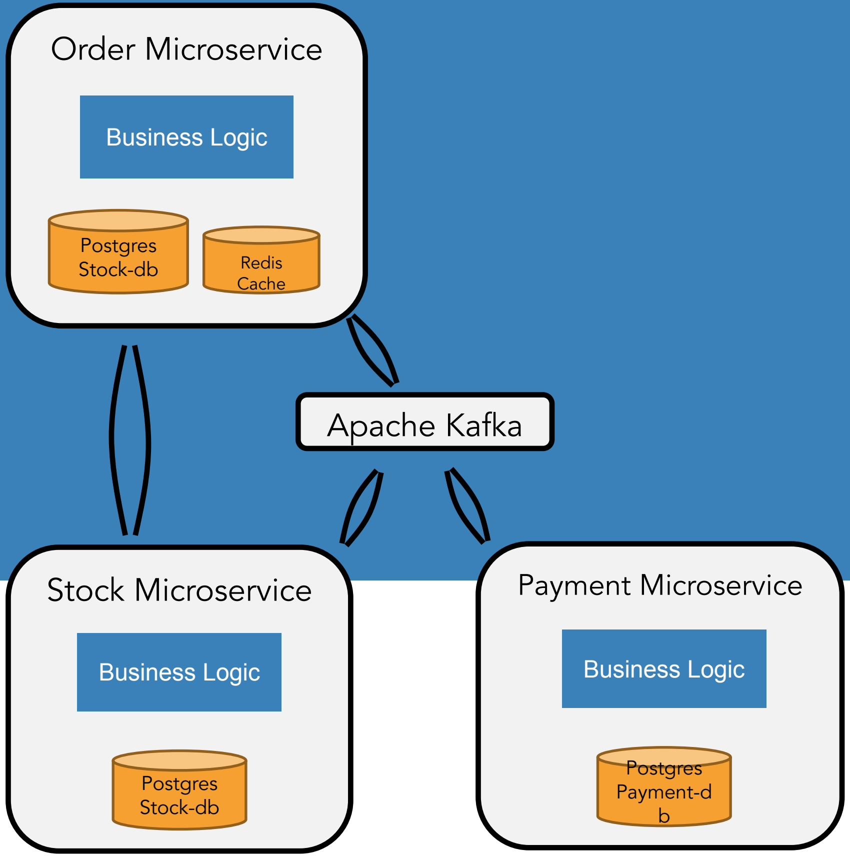
Team Number: 8

IN4331 - WSDB project / Python Flask Apache Kafka

Pierluigi Negro (5897300), Nicky Ju (4856708),
Ee Xuan Tan (4907531), Bryan He (48153000)

TU Delft

Systems used



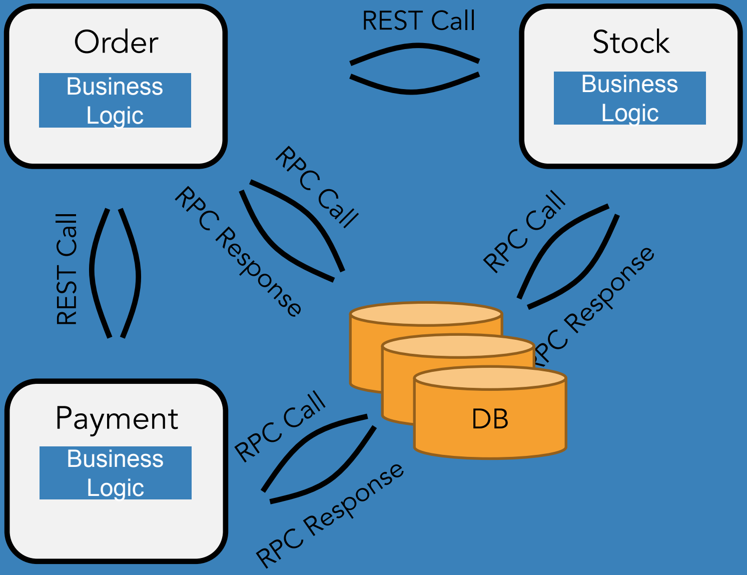
PostgreSQL

APACHE kafka®
A distributed streaming platform

redis

Our Solution (pt. 1)

Started with a Flask Backend with the 3 microservices and connected it with a PostgreSQL central database, serving the 3 microservices.

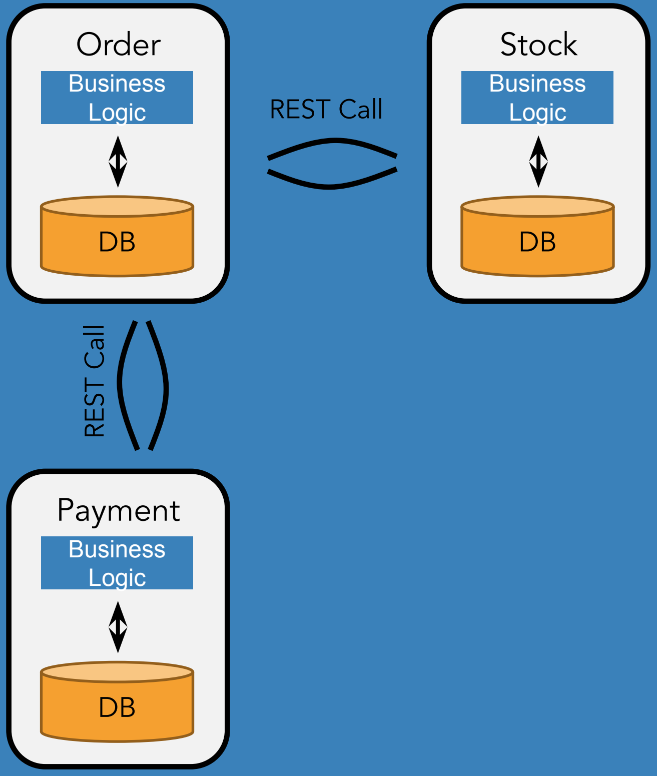


PostgreSQL

Flask

Our Solution (pt. 2)

Moved on to splitting the database for the different microservices and started caching data using redis. This way fewer REST Calls were necessary to improve the performance.



Order

Business Logic

DB

REST Call

Stock

Business Logic

DB

REST Call

Payment

Business Logic

DB

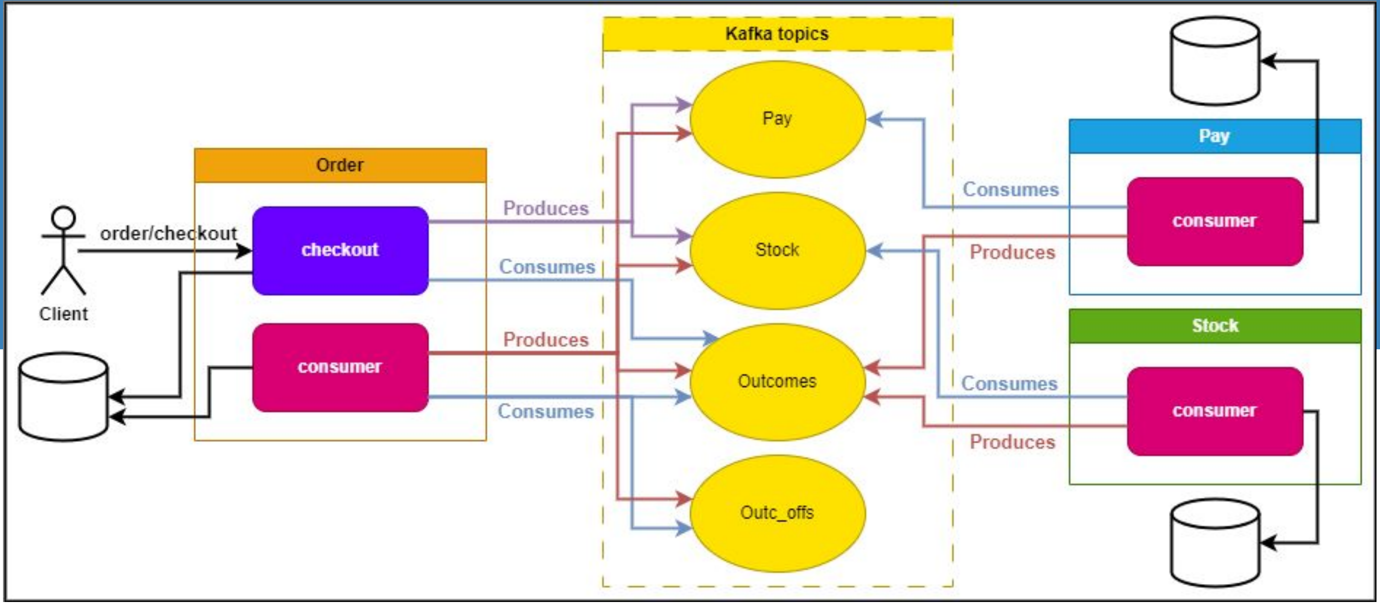
Our Solution (final)

Implemented design pattern SAGA’s

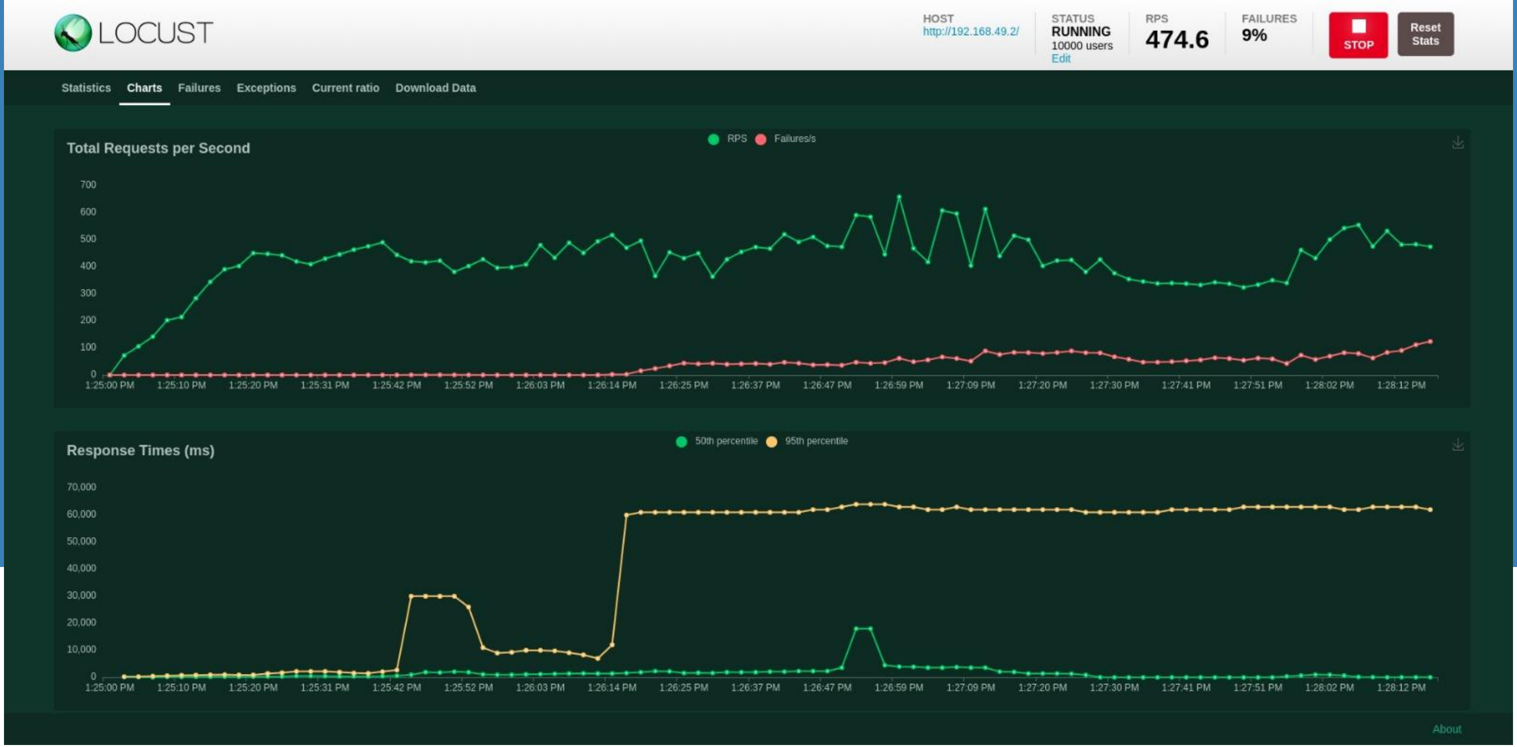
Guarantees eventual Consistency, making use of idempotency keys

Scalability - Partitions/Consumers

Fault tolerance - Checks for failing database



Results



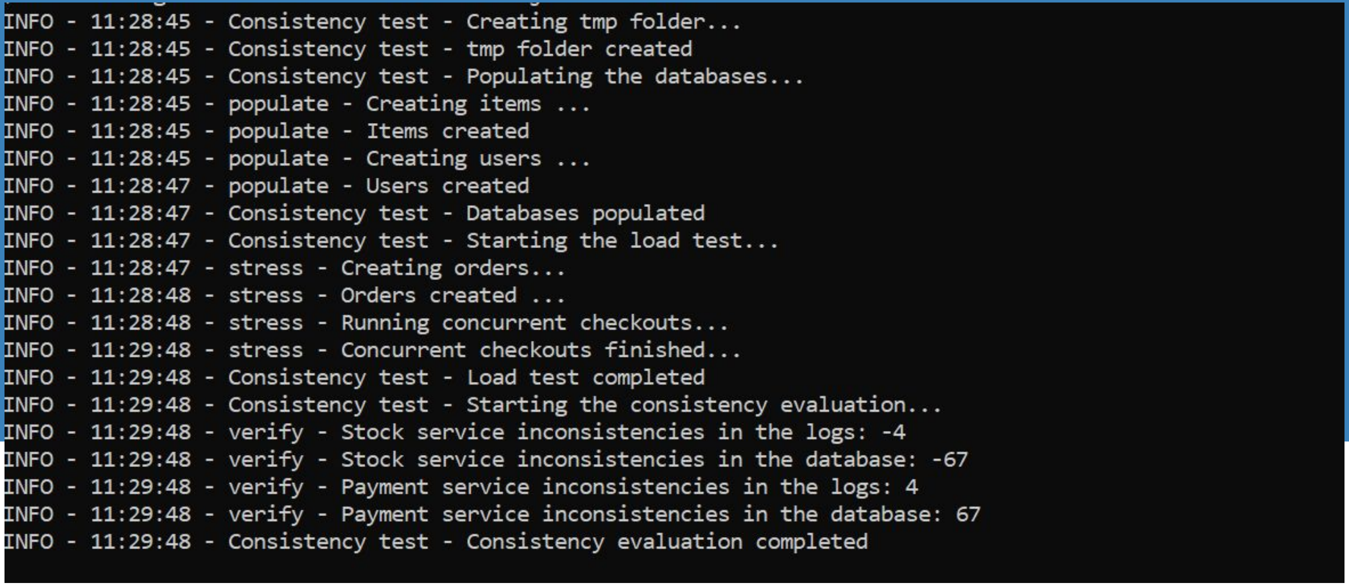
HOST http://192.168.49.2/

STATUS RUNNING
10000 users
Edit

RPS 474.6

FAILURES 9%

Results



```
INFO - 11:28:45 - Consistency test - Creating tmp folder...
INFO - 11:28:45 - Consistency test - tmp folder created
INFO - 11:28:45 - Consistency test - Populating the databases...
INFO - 11:28:45 - populate - Creating items ...
INFO - 11:28:45 - populate - Items created
INFO - 11:28:45 - populate - Creating users ...
INFO - 11:28:47 - populate - Users created
INFO - 11:28:47 - Consistency test - Databases populated
INFO - 11:28:47 - Consistency test - Starting the load test...
INFO - 11:28:47 - stress - Creating orders...
INFO - 11:28:48 - stress - Orders created ...
INFO - 11:28:48 - stress - Running concurrent checkouts...
INFO - 11:29:48 - stress - Concurrent checkouts finished...
INFO - 11:29:48 - Consistency test - Load test completed
INFO - 11:29:48 - Consistency test - Starting the consistency evaluation...
INFO - 11:29:48 - verify - Stock service inconsistencies in the logs: -4
INFO - 11:29:48 - verify - Stock service inconsistencies in the database: -67
INFO - 11:29:48 - verify - Payment service inconsistencies in the logs: 4
INFO - 11:29:48 - verify - Payment service inconsistencies in the database: 67
INFO - 11:29:48 - Consistency test - Consistency evaluation completed
```

Future work

Write more tests
Compare it with a 2 phase lock implementation
Try other databases out
Shard the database
Database replication
Try benchmarking in AWS cloud instead of on local machines (more machine resources)