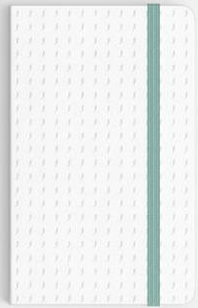
cantamen



Live migration of cantamen's primary database cluster

From MySQL 5.7 to MariaDB 10.11 in under 10 minutes

Dirk Hillbrecht,
cantamen co-founder, system architect and initial developer

Who we are & what we do

cantamen

- cantamen GmbH: Germany’s largest independent provider for car-sharing software and system solutions
  - Management software “EBuS” (since 1997)
  - Call-center for fleet users (since 2009)
  - In-vehicle hardware “Share Wizard” (since 2016)
- Founded in 2003, started with 3 people, now ~90 empolyees
- Today:
  - ~60 car-sharing providers
  - ~10.000 vehicles
  - ~120.000 users/customers

Our application

cantamen

- “EBuS” - “Elektronisches Buchungssystem” (“electronic booking system)
- Server written completely in Java (currently Java 17)
- Database access via Hikari connection pool
- ~80% hand-written SQL statements directly on JDBC
- ~20% JPA using Eclipselink
- Currently
  - One server serves everything
  - Some inflexibility regarding the database connection
    → *Shutdown during database update inevitable*

Our database

- MySQL 5.7 (after starting form 3.22 back in 1997)
- Primary master-slave cluster with three members
- Secondary slave attached to the cluster
- Ternary slaves attached to the secondary slave
- Homogenous replication: Everyone has the same data
- Master serves around 1.2*10^9 queries per day (1.2 millards/billions)

The migration

cantamen

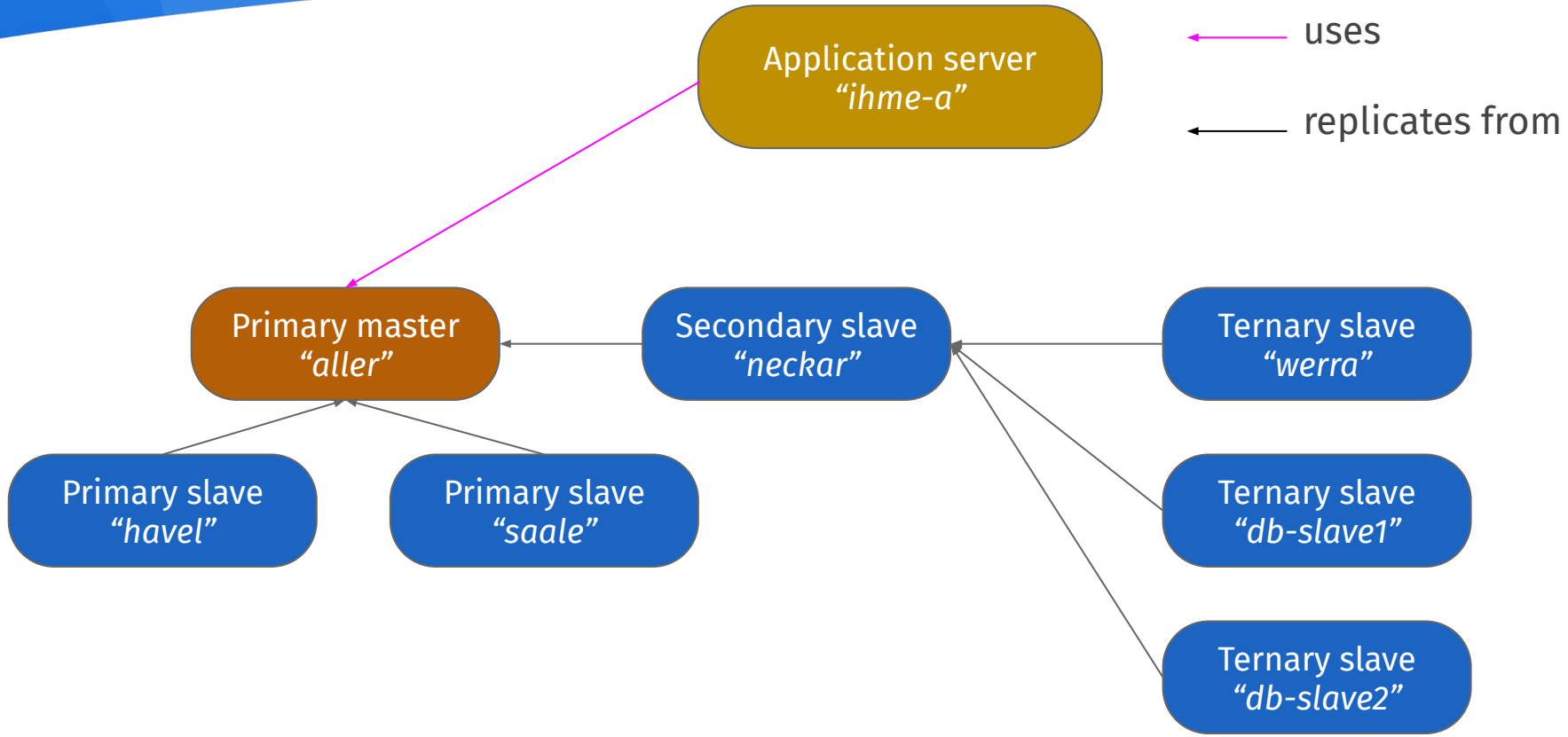
- MySQL 5.7 out of support since August 2023
- Out of all modern Linux distributions
- MariaDB impressed in side projects, esp. regarding updates
- Migration must happen without long production stop
- Everything is to be migrated

- *Known procedure:* MariaDB replicates from MySQL
- *Unknown procedure:* MySQL replicates from MariaDB

Stick with known procedures

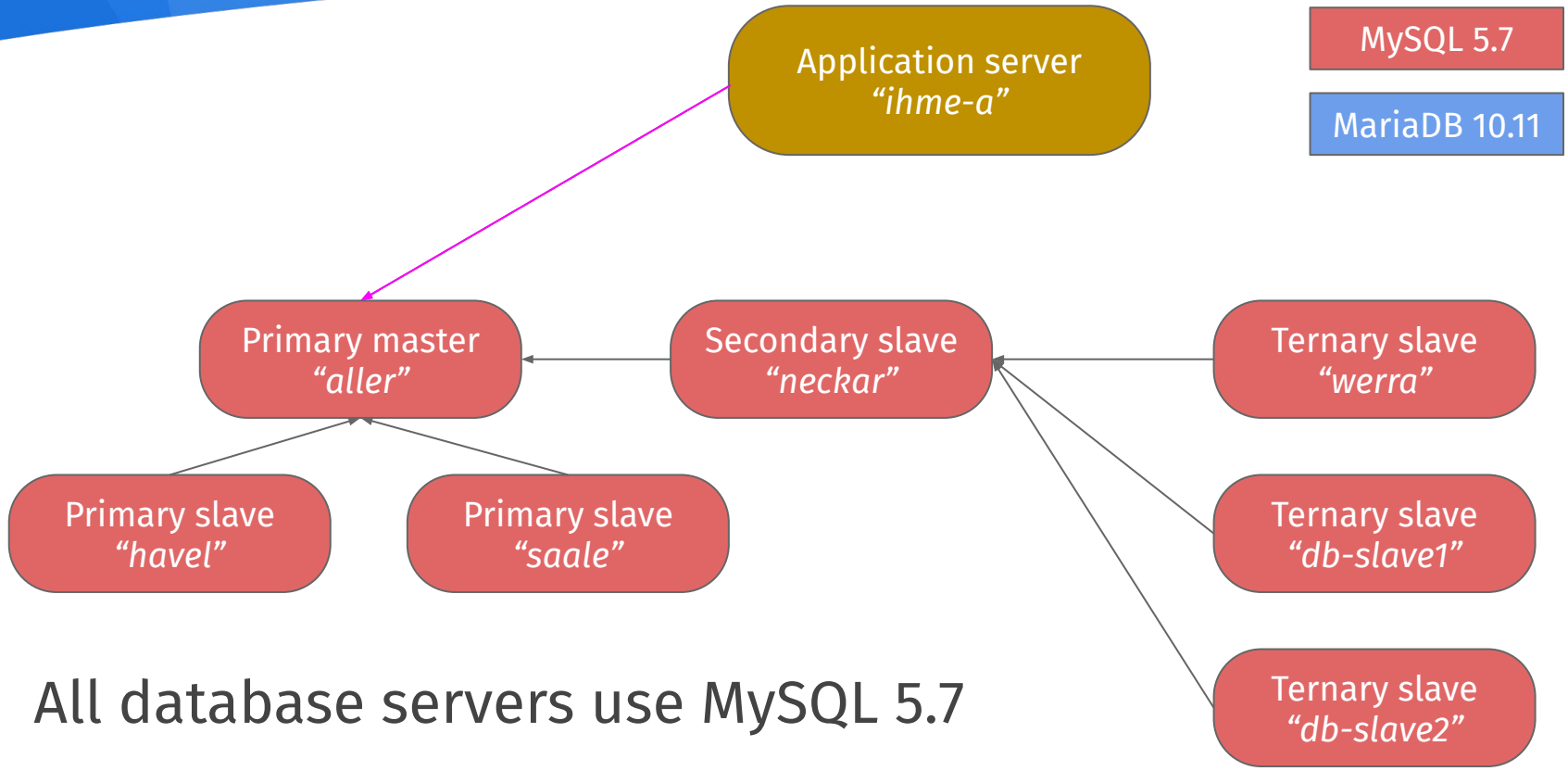
The original setup

cantamen



Database server system

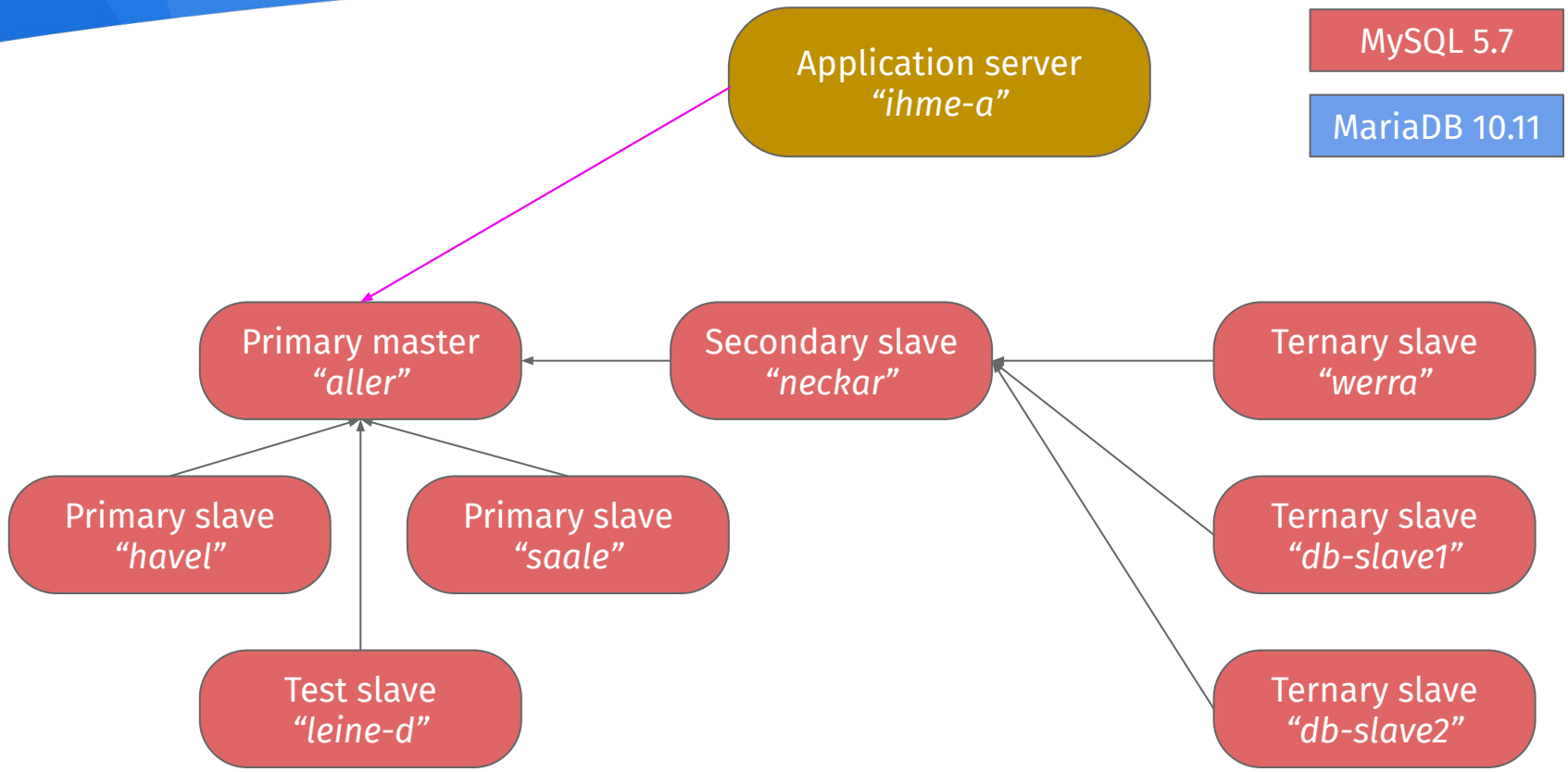
cantamen



All database servers use MySQL 5.7

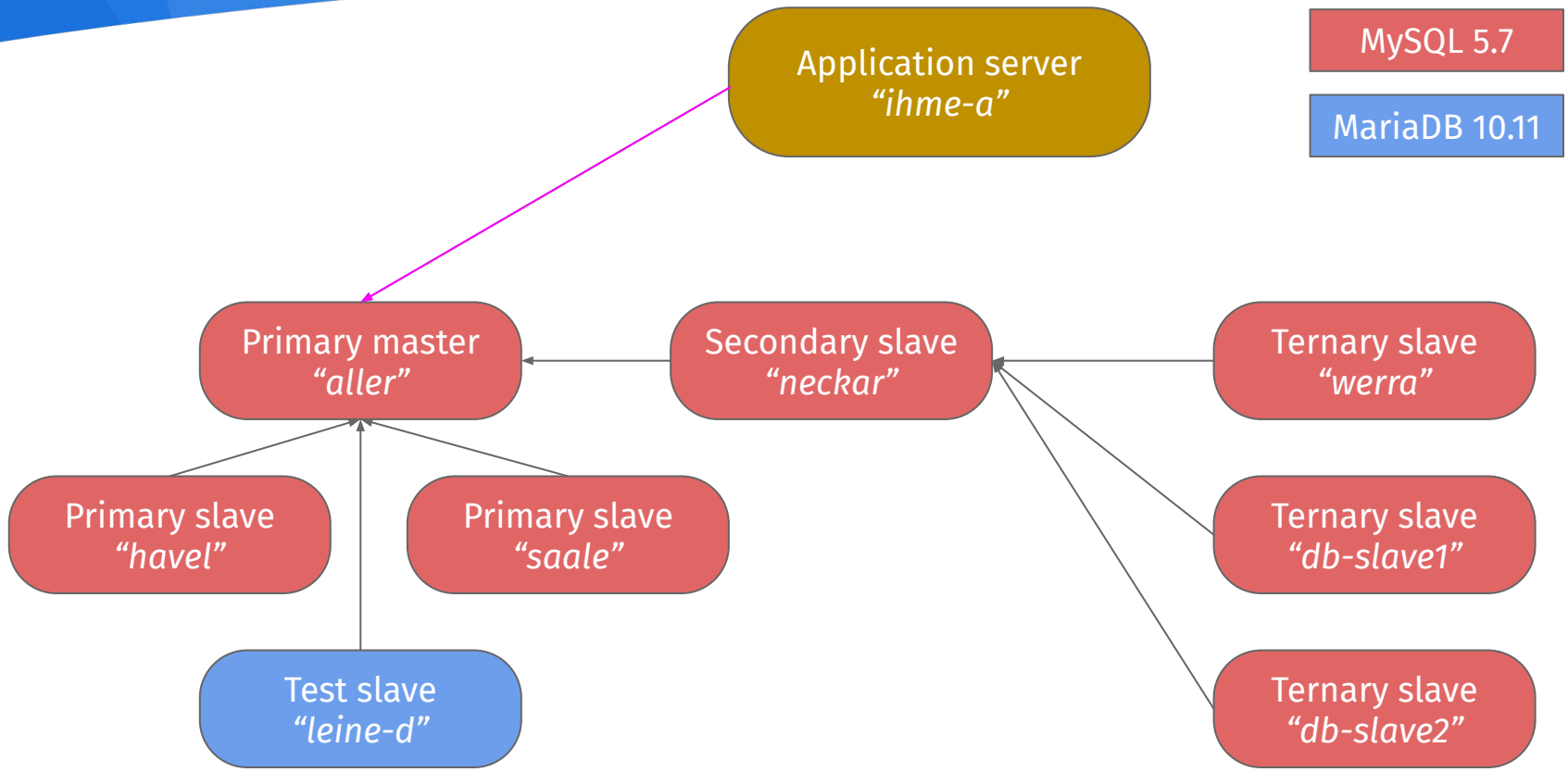
Adding a new slave

cantamen



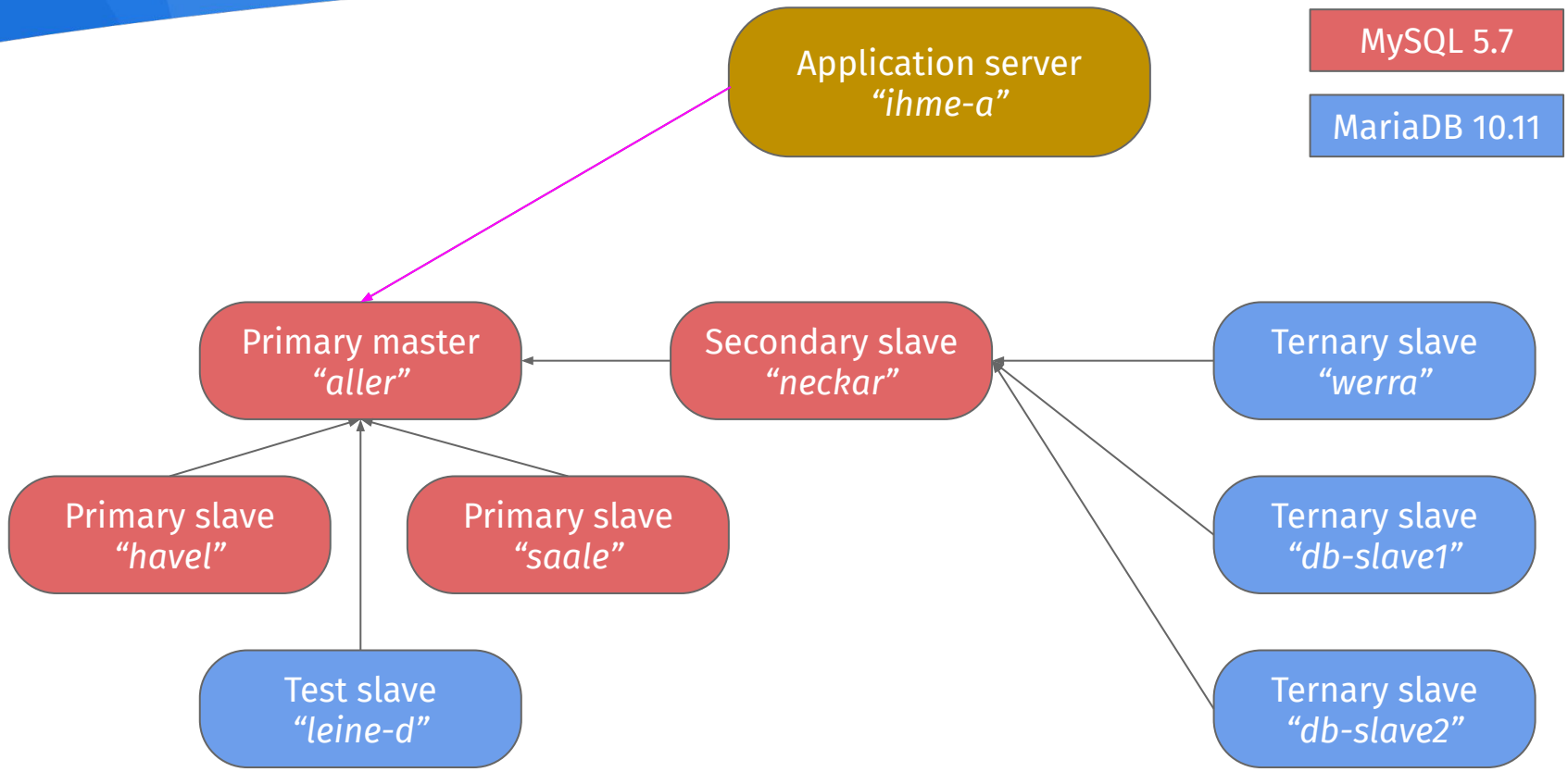
Converting the test slave to MariaDB

cantamen



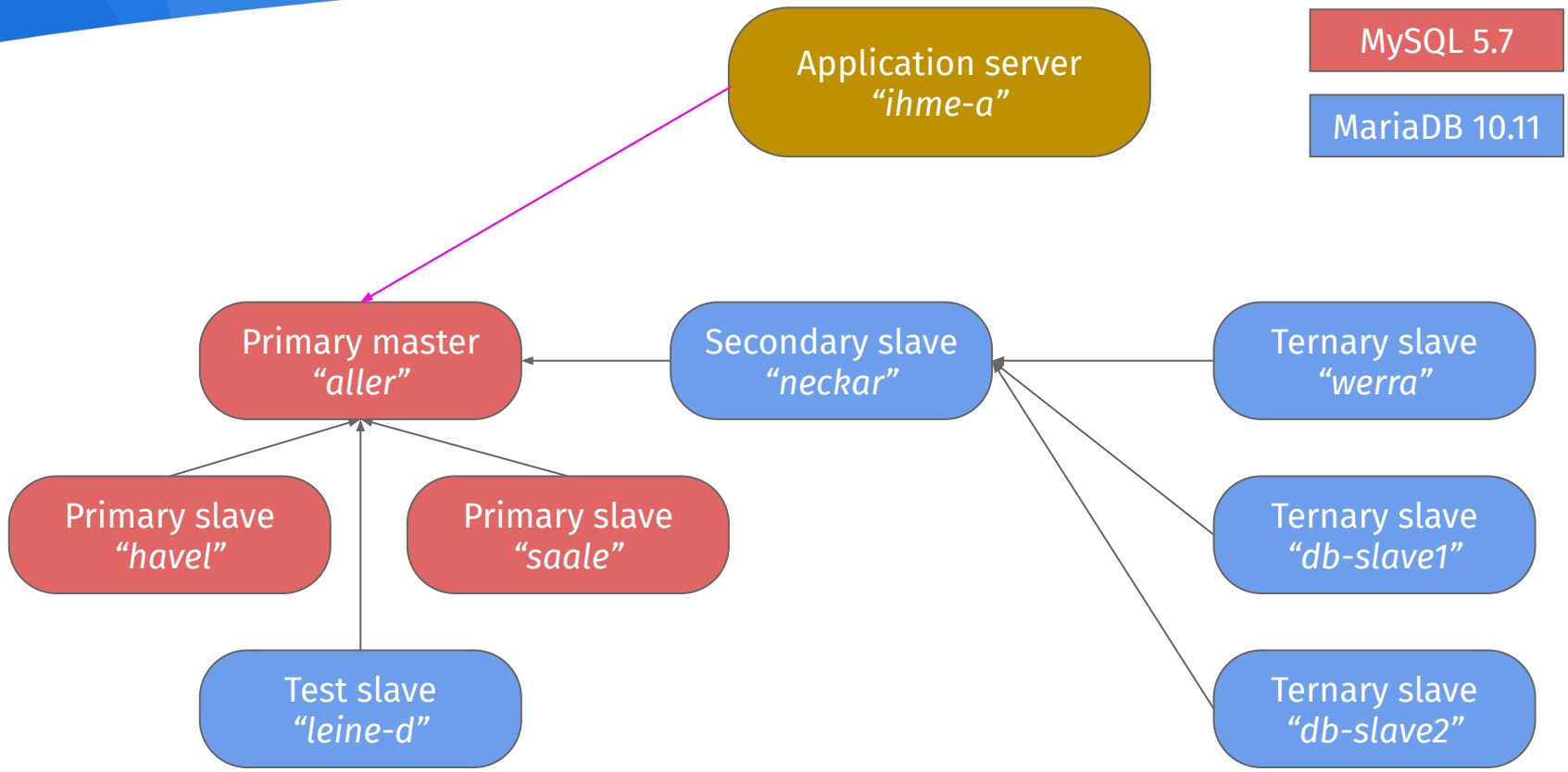
Converting the ternary slaves

cantamen



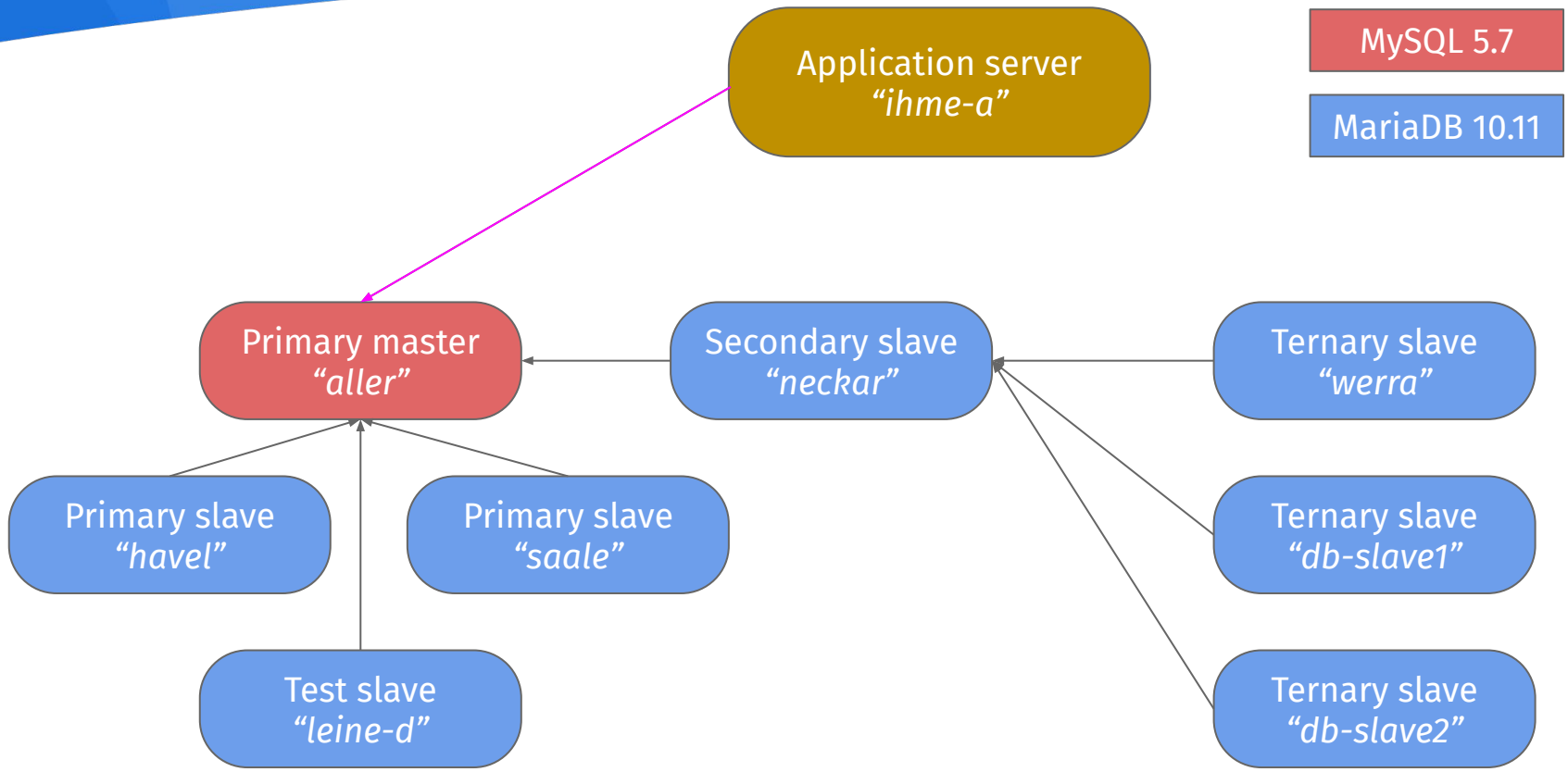
Converting the secondary slave

cantamen



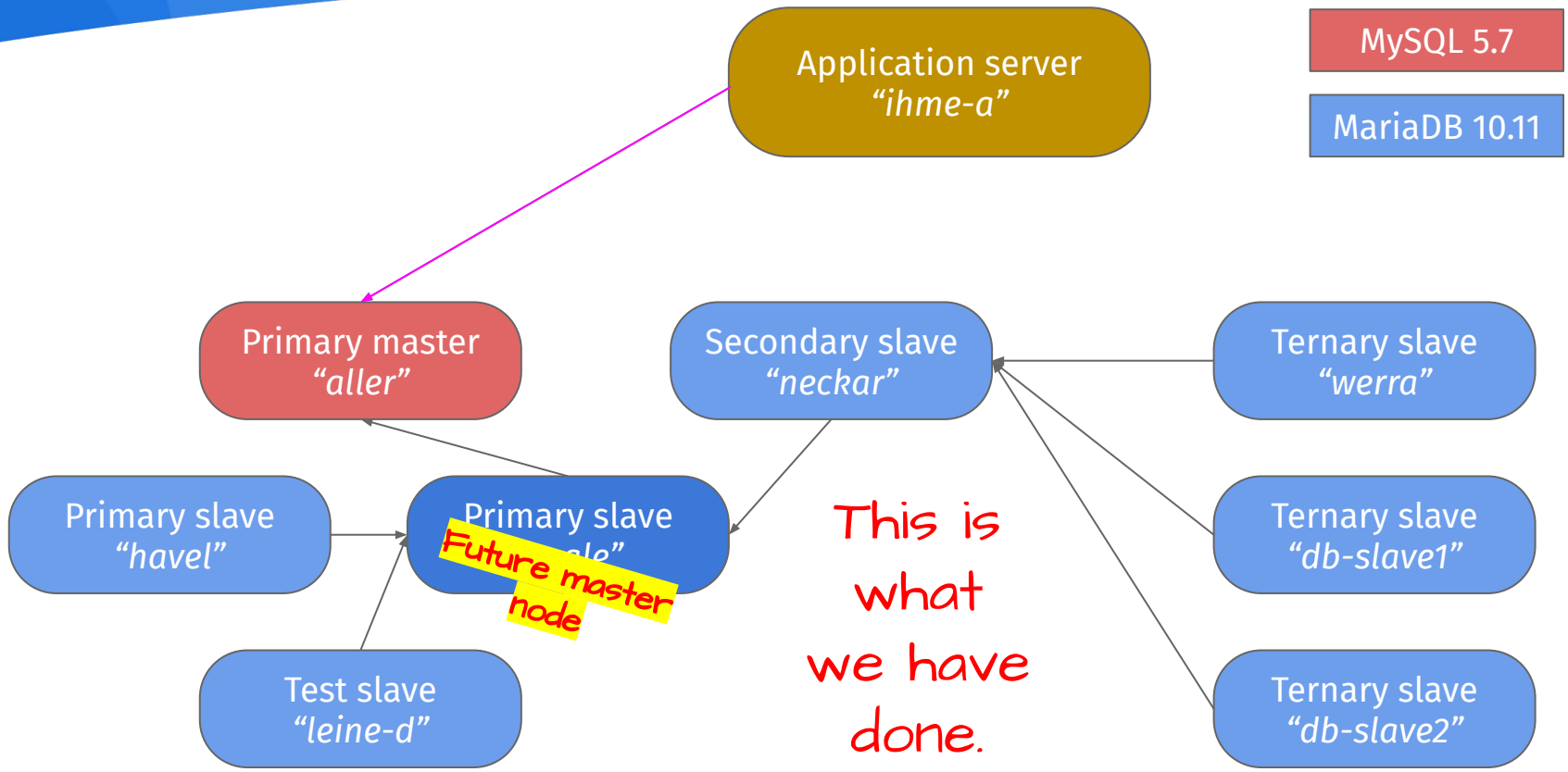
Converting the primary slaves

cantamen



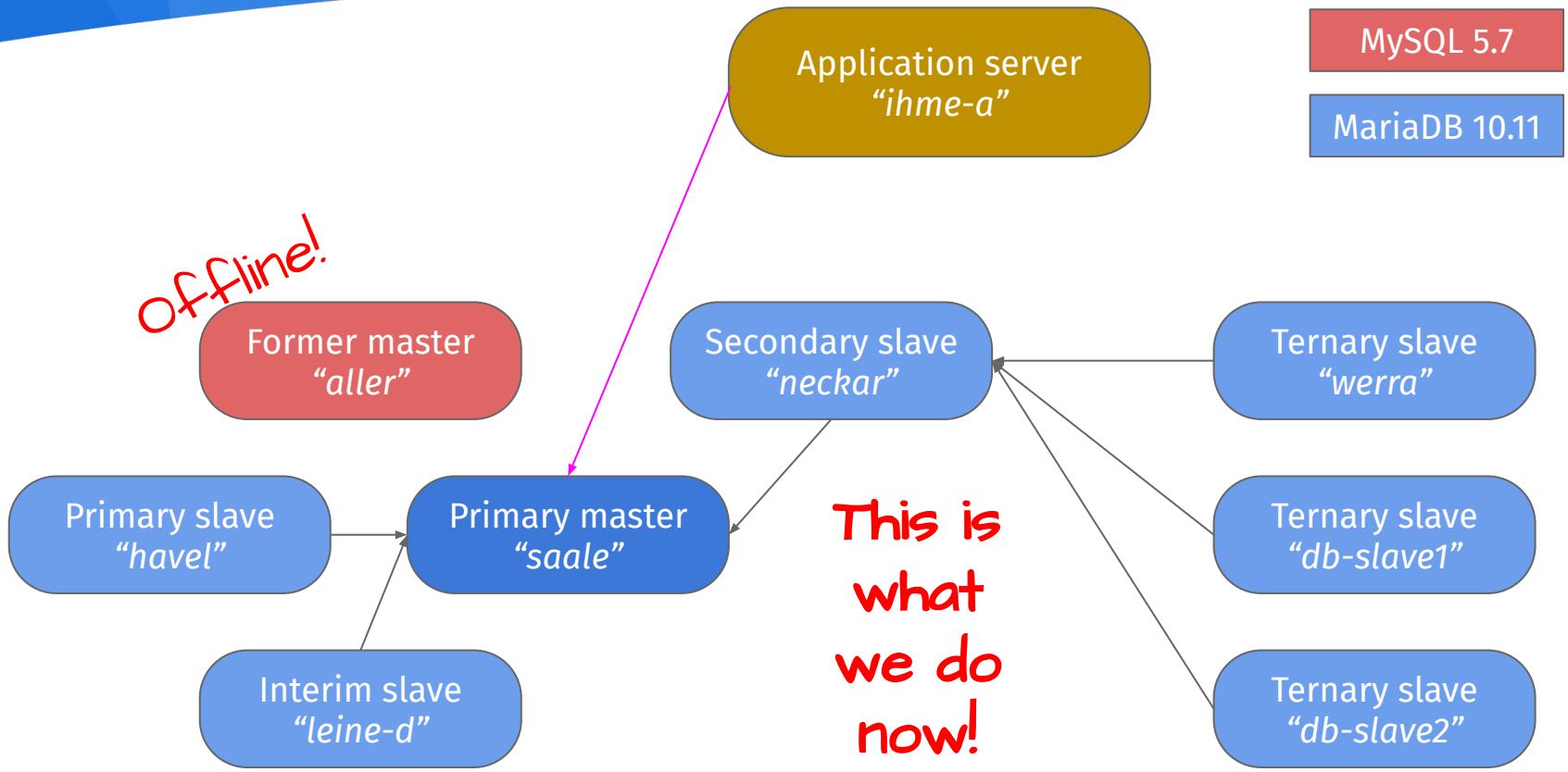
Prepare future master node

cantamen



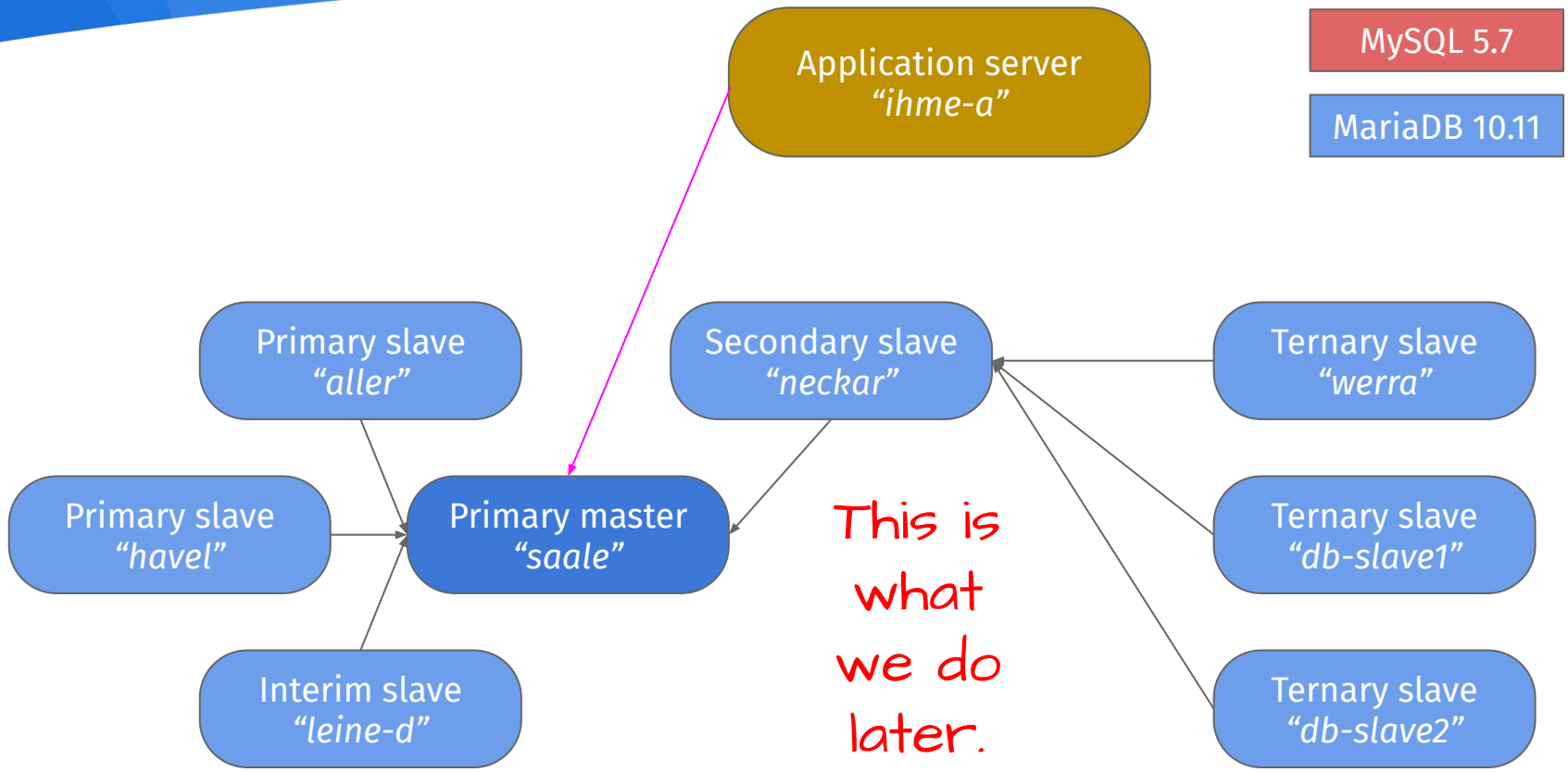
Switch over master node

cantamen



Convert old master to slave (and MariaDB)

cantamen



How do we do it

cantamen

- Shut down application
- Shut down former primary master
- Change “`dbmaster`” alias in DNS and `/etc/hosts`
- Restart application

Note 1: No database shutdown or restart, no configuration change in database or application.
Note 2: Application shutdown due to restriction in application’s framework, not strictly needed by MariaDB or MySQL.

Enough of all that theory...

Time for action!

Conversion of one node to MariaDB

- Stop replication: `STOP SLAVE io_thread`, wait, `STOP SLAVE`
- Note replication coordinates (`RELAY_MASTER_LOG_FILE`, `EXEC_MASTER_LOG_POS`)
- Stop MySQL, start MariaDB in the same data
- Run mariadb-upgrade
- In theory: `START SLAVE` and it continues
- In practice: `RESET SLAVE`, `CHANGE MASTER TO`, `START SLAVE`
- Some weirdness in the binary/relay logs

How did we know it will work?

cantamen

- Convert development system to MariaDB, develop against it
- Convert integration test servers to MariaDB, run test suite
- Add MariaDB production node, replicate all write operations
- Migrate internal manual test systems for special purposes
- Migrate preview server systems, run manual operations on larger scale
- Throw large chunks of real-world-read operations to servers

Did we have a desaster recovery strategy?

cantamen

- Switch back to MySQL after migration? Difficult due to replication direction.
- Only possible scenario: Unforeseen performance issues
- If needed: Go to unknown procedure:
  - Attach MySQL-running aller to MariaDB-running master saale
  - Let it catch up
  - Stop application & move it back to saale
  - Make aller again master to saale
  - Fix the issue & try again.