Zero downtime on Kubernetes with Hazelcast

Nicolas Fränkel

Me, myself and I

- Developer
- Developer Advocate
- We live in interesting times and I’m curious

Hazelcast

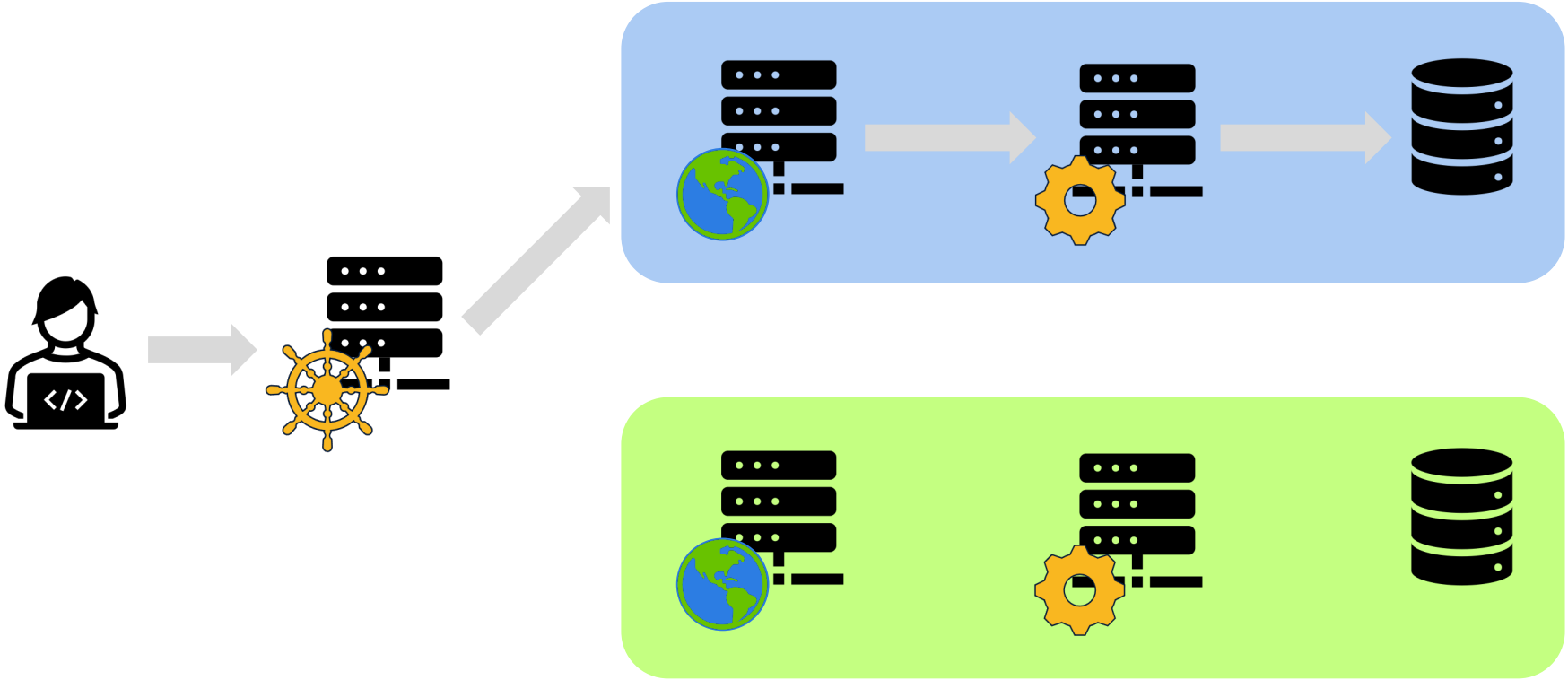
HAZELCAST IMDG is an **operational, in-memory**, distributed computing platform that manages data using in-memory storage and performs parallel execution for breakthrough application speed and scale.

HAZELCAST JET is the ultra fast, application embeddable, 3rd generation stream processing engine for low latency batch and stream processing.

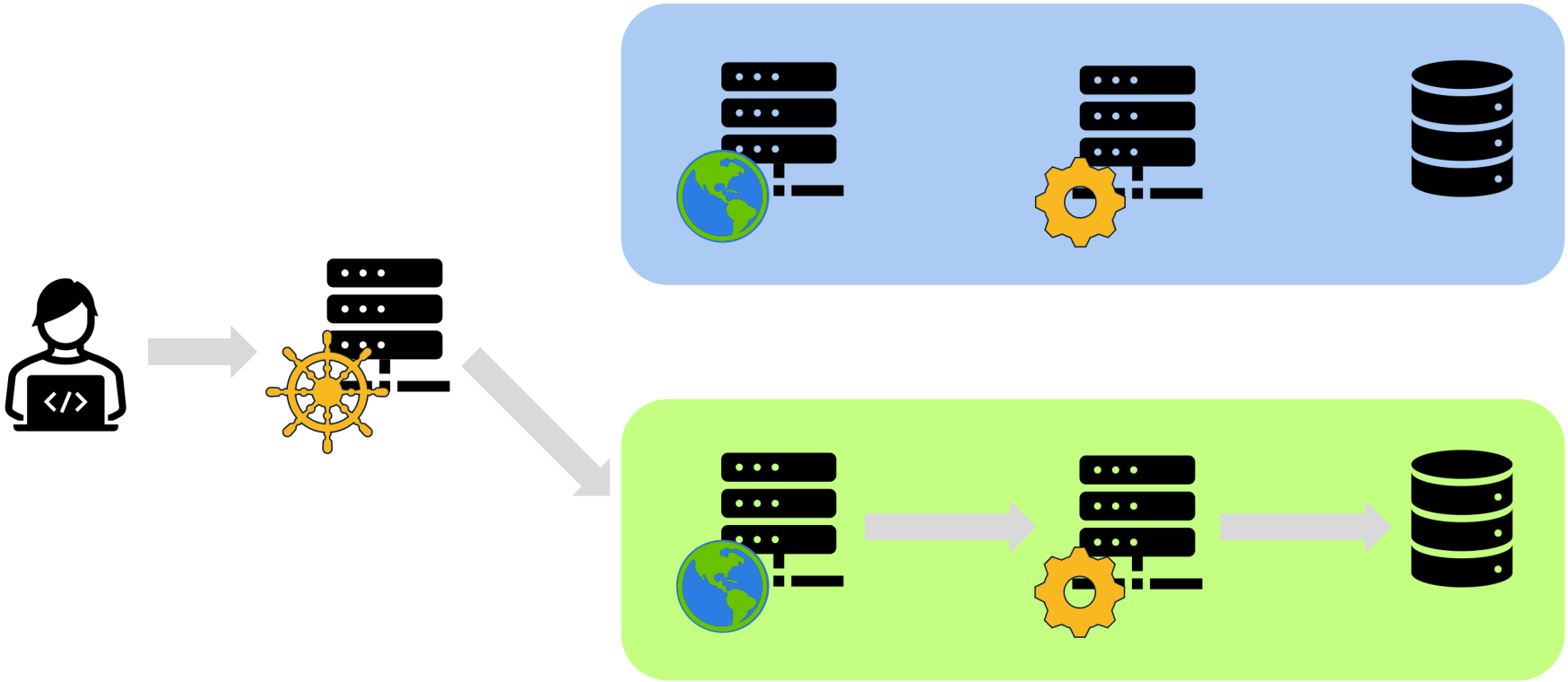
Why zero downtime?

1. Business wants it
   - Downtime has a cost
2. Users expect it
   - When was the last time you saw Google Search display “Please come back later”?

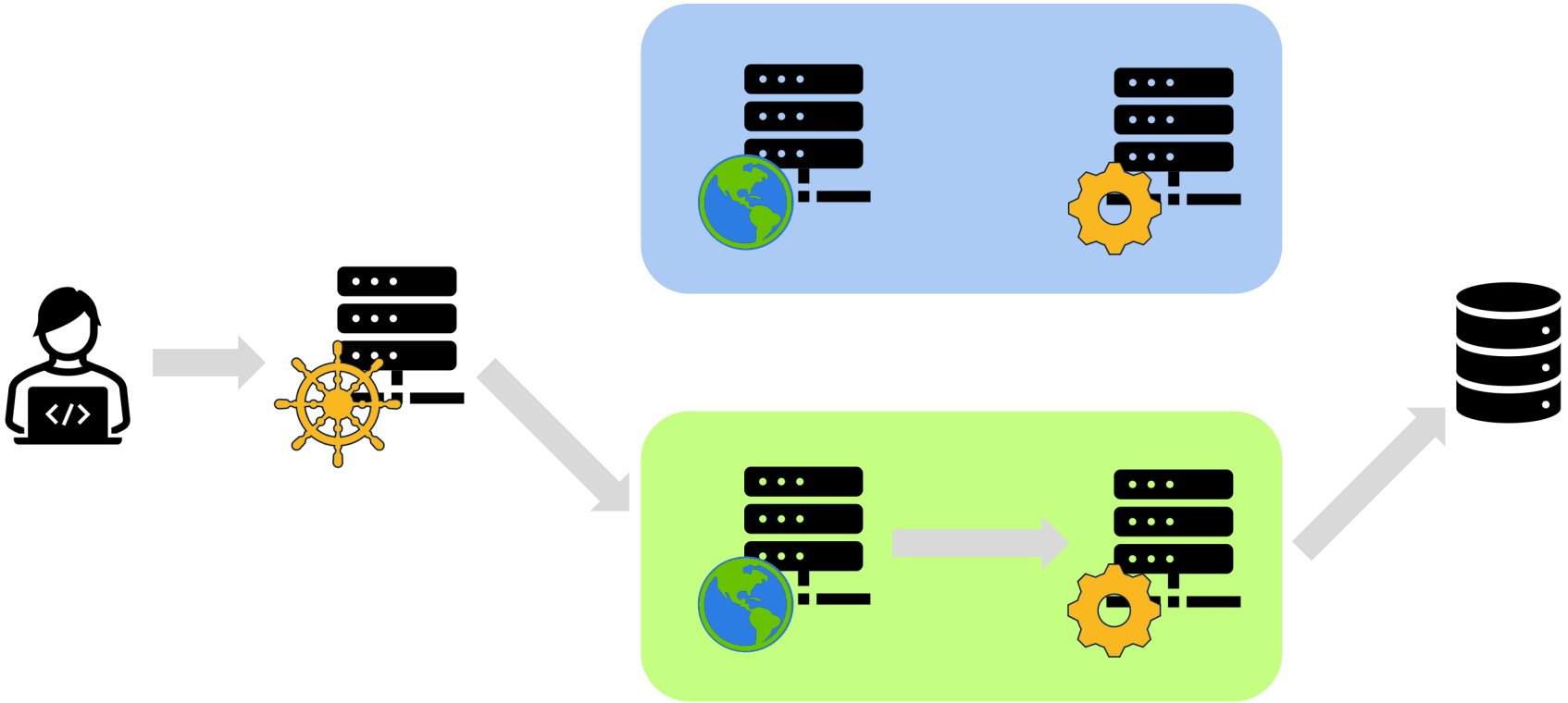
Blue-Green deployment



Blue-Green deployment



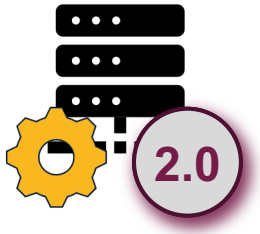
Blue-Green deployment variant

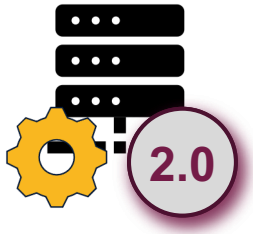


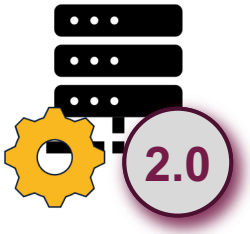
Zero downtime's issues relate to state

- State **in memory**
  - User sessions
- State **in the database**

Kubernetes rolling updates principle







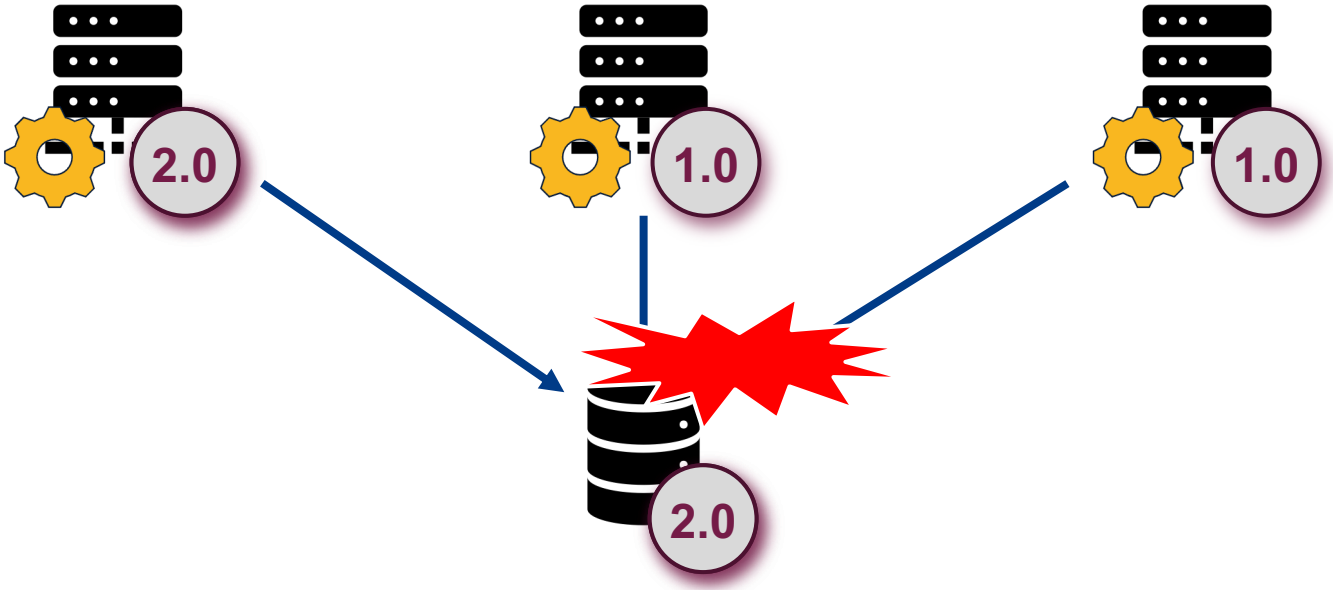
Solving state-related issues

- Sessions
  - Session replication
- Database
  - That’s the hard spot!

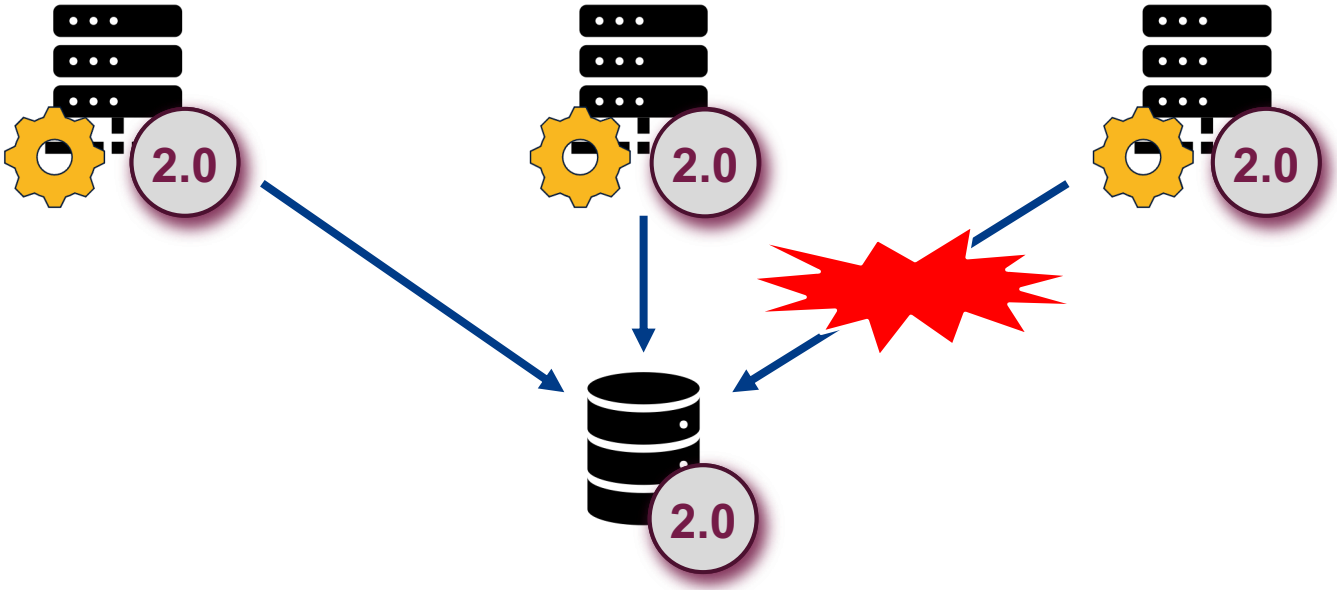
Option 1: Keep the same database

- The application needs to cope with two versions of the schema

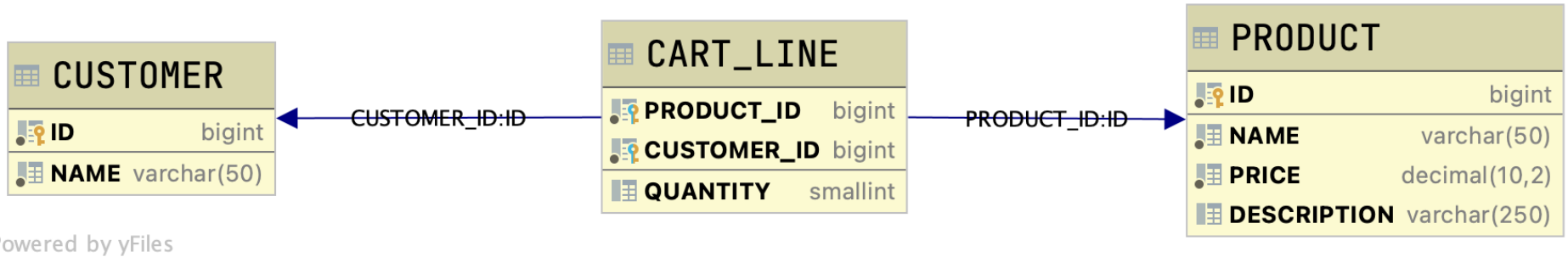
Rolling upgrade issue with a single database



More issues with rollback

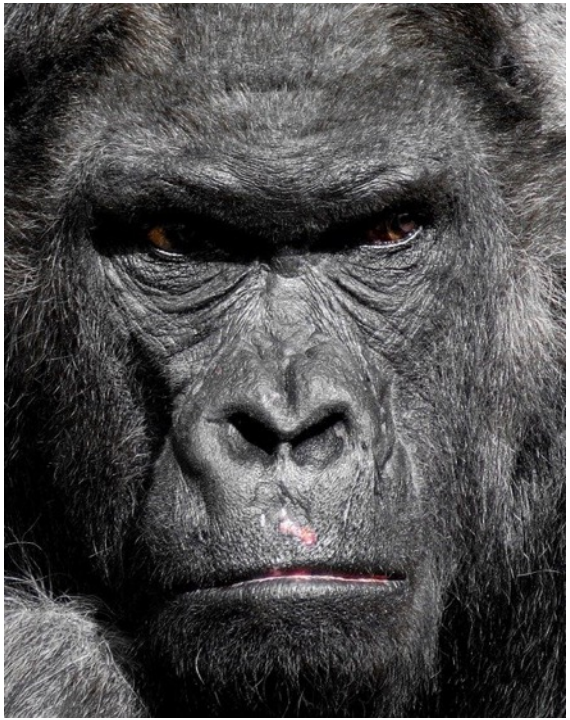


An e-commerce use-case

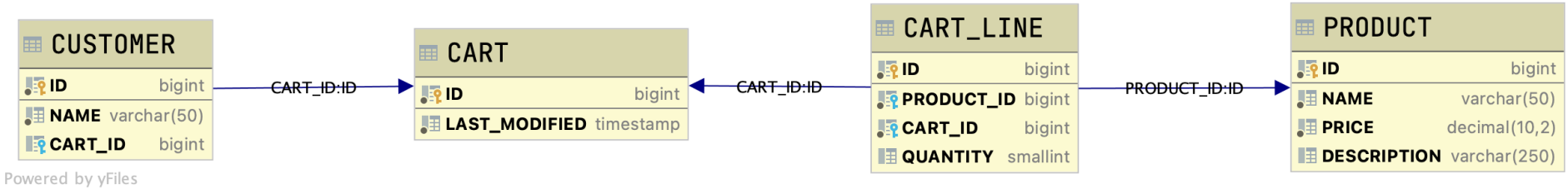


New business requirement comes in!

- We want to keep a track when a cart was created
  - To send a reminder email after some time has passed



Target schema



Handling schema-related breaking changes

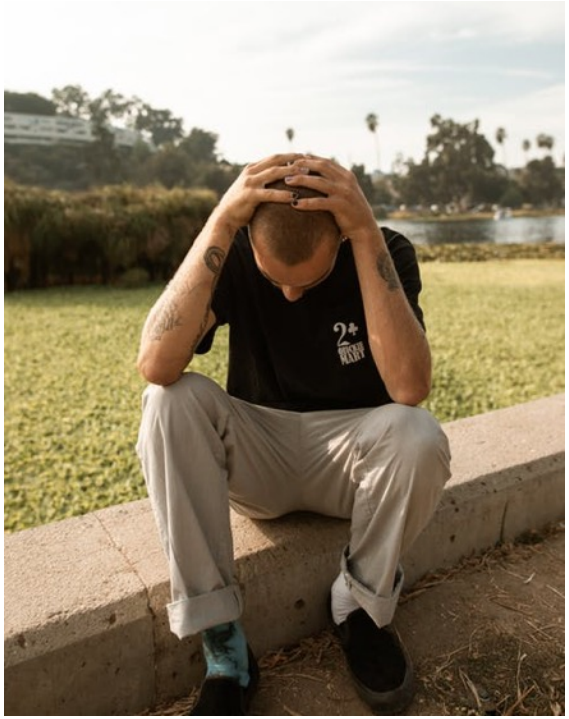
- Split the breaking change into a series of changes compatible side-by-side
- Plan for rollback (it happens!)

Steps' decomposition

1. Create `CART` table
   - App uses "old" data model
   - Trigger inserts `CART` when the first `CART_LINE` is inserted
2. `CART` becomes the "source of truth"
   - App uses the `CART` table
   - Trigger updates `CART_LINE` with `CUSTOMER_ID` every time it's inserted
3. **Migration of untouched data**
4. Cleanup

Issues of keeping the same database

- Requires steps' decomposition
- Rollback a single step only
- Needs planning across the organization (devs, DBAs, Ops)
- You will **need to migrate data anyway**



Option 2: Embrace data migration

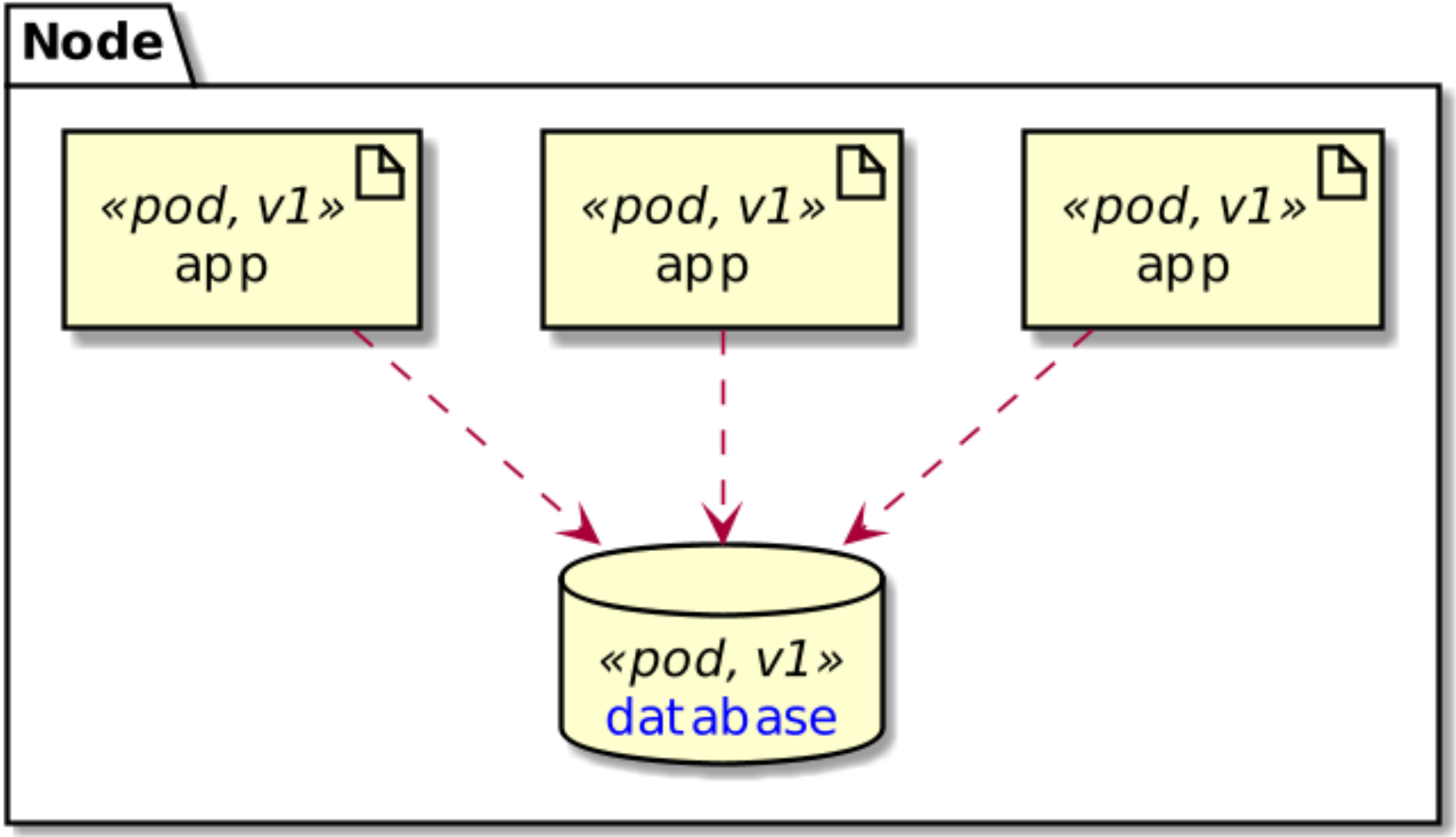
- Have two different databases
- Migration implemented by:
  - Change-Data-Capture
  - **Data streaming**
- Developers are not impacted by Ops' concerns
- It works with any deployment option *e.g.* canary release

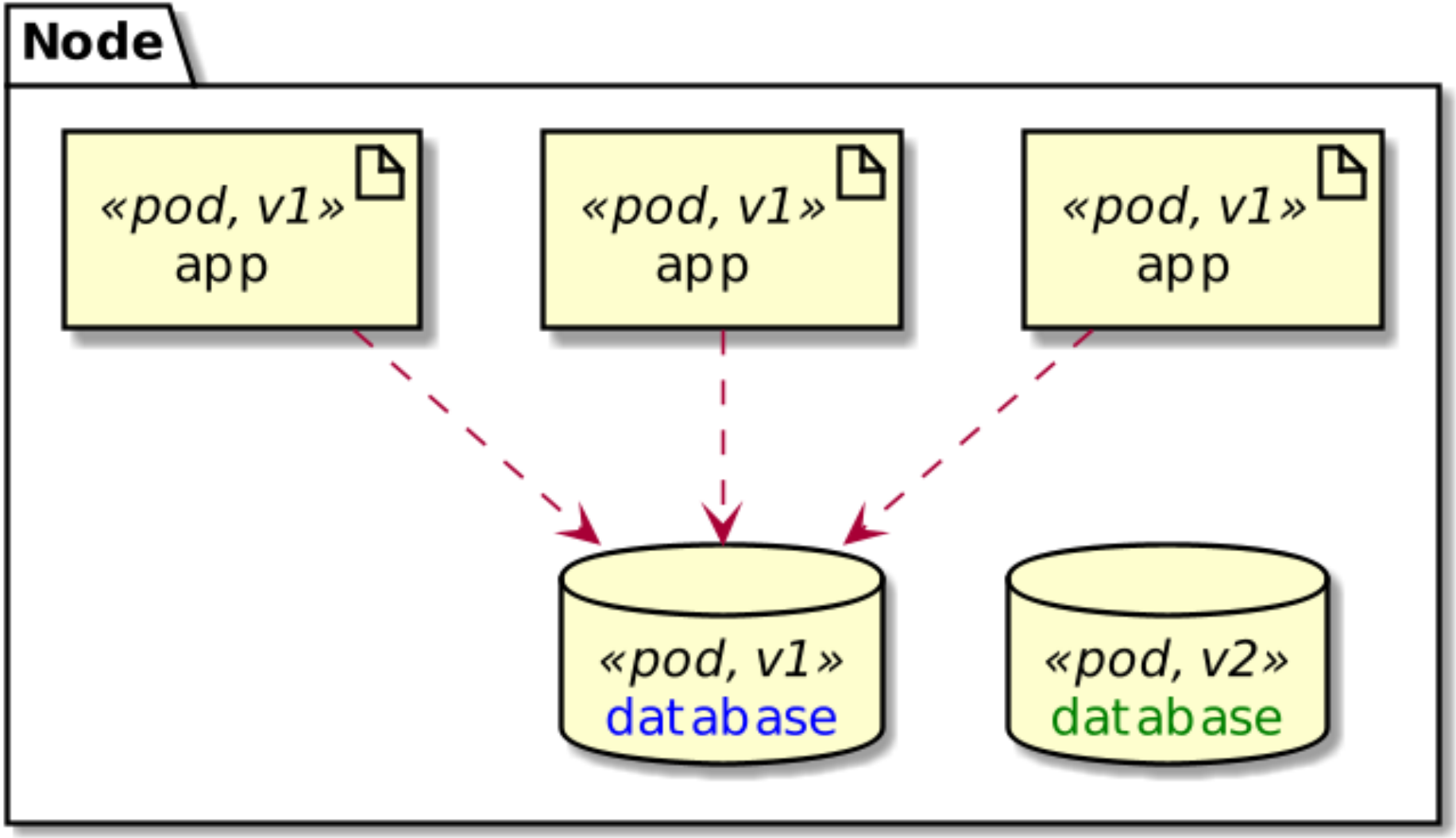
Change-Data-Capture

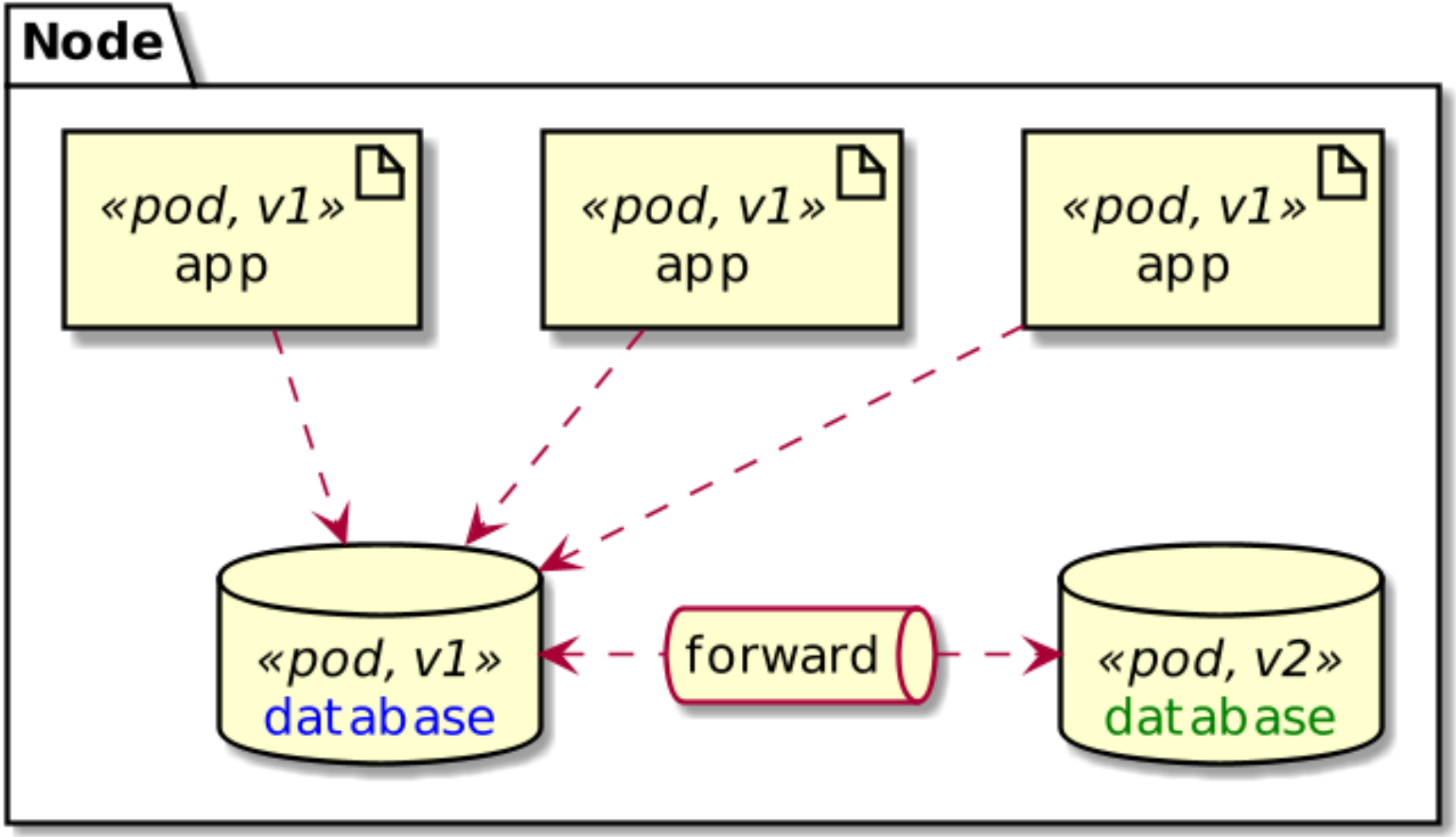
“In databases, Change Data Capture is a set of software design patterns used to **determine and track the data that has changed** so that action can be taken using the changed data.

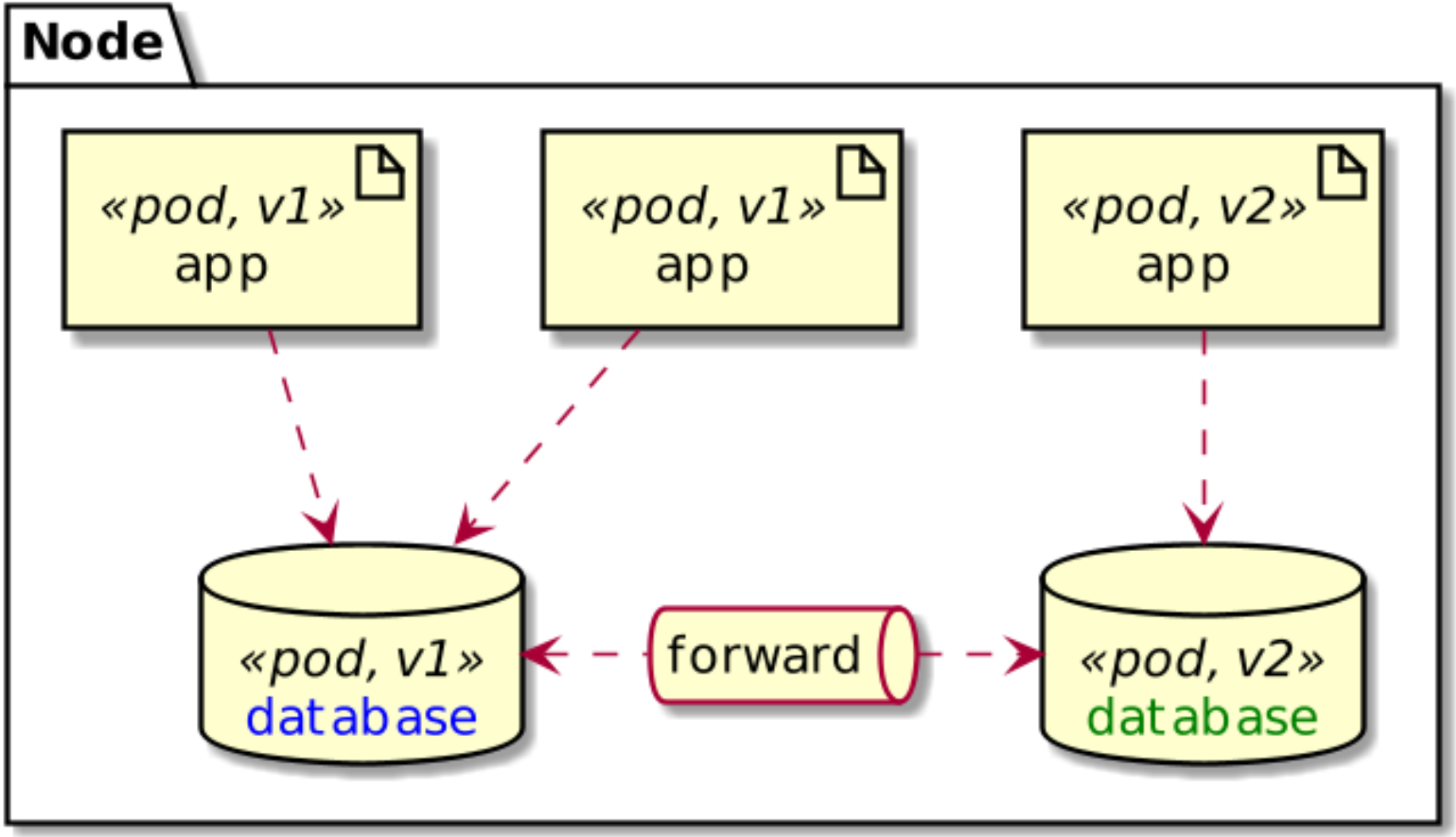
CDC is an approach to data integration that is based on the **identification, capture and delivery of the changes made to enterprise data sources**.”

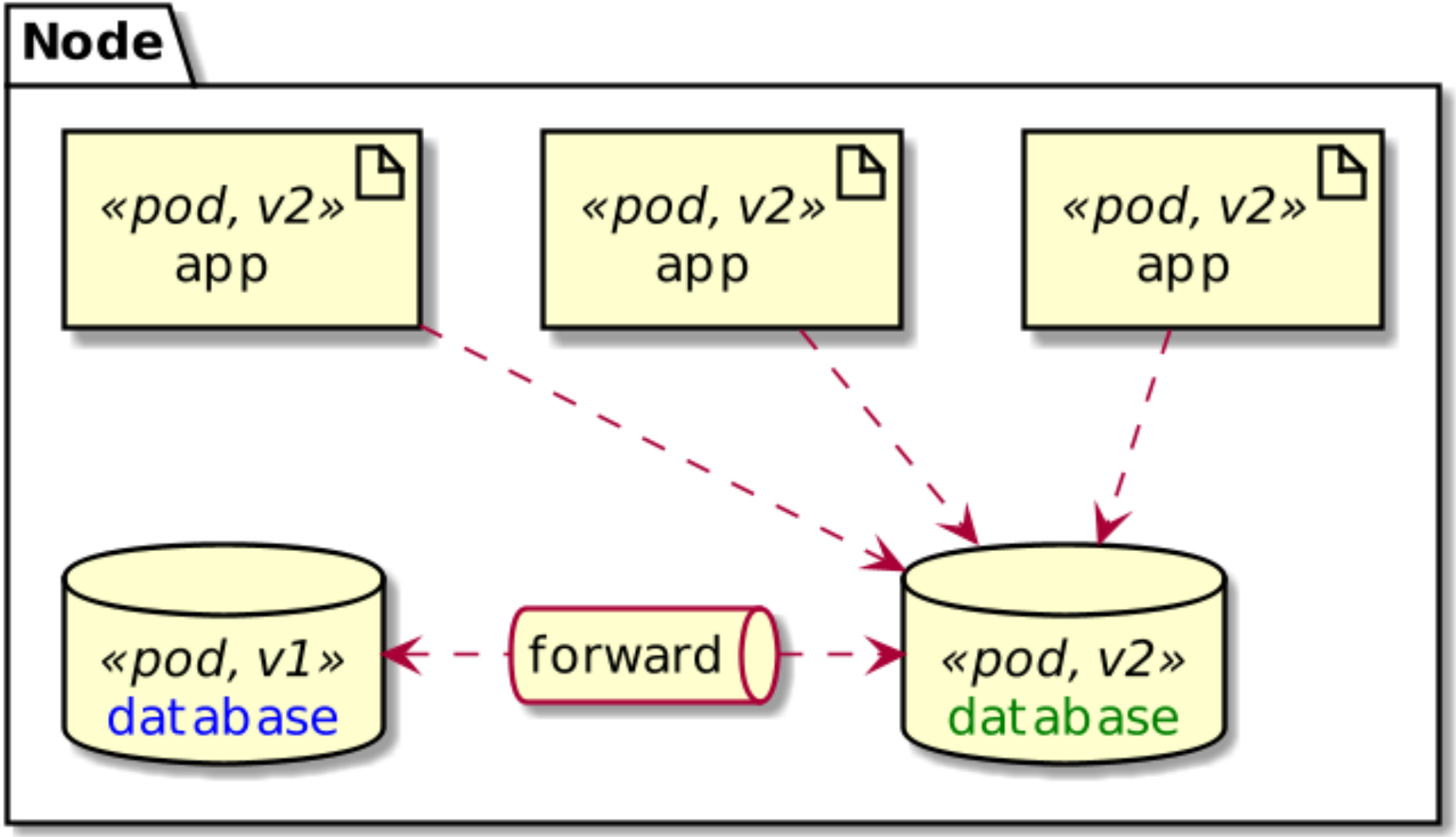
-- https://en.wikipedia.org/wiki/Change_data_capture

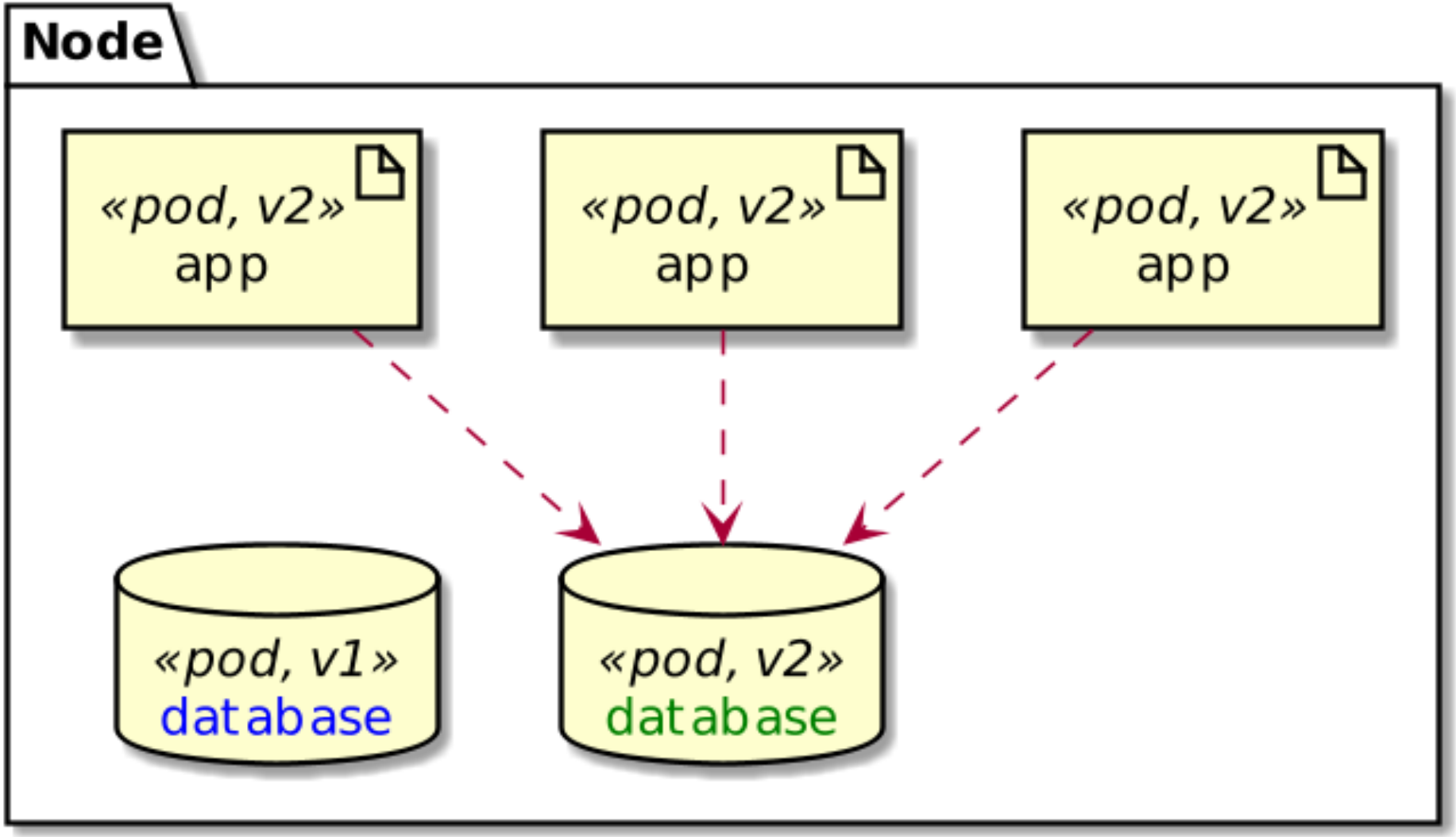

Node
«pod, v1»
app
«pod, v1»
app
«pod, v1»
app
«pod, v1»
database


Node
«pod, v1»
app
«pod, v1»
app
«pod, v1»
app
«pod, v1»
database
«pod, v2»
database


Node
«pod, v1»
app
«pod, v1»
app
«pod, v1»
app
«pod, v1»
database
forward
«pod, v2»
database


Node
«pod, v1»
app
«pod, v1»
app
«pod, v2»
app
«pod, v1»
database
forward
«pod, v2»
database


Node
«pod, v2»
app
«pod, v2»
app
«pod, v2»
app
«pod, v1»
database
forward
«pod, v2»
database


Node
«pod, v2»
app
«pod, v2»
app
«pod, v2»
app
«pod, v1»
database
«pod, v2»
database

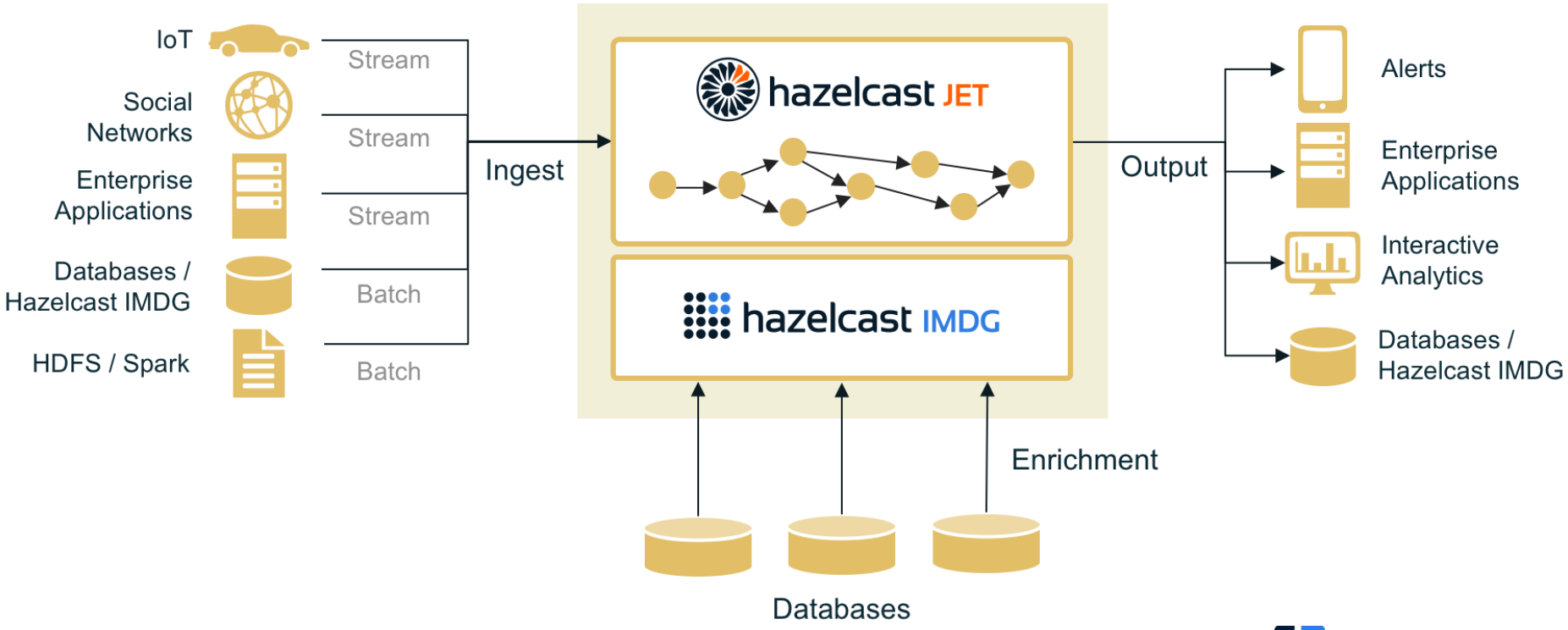
Implementation details

- Hazelcast for Session Replication
  - Via Spring Session
- Hazelcast for CDC
  - With Debezium

Hazelcast IMDG

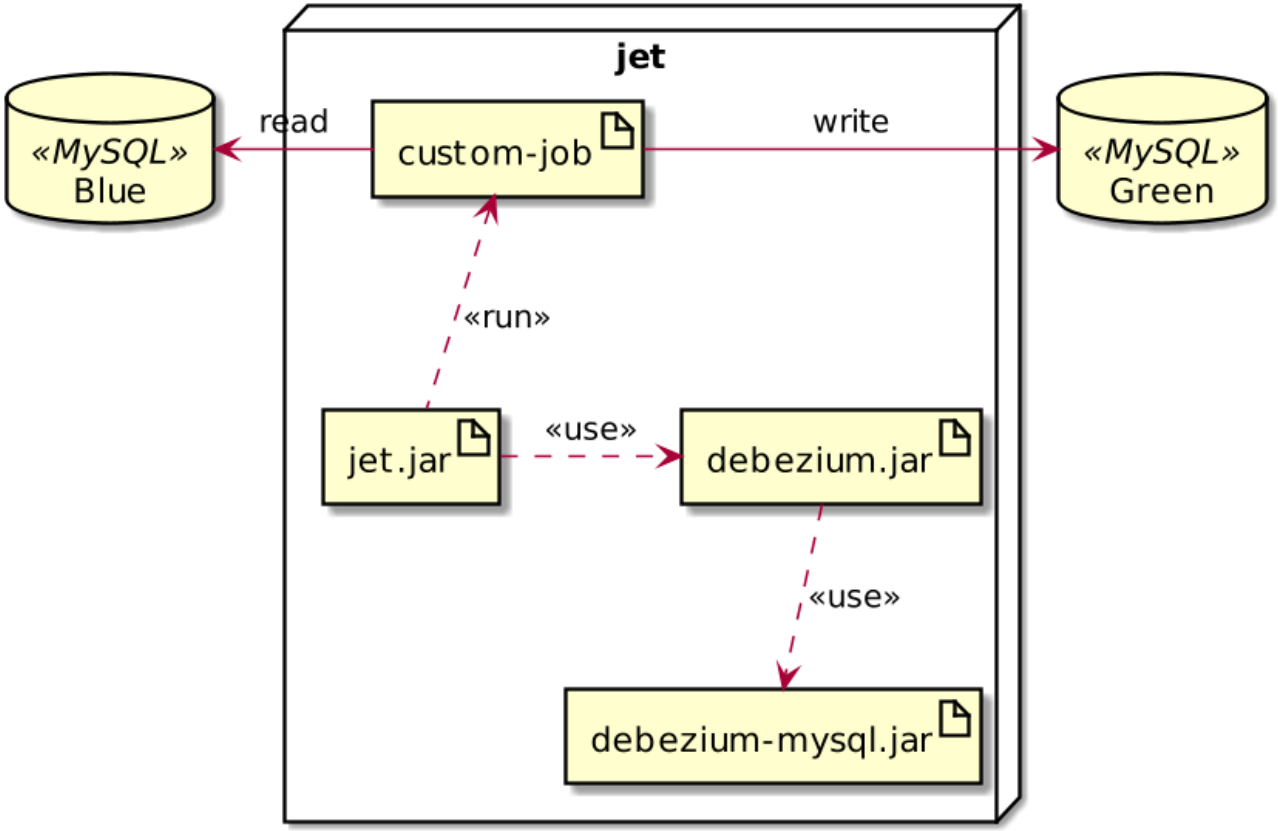
"Use Hazelcast IMDG to store your data in RAM, spread and replicate it across a cluster of machines, and perform data-local computation on it. ***Replication gives you resilience to failures of cluster nodes.***"

Hazelcast Jet



hazelcast®

Hazelcast Jet & Debezium




Time for DEMO

Takeaways

1. Zero-downtime is within your reach
2. Session replication
3. Change-Data-Capture + Data Streaming for the database

Thanks for your attention!

- https://blog.frankel.ch/
- @nicolas_frankel
- https://bit.ly/zero-downtime
- https://slack.hazelcast.com/
- https://training.hazelcast.com/