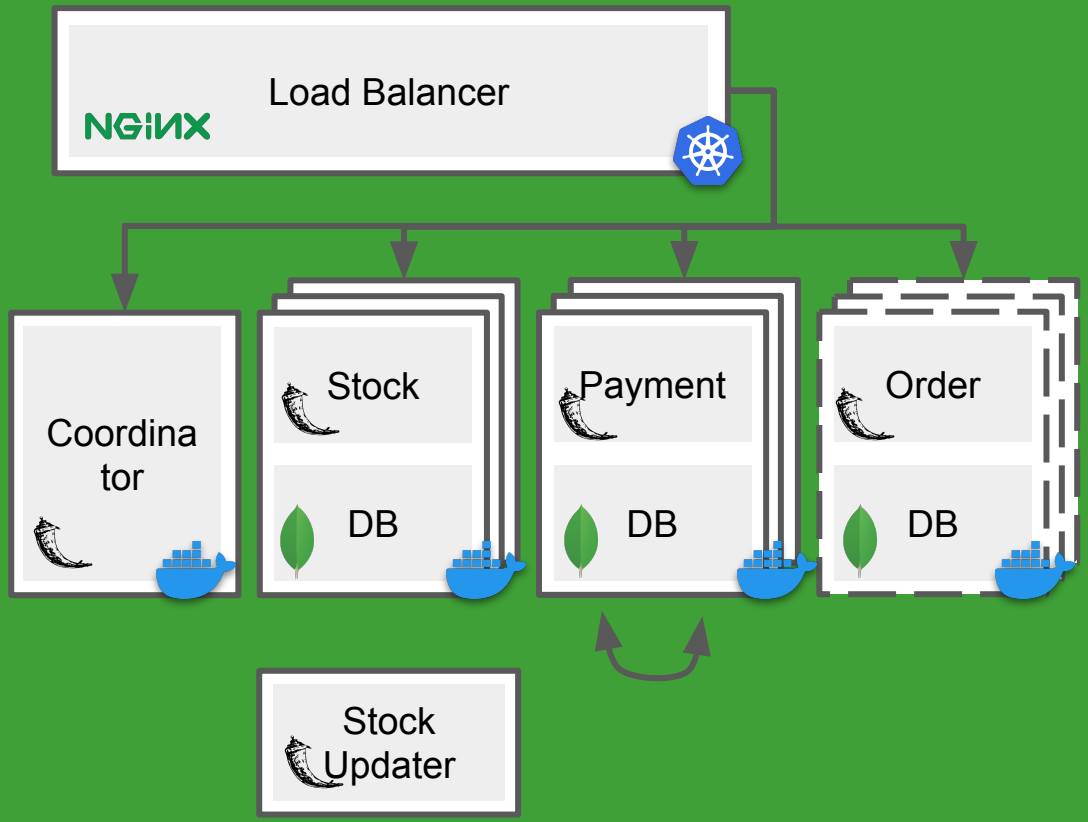
Group 9

Web scale Data Management
Final Presentation

Overview of the Architecture



mongoDB.

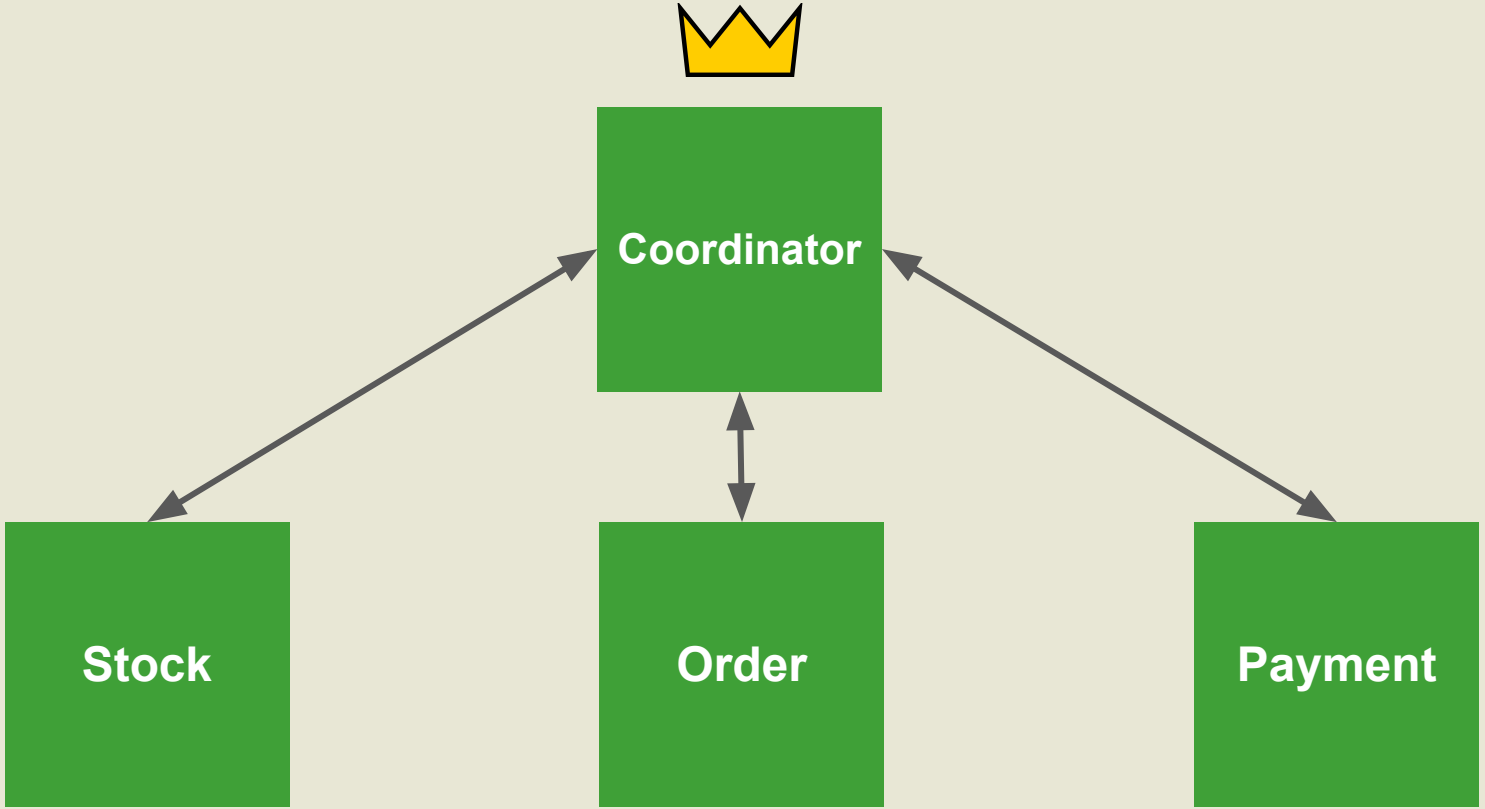
Flask

NGINX

Distributed transactions

How do we guarantee consistent transactions between the stock and payment service?

2 Phase Commit


Coordinator
Stock
Order
Payment

Problem

2PC is a blocking protocol

Items and users are locked during payment

Bad Availability

Solution

Modify the 2PC protocol to not block the items

Comparison modified 2PC

Consistency

```
verify - Stock service inconsistencies in the logs: 1
verify - Stock service inconsistencies in the database: 1
verify - Payment service inconsistencies in the logs: 1
verify - Payment service inconsistencies in the database: 3.0
```

Replication Consistency Strategies

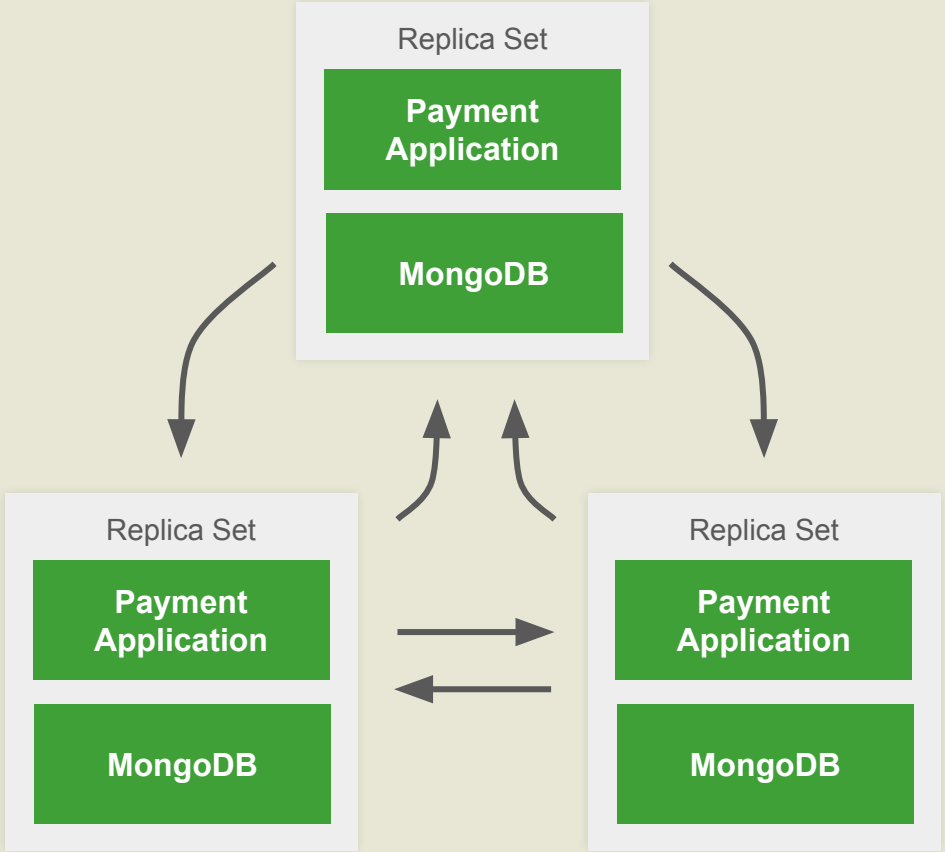
Payment	Stock	Order
Replicas	Replicas	Shards
Paxos	Read Quorum Write Quorum	User id

Durability & Consistency

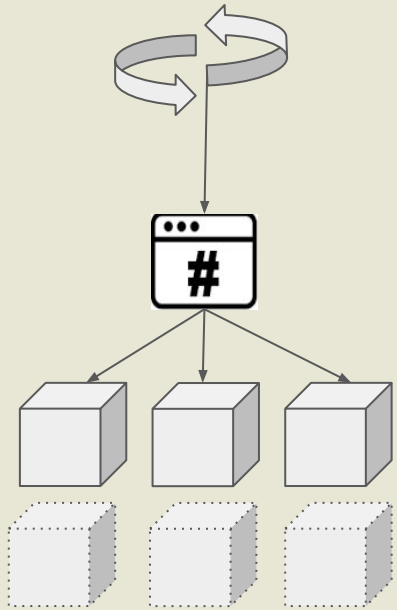
Availability

How we use PAXOS?

- Payment Service
- Implementation details
  - Phase 1
    - Prepare
    - Promise
  - Phase 2
    - Accept
    - Accepted
  - Unique paxos round id
  - Transaction id

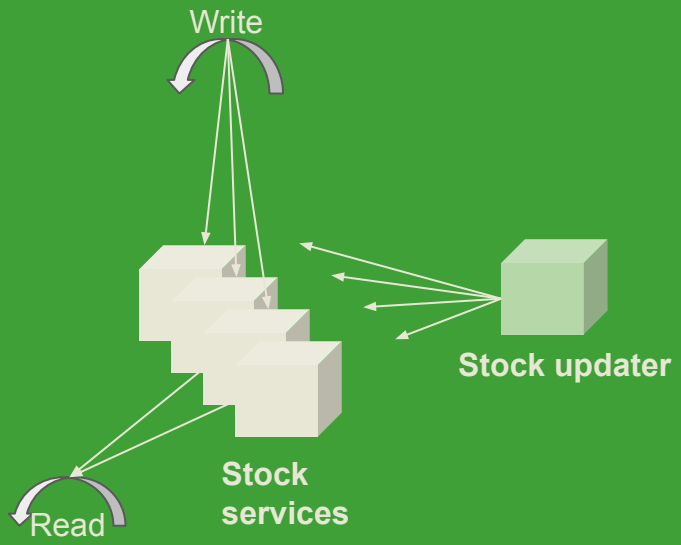


How do we shard?



Order services

Replication strategy stock service



Discussion

Coordinator is the single point of failure

Persistent volume for **sharding**

Empirically determine the value of **quorums**