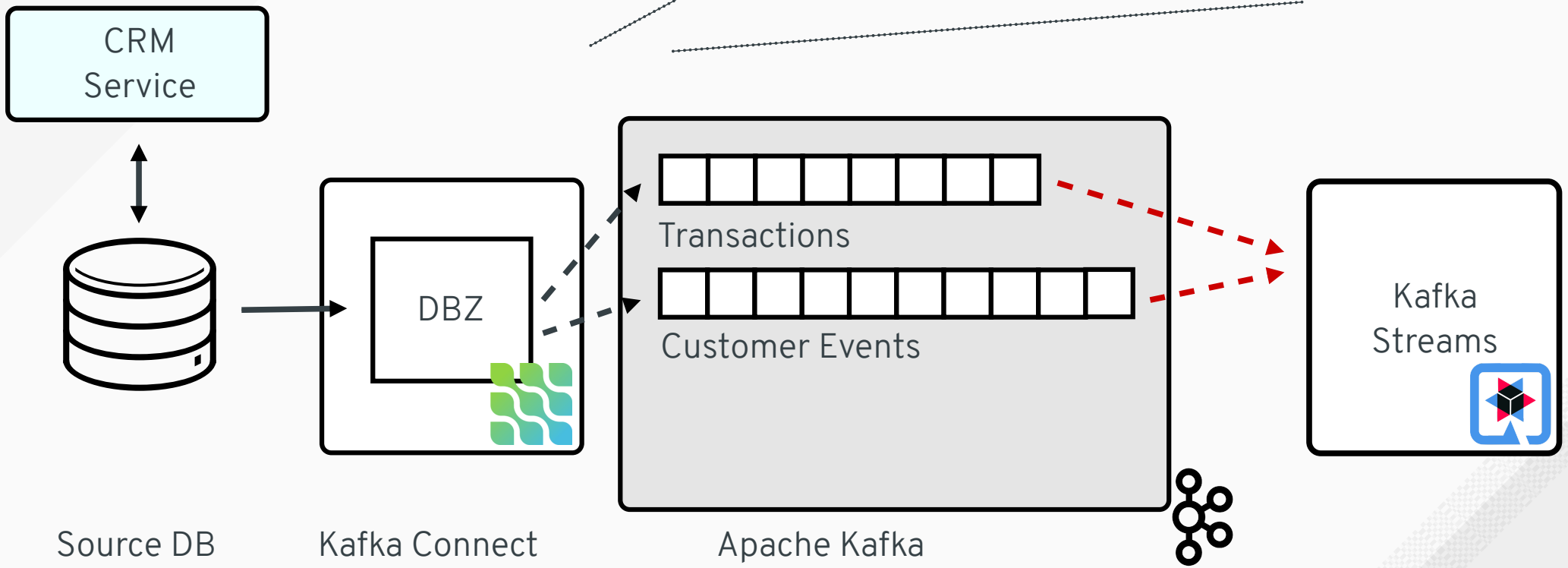
Id	User	Use Case
tx-1	Bob	Create Customer
tx-2	Sarah	Delete Customer
tx-3	Rebecca	Update Customer



Auditing

"Transactions" table

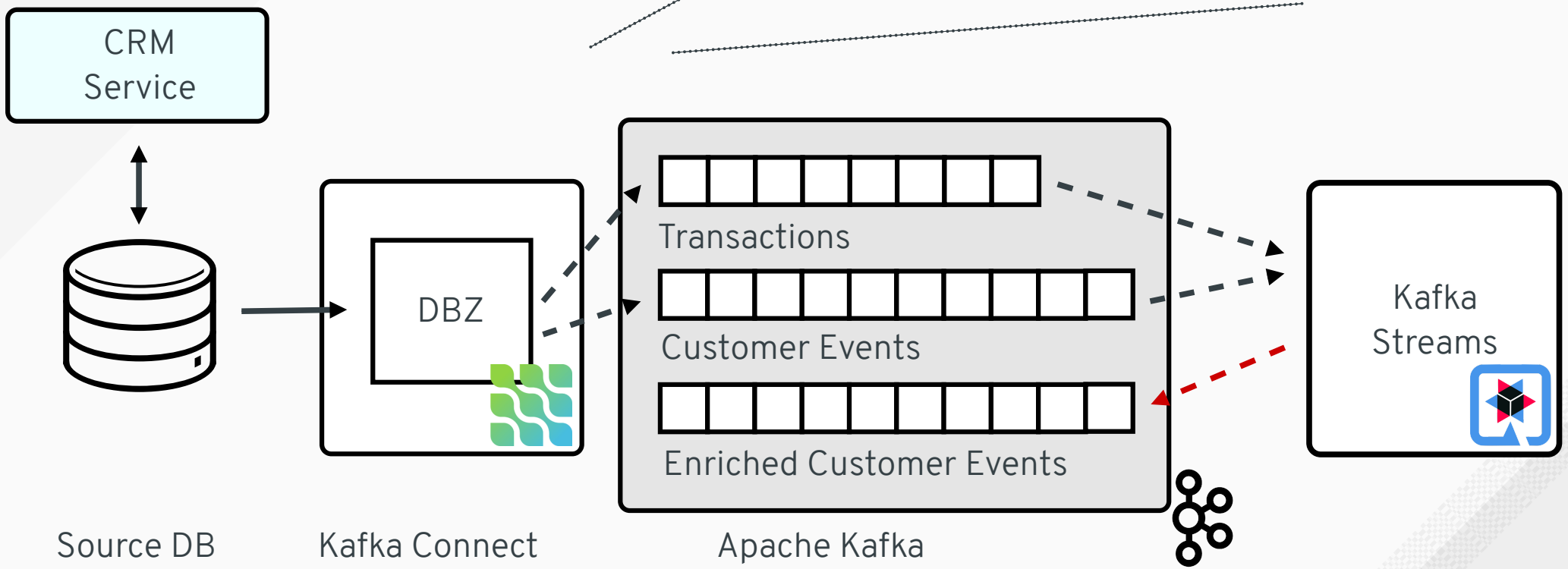
Id	User	Use Case
tx-1	Bob	Create Customer
tx-2	Sarah	Delete Customer
tx-3	Rebecca	Update Customer



Auditing

"Transactions" table

Id	User	Use Case
tx-1	Bob	Create Customer
tx-2	Sarah	Delete Customer
tx-3	Rebecca	Update Customer



Auditing

```
{
  "before": {
    "id": 1004,
    "last_name": "Kretchmar",
    "email": "annek@example.com"
  },
  "after": {
    "id": 1004,
    "last_name": "Kretchmar",
    "email": "annek@noanswer.org"
  },
  "source": {
    "name": "dbserver1",
    "table": "customers",
    "txId": "tx-3"
  },
  "op": "u",
  "ts_ms": 1486500577691
}
```

Customers

```
{
  "id": "tx-3"
}
```

```
{
  "before": null,
  "after": {
    "id": "tx-3",
    "user": "Rebecca",
    "use_case": "Update customer"
  },
  "source": {
    "name": "dbserver1",
    "table": "transactions",
    "txId": "tx-3"
  },
  "op": "c",
  "ts_ms": 1486500577691
}
```

Transactions

```
{
  "before": {
    "id": 1004,
    "last_name": "Kretchmar",
    "email": "annek@example.com"
  },
  "after": {
    "id": 1004,
    "last_name": "Kretchmar",
    "email": "annek@noanswer.org"
  },
  "source": {
    "name": "dbserver1",
    "table": "customers",
    "txId": "tx-3"
  },
  "op": "u",
  "ts_ms": 1486500577691
}
```

Customers

```
{
  "id": "tx-3"
}
```

```
{
  "before": null,
  "after": {
    "id": "tx-3",
    "user": "Rebecca",
    "use_case": "Update customer"
  },
  "source": {
    "name": "dbserver1",
    "table": "transactions",
    "txId": "tx-3"
  },
  "op": "c",
  "ts_ms": 1486500577691
}
```

Transactions

```
{
  "before": {
    "id": 1004,
    "last_name": "Kretchmar",
    "email": "annek@example.com"
  },
  "after": {
    "id": 1004,
    "last_name": "Kretchmar",
    "email": "annek@noanswer.org"
  },
  "source": {
    "name": "dbserver1",
    "table": "customers",
    "txId": "tx-3"
  },
  "op": "u",
  "ts_ms": 1486500577691
}
```

Customers

Auditing

bit.ly/debezium-auditlogs

Enriched Customers

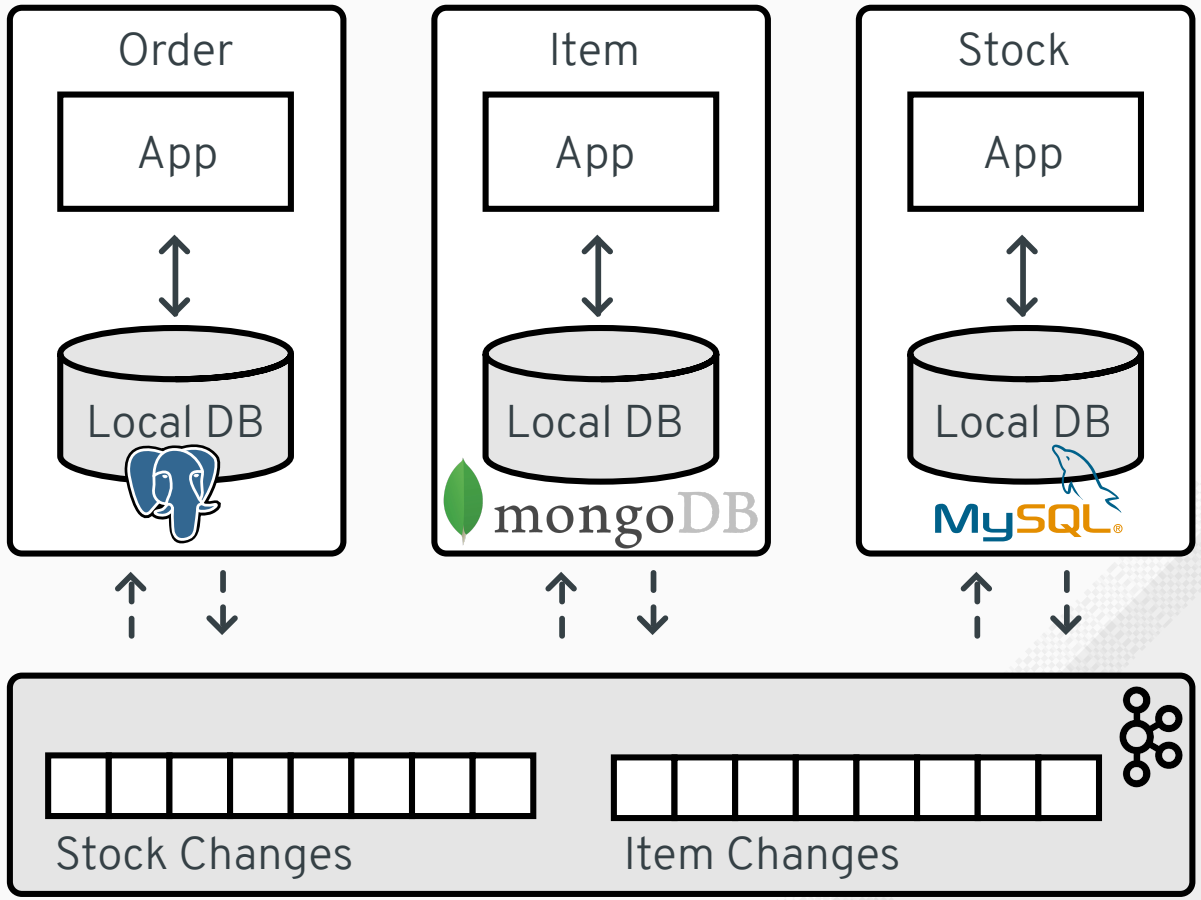
Demo:
Streaming Queries


Microservice
CDC Patterns

Microservice Architectures

Data Synchronization

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

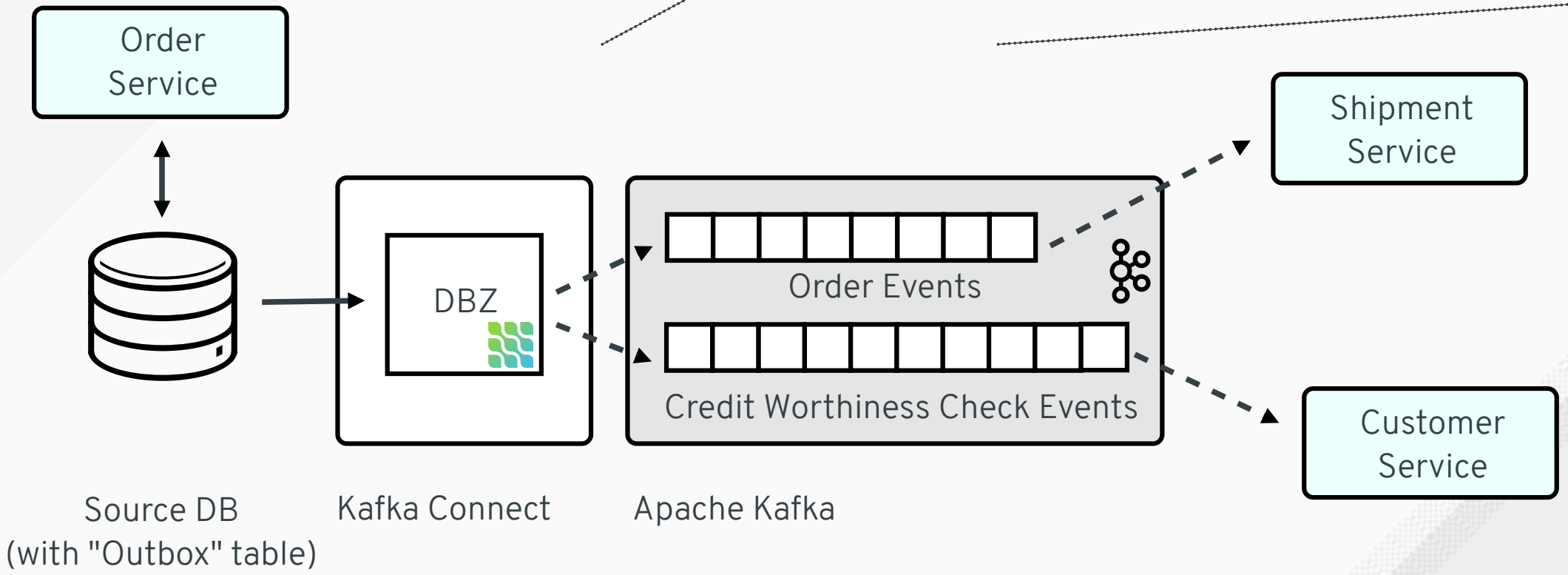


Outbox Pattern

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, ... }



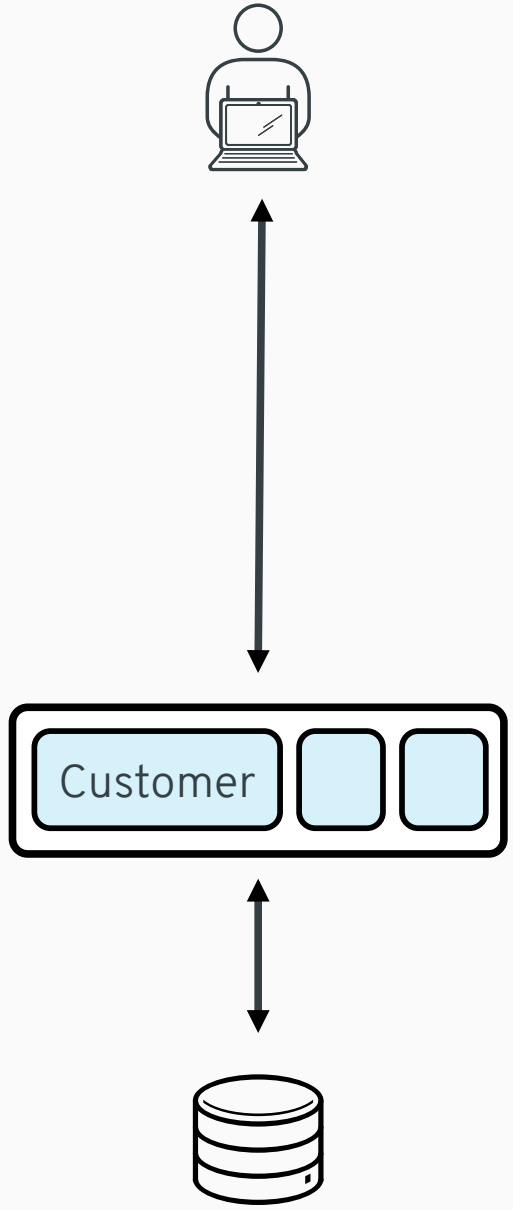
bit.ly/debezium-outbox-pattern

Strangler Pattern

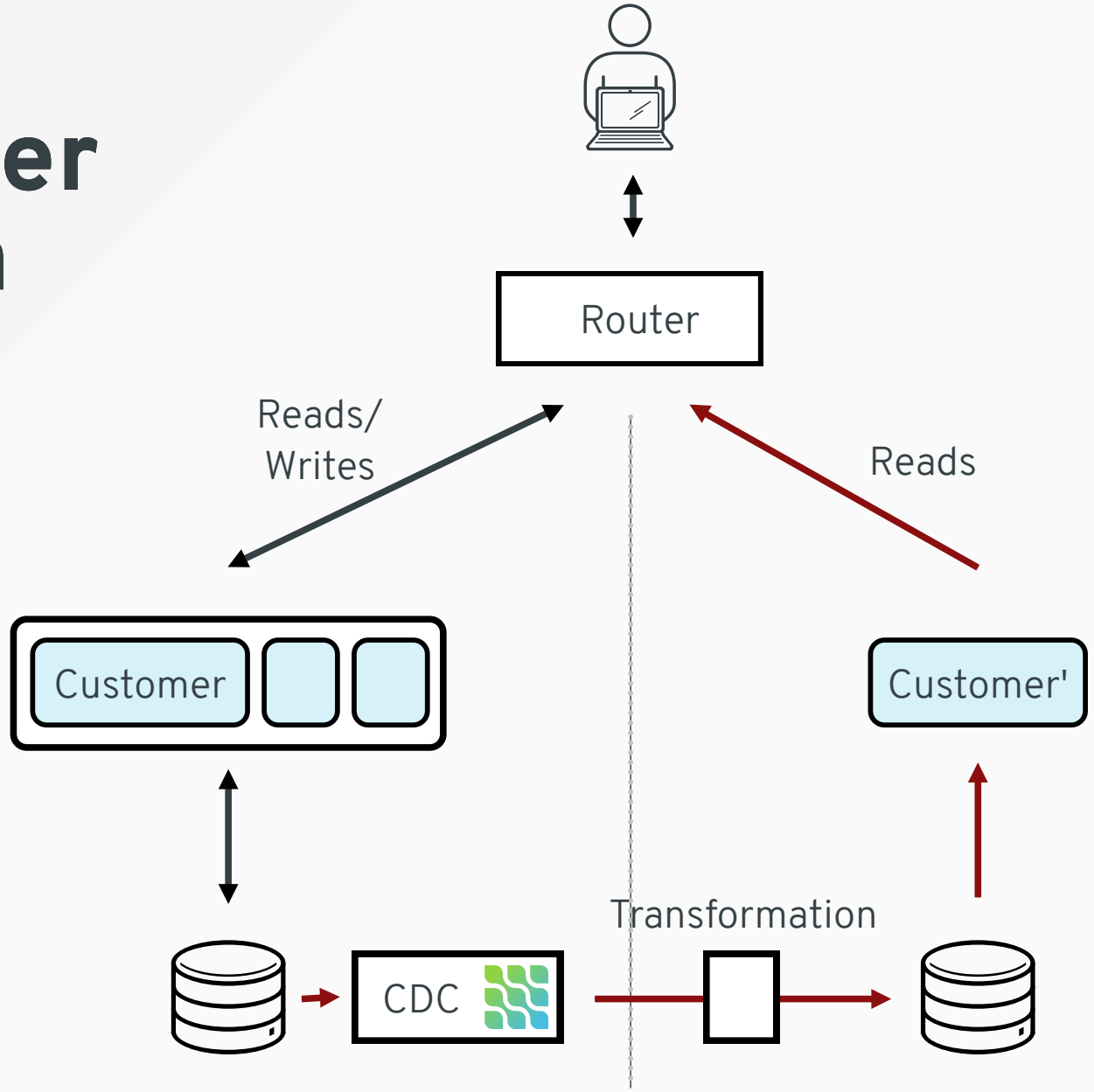
Migrating from Monoliths to Microservices

https://martinfowler.com/bliki/StranglerFigApplication.html

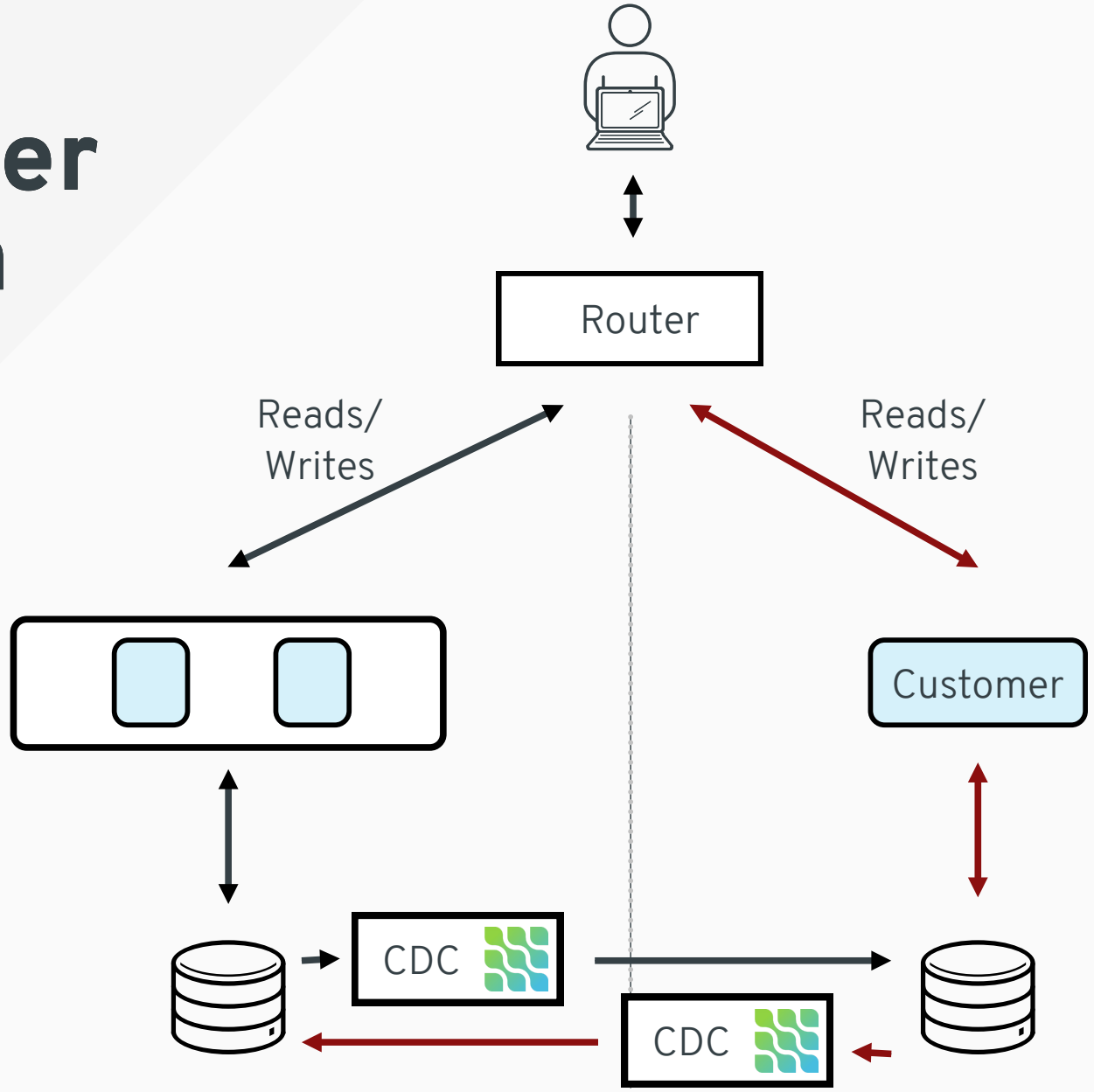
Strangler Pattern



Strangler Pattern



Strangler Pattern



The Issue with Dual Writes

What's the problem?
Change data capture to the rescue!

1

CDC Use Cases & Patterns

Replication
Audit Logs
Microservices

2

Practical Matters

Deployment Topologies
Running on Kubernetes
Single Message Transforms

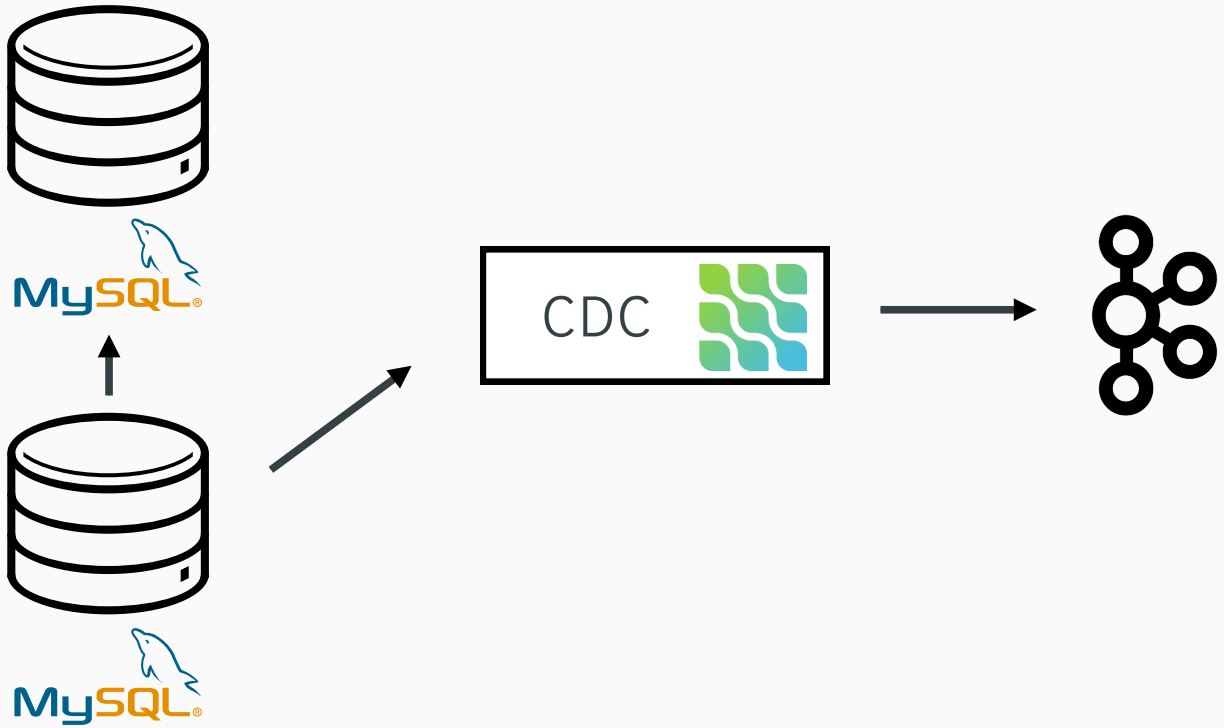
3

Deployment Topologies

Basic Set-Up

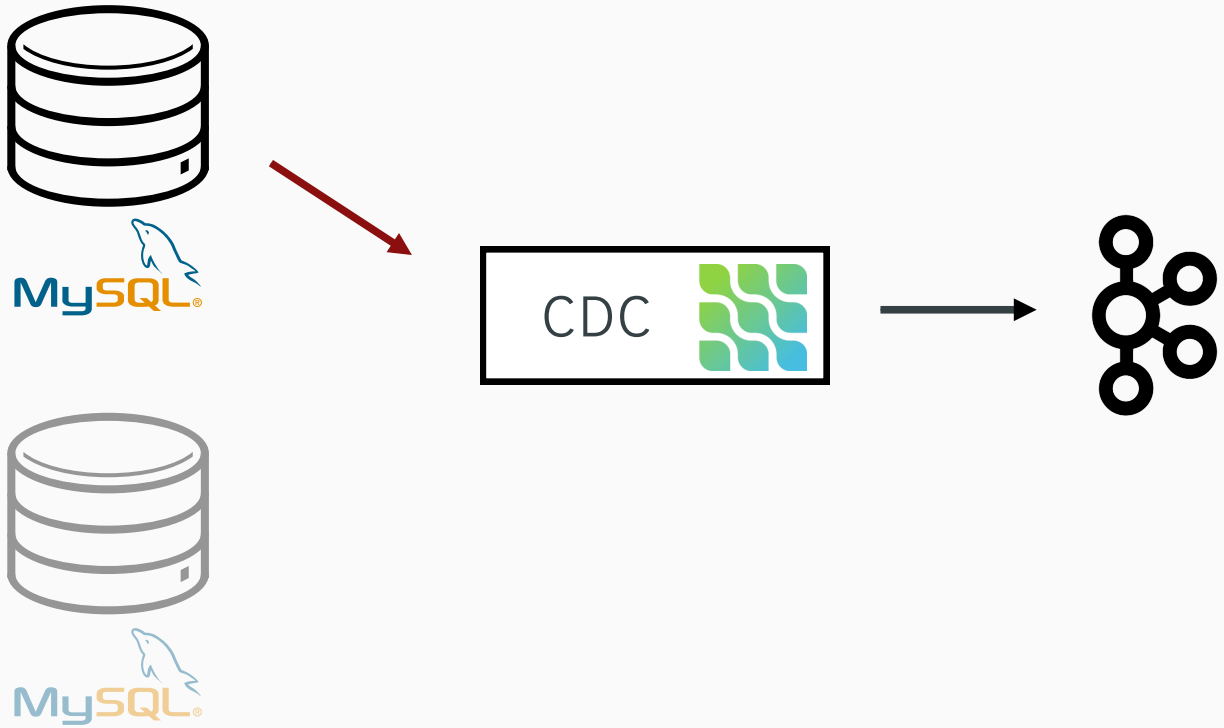
Deployment Topologies

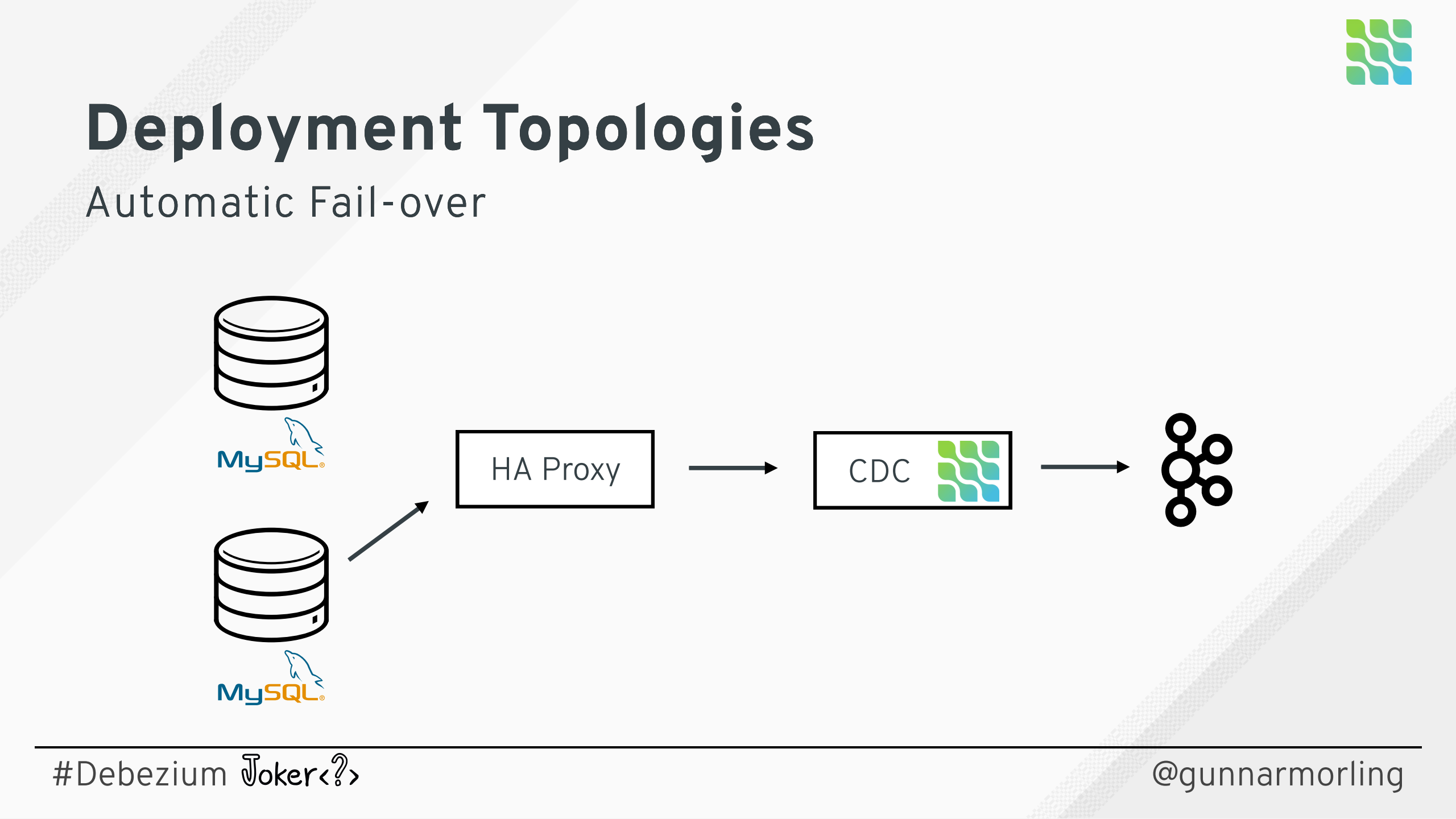
Database High Availability



Deployment Topologies

Database High Availability



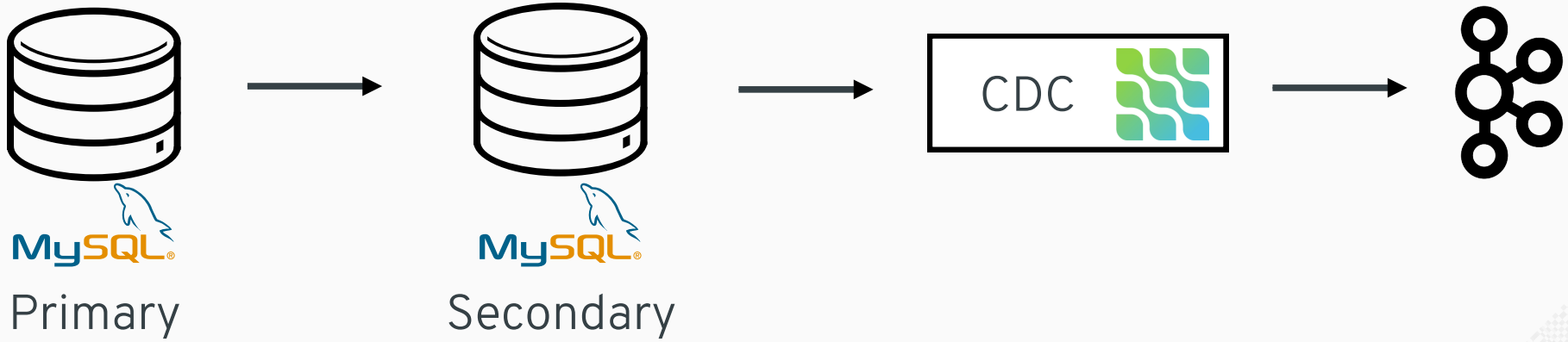


Deployment Topologies

Automatic Fail-over

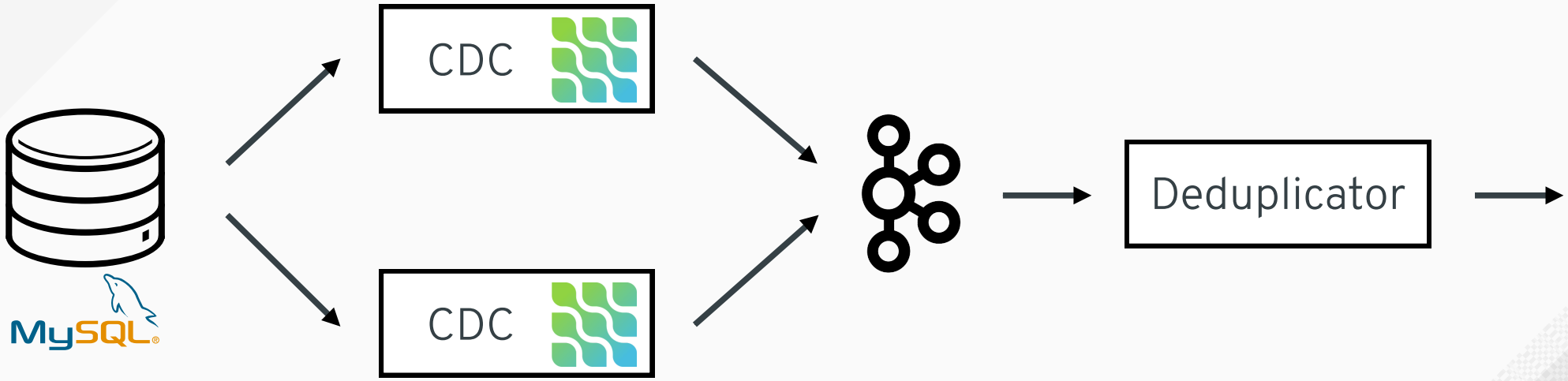
Deployment Topologies

Can't change binlog mode?



Deployment Topologies

High Availability for Connectors



Running on Kubernetes

Deployment via Operators

- YAML-based **custom resource definitions** for Kafka clusters, Kafka Connect, topics and users
- **Operator** applies configuration
- Advantages
  - Automated deployment of Kafka, ZooKeeper etc.
  - Easier Scaling (up and down)
  - Simplified Upgrading
  - Portability across clouds

Running on Kubernetes

Operating Kafka Connect

- **Distributed mode**
  - Offsets stored in Kafka
  - Configuration via REST
- **Single node**: no re-balancing issues (< Apache Kafka 2.3)
- **Single connector**: health checks based on REST API

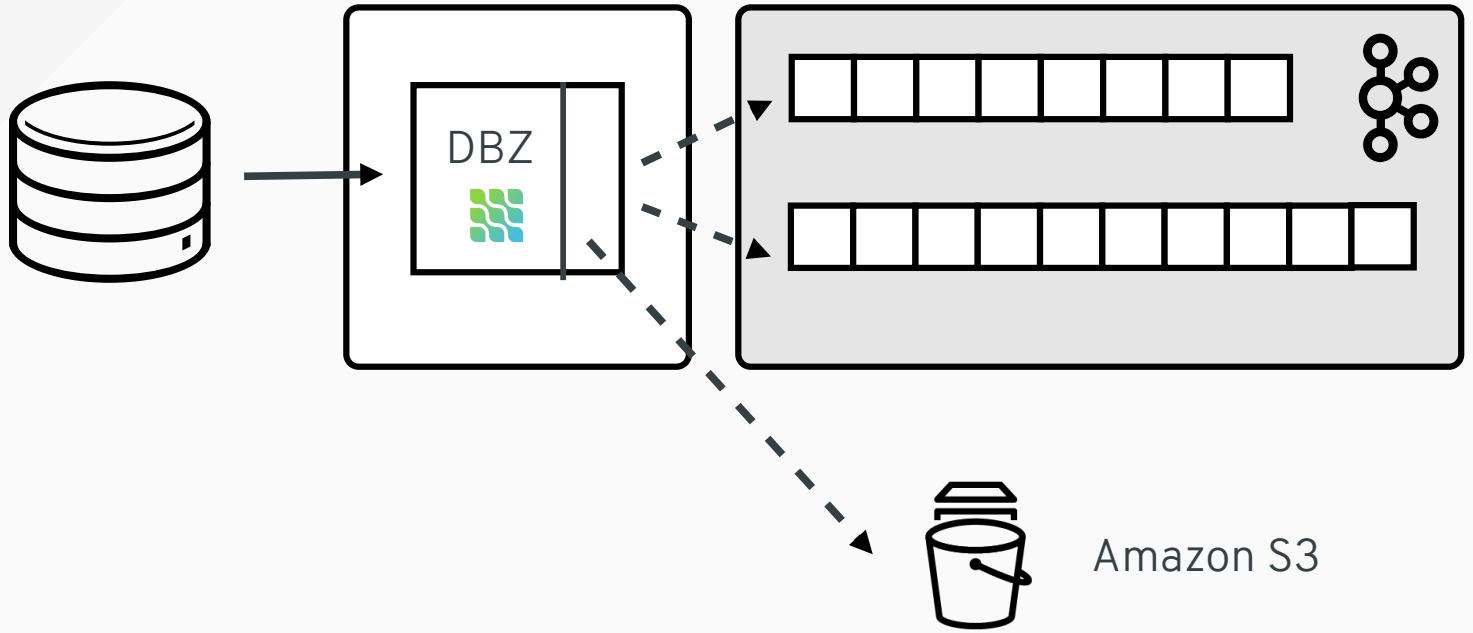
Single Message Transformations

The Swiss Army Knife of Kafka Connect

- **Format conversions**
  - Time/date fields
  - Extract new row state
- **Aggregate** sharded tables to single topic
- **Keep compatibility** with existing consumers

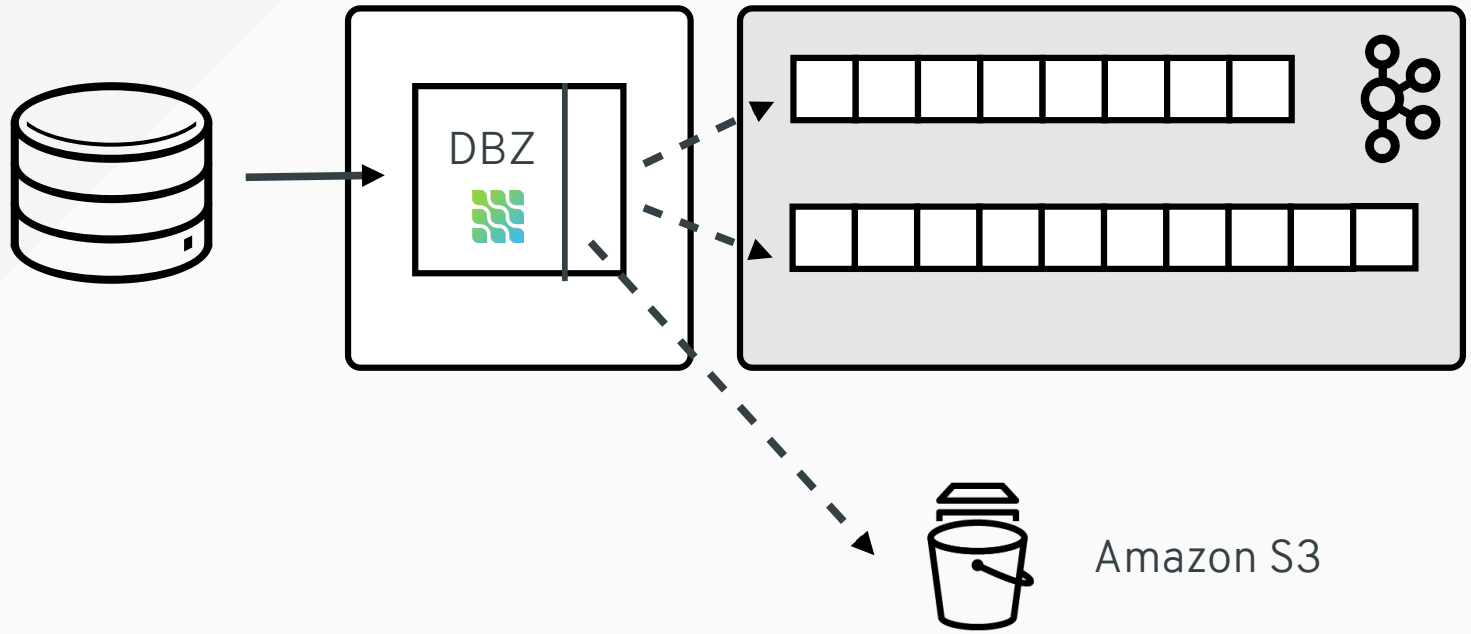
Single Message Transformations

Externalizing large field values



Single Message Transformations

Externalizing large field values

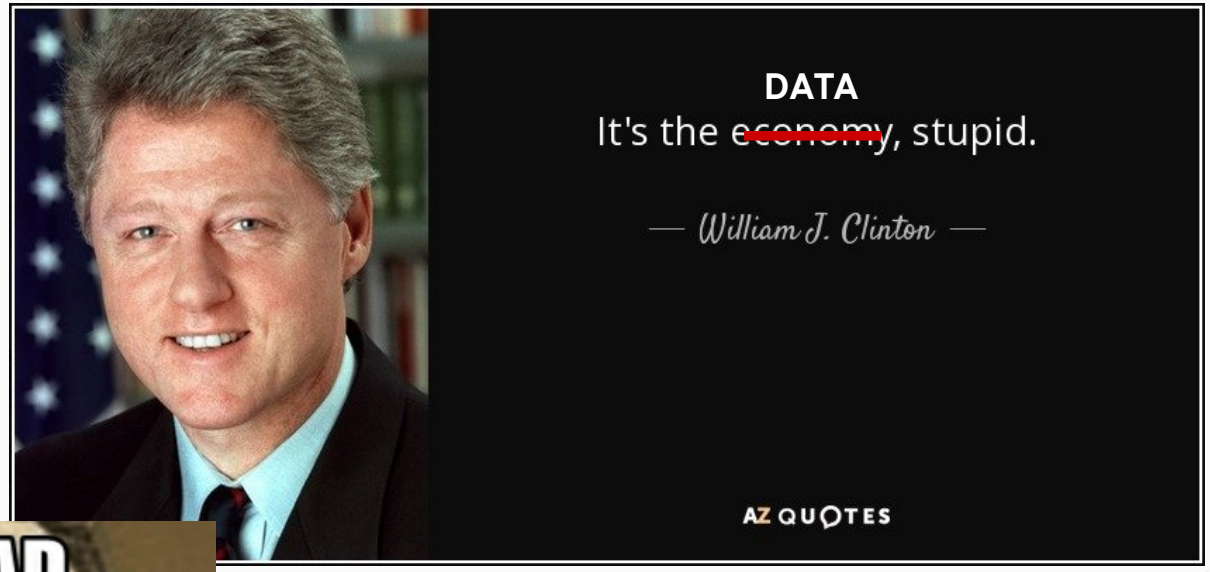



```
{
  "before": { ... },
  "after": {
    "id": 1004,
    "last_name": "Kretchmar",
    "email": "annek@noanswer.org",
    "image":
      "imgs-<offset>-after"
  },
  ...
}
```

Summary

- Change Data Capture – **Liberation for your data**!
- Enabling use cases such as replication, microservices data exchange and much more
- **Debezium:** open-source CDC for a growing number of databases

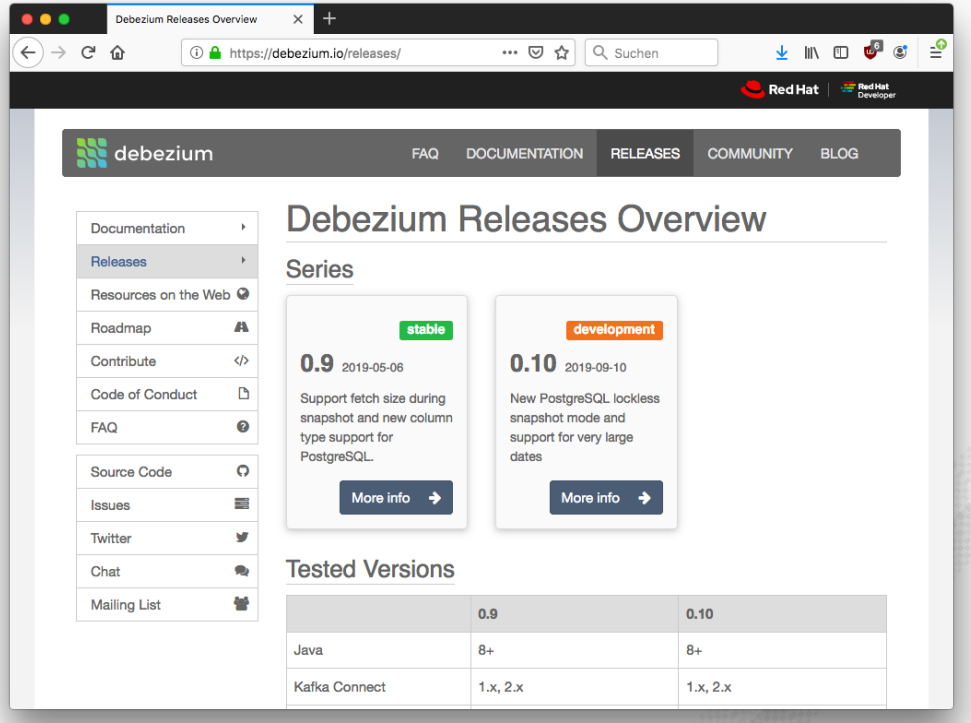

DATA, DATA, DATA!


DATA
It's the economy, stupid.
— William J. Clinton —
AZ QUOTES


OOPS, MY BAD...
WRONG DATA
memegenerator.net

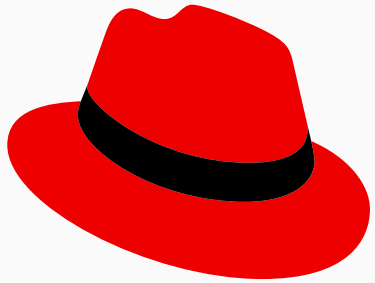
Resources

- **Website:** https://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) https://strimzi.io/
- **Latest news:** @debezium



gunnar@hibernate.org

@gunnarmorling



Red Hat Middleware