Group 11: amazing

MongoDB, PostgreSQL, RabbitMQ, Flask

…to serve and protect data.

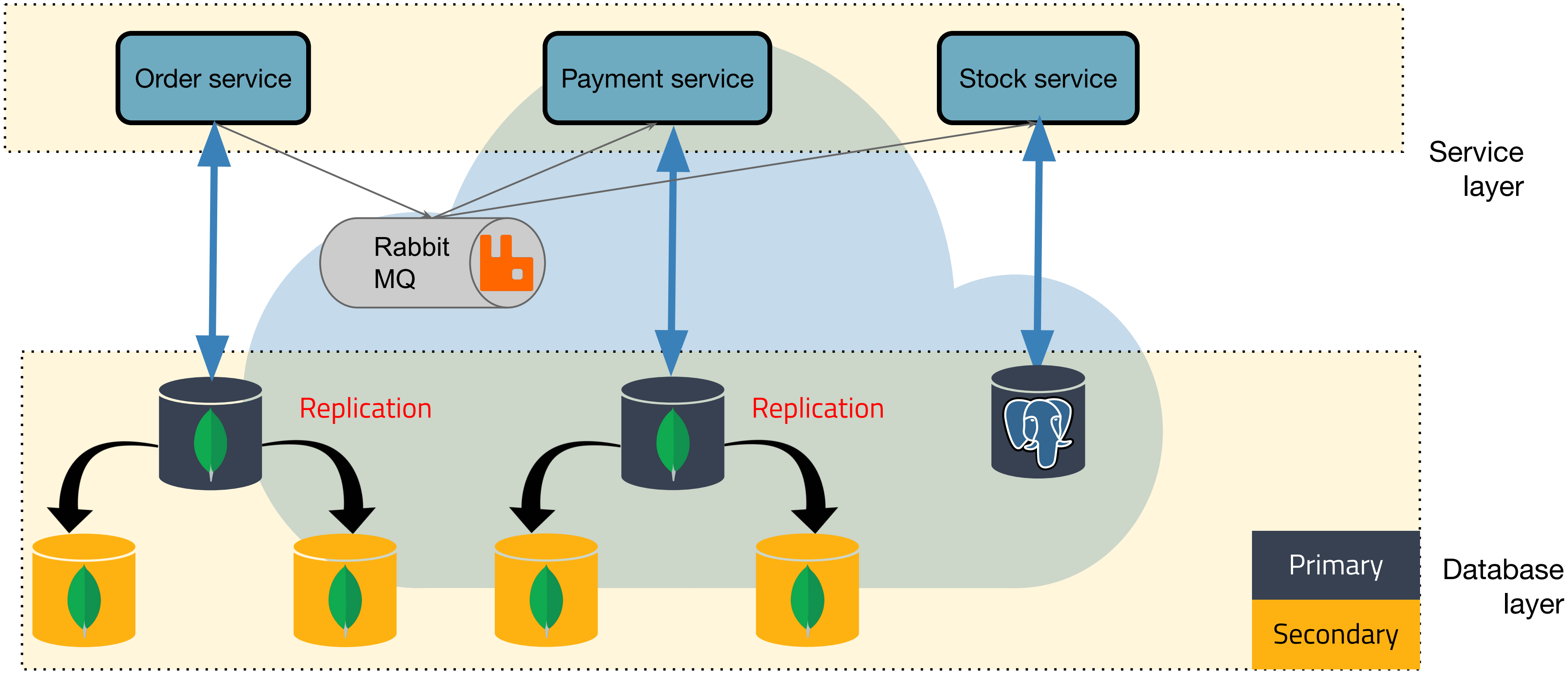
Pascal Benschop, Florian Ecker-Eckhofen,
Casper Hildebrand, Leon Kempen, Gustav Nobert,

TU Delft

Tech Stack

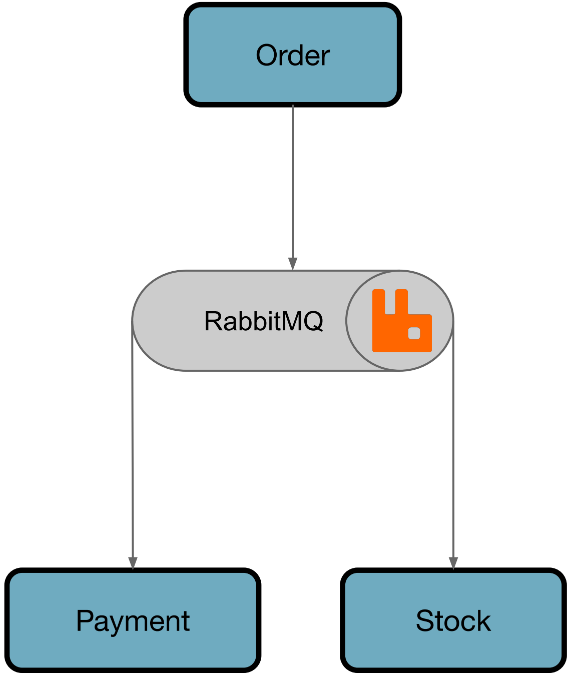
- Flask for services
- MongoDB
- PostgreSQL
- RabbitMQ

A tale of three Cloud services



Message queue

- SAGAS protocol
- Eventual consistency
- Best-effort delivery

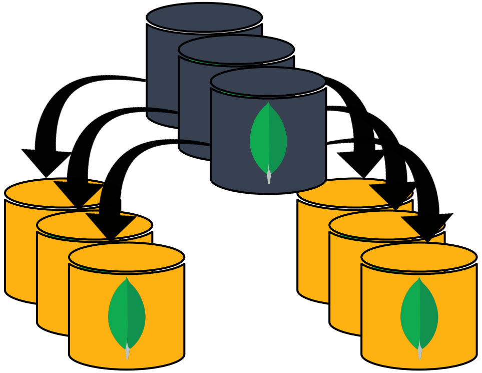


amazing

Replication

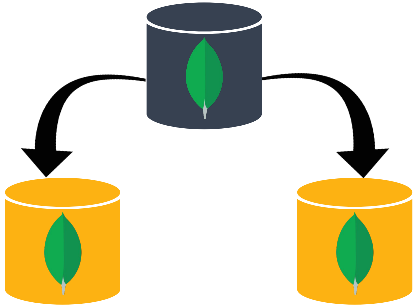
- MongoDBs are split into three shards with three replicas each

- 2 routers forward the incoming requests.
- 3 configservers host handle the replica's



Fault tolerance

- Requests are **idempotent**; each request contains an idempotency key
- Databases are replicated, meaning the application is more fault-tolerant

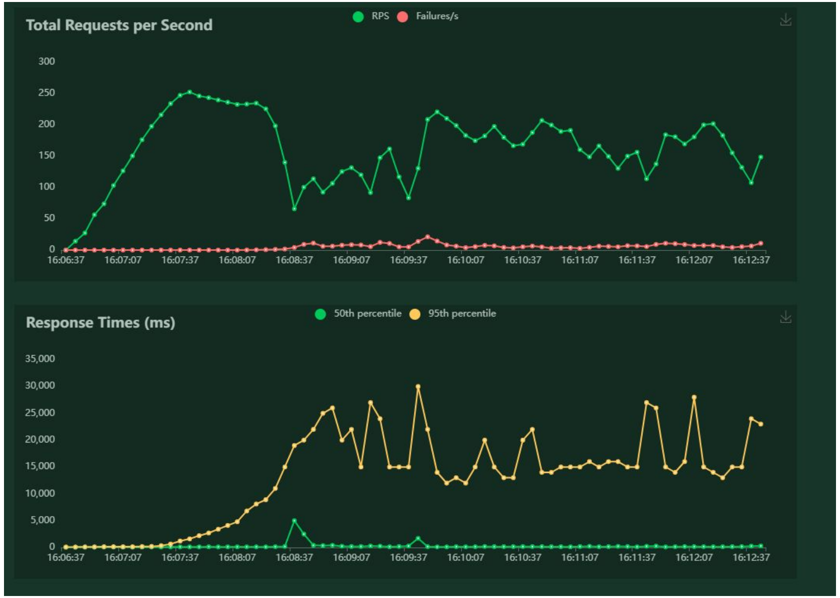


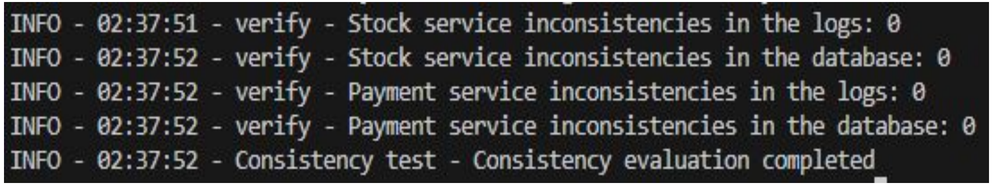
amazing

Honorable mention

- Redis cluster with lock manager for stock management
- Little to no documentation
- Hard to migrate to k8s

Consistency vs. performance





```
INFO - 02:37:51 - verify - Stock service inconsistencies in the logs: 0
INFO - 02:37:52 - verify - Stock service inconsistencies in the database: 0
INFO - 02:37:52 - verify - Payment service inconsistencies in the logs: 0
INFO - 02:37:52 - verify - Payment service inconsistencies in the database: 0
INFO - 02:37:52 - Consistency test - Consistency evaluation completed
```

amazing

What we would do better if we had more time

- Implement sharding for PostgreSQL
- Replace HTTP between services
- Make the API async

Reflection

- Start earlier with migrating to k8s
- Don’t overengineer before a basis
- Have a plan B ready when experimenting