Web-scale Data Management

Group 2 - Pragmatic Project

Technologies used
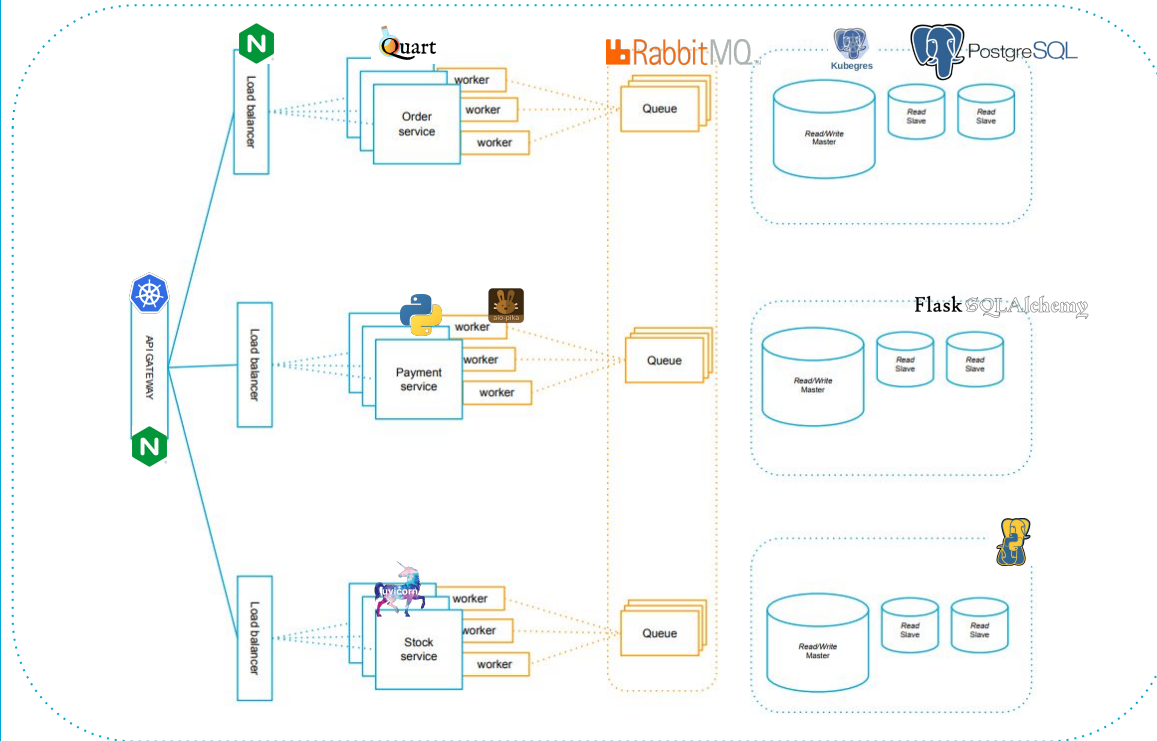
Architecture Diagram

Messaging
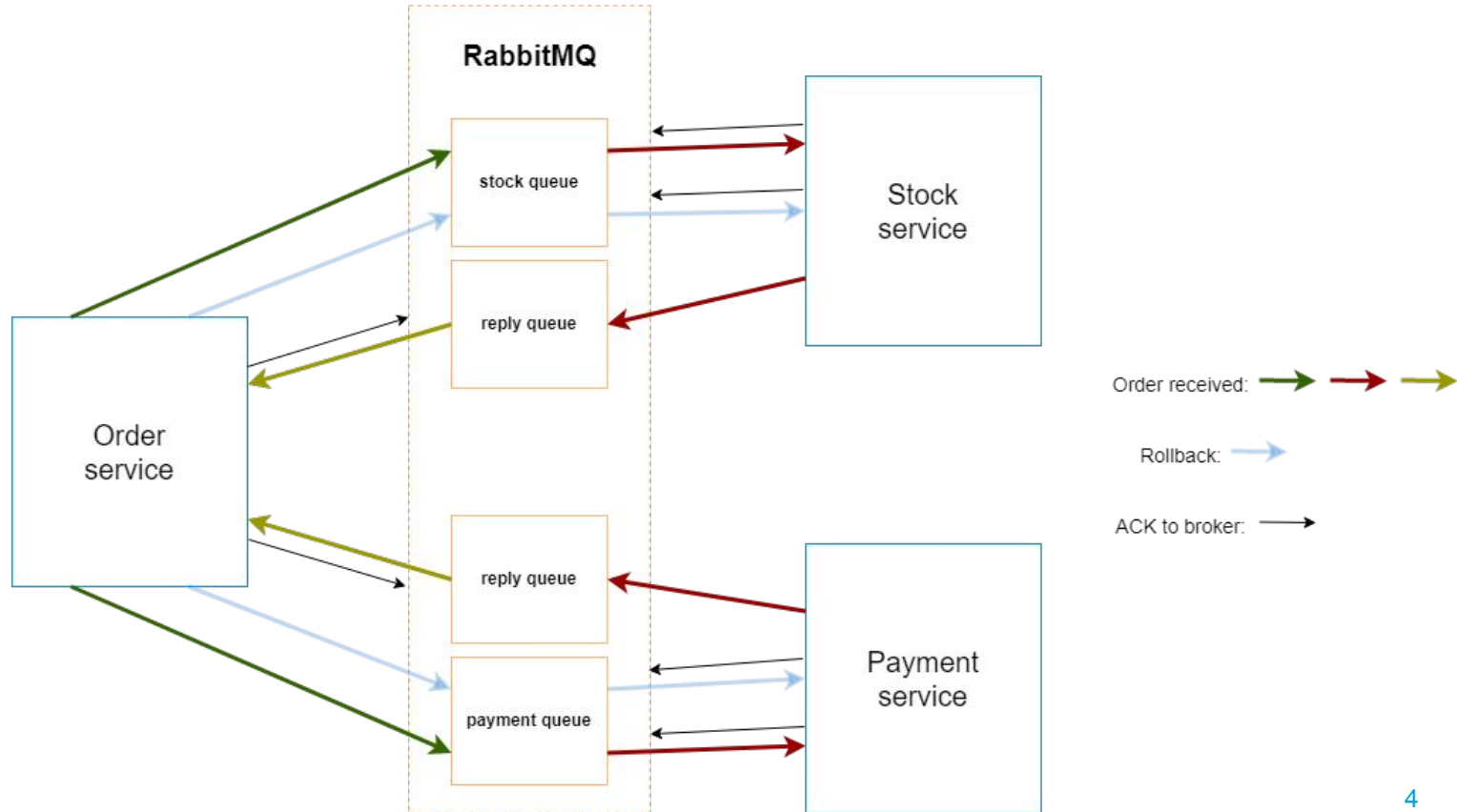
Orchestration based SAGA

Method of transaction execution

/orders/checkout/{order_id}

| Order Service | Stock Service | Payment Service |
| | idempotent | idempotent |

subtract_Items()

remove_credit()

200

400

Compensating Transaction

increase_Items()
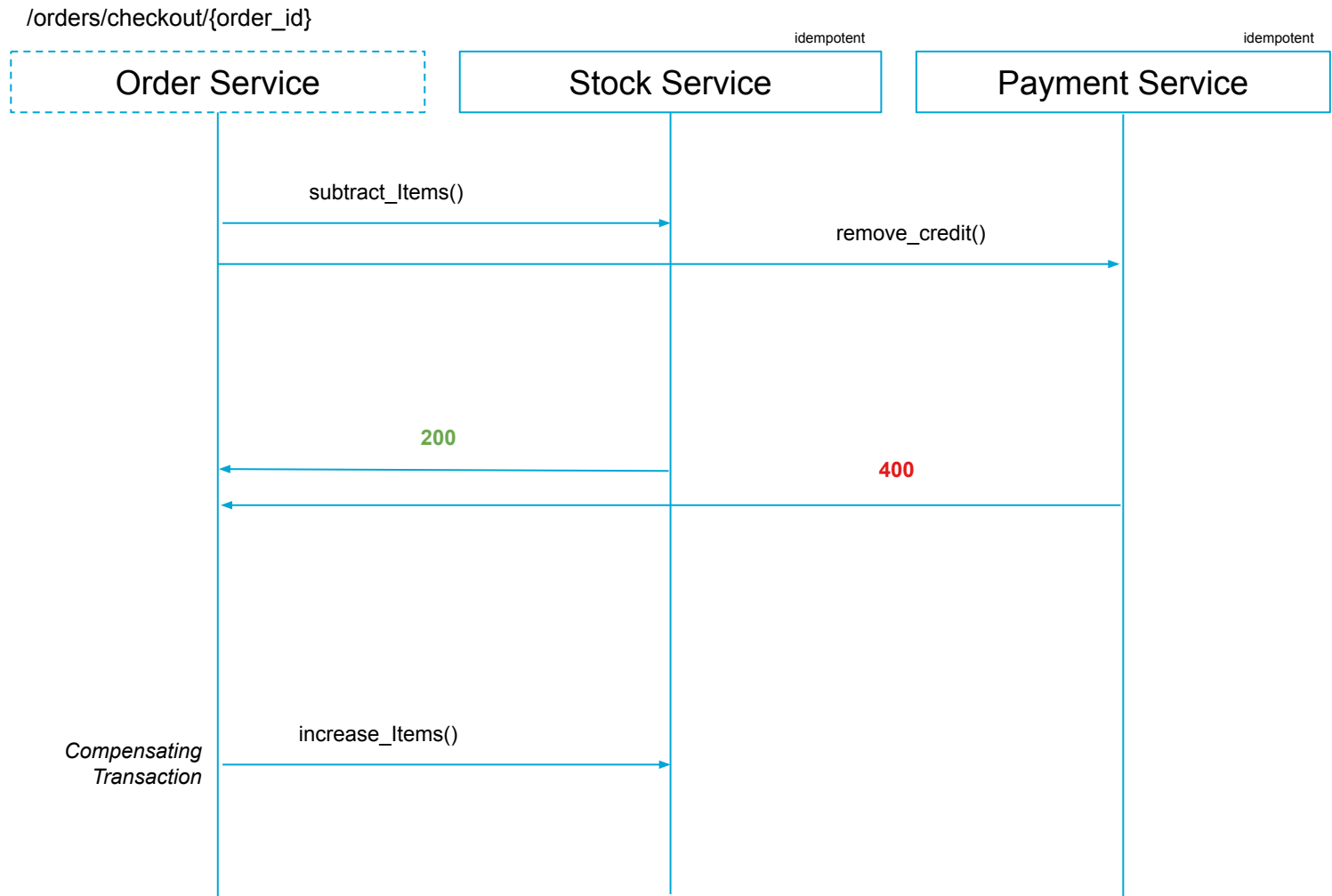
SAGA Guarantees

ACID

Atomicity: ✅

Consistency: ✅

Isolation: ❌

Durability: ✅

BASE

Base **A**vailability: ✅

Soft state: ✅

Eventual Consistency: ✅

Consistency

Eventual consistency

- Eventual consistency using SAGA [1], see SAGA guarantees
- BASE guarantees: Basic Availability, Soft State, Eventual Consistency [2]

PostgreSQL consistency

- Enforcing correct data using database constraints [3]
- "READ COMMITTED" transaction level [4]
  - Read sees snapshot of database, dirty read not possible
  - Write waits for concurrent transactions on target rows

Correctness of transactions

- Correctness through code checks
- Distributed Transactions possible through acknowledgements using RabbitMQ

Scalability

- Kubernetes Horizontal Pod Autoscaling

- Kubegres for PostgreSQL clusters + replicas

- Load balancing using Ingress

- Asynchronous messaging

- **No sharding and multiple masters**
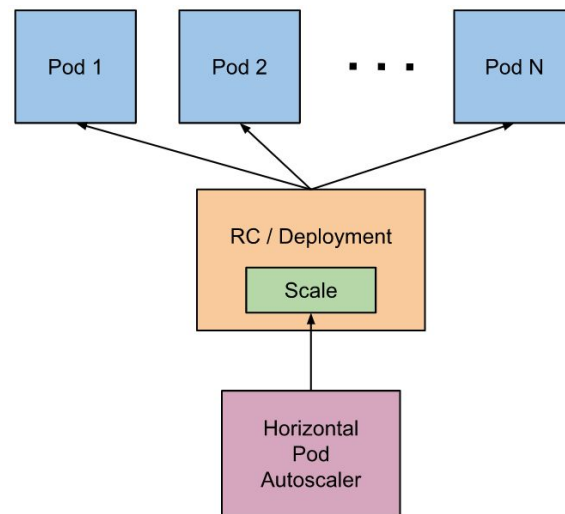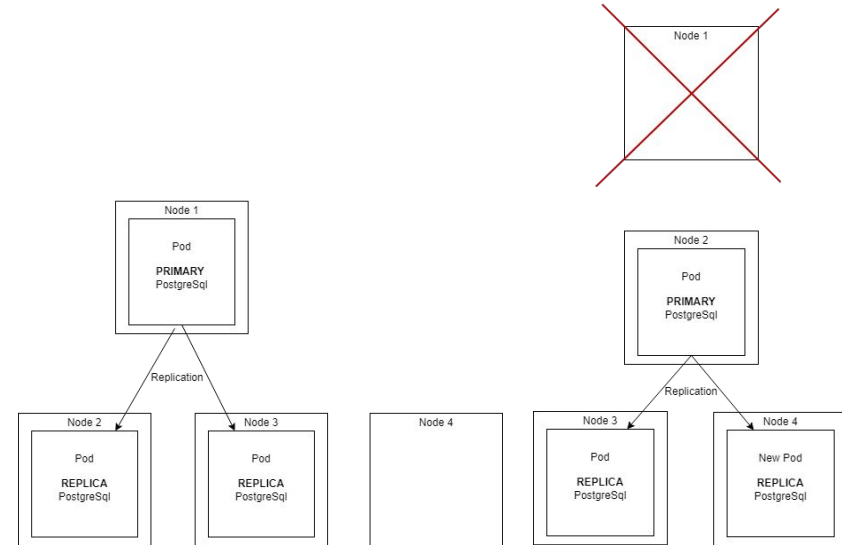
Fault Tolerance

Database Fault Tolerance
- Pod-anti affinity [5]
- Failover [5]

Microservice Fault Tolerance
- Stateless and replicated
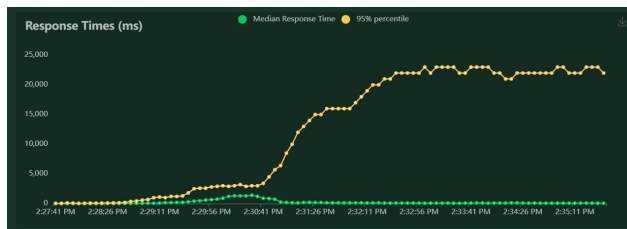- Self-healing [6]

Communication Fault Tolerance
- Acknowledgements [7]
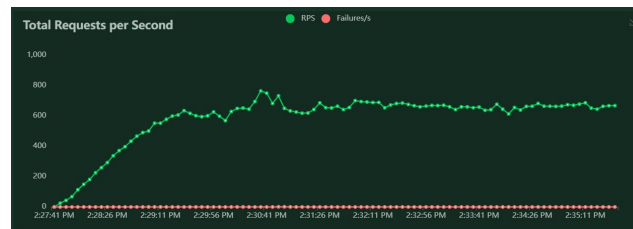- Quorum Queues [8]
- RabbitMQ Cluster [9]

Results

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
	Aggregated	282031	30	1391	3	25278	30	594.7	0.1

● **Latency**   ● **Throughput**

Limitations

- *Kubegres* does not allow sharding or multi-master replication. An example which does, is *BDR* [10].
- There is probably a bug somewhere, causing slow response times.

References

[1] https://medium.com/trendyol-tech/saga-pattern-briefly-5b6cf22dfabc

[2] https://www.scylladb.com/glossary/database-consistency/

[3] https://stackoverflow.com/questions/14225998/flask-sqlalchemy-column-constraint-for-positive-integer

[4] https://www.postgresql.org/docs/current/transaction-iso.html#XACT-READ-COMMITTED

[5] https://www.kubegres.io/doc/replication-and-failover.html

[6] https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

[7] https://www.rabbitmq.com/confirms.html

[8] https://www.rabbitmq.com/quorum-queues.html#usage

[9] https://www.rabbitmq.com/kubernetes/operator/operator-overview.html

[10] https://www.linkedin.com/pulse/multi-master-replication-relational-databases-scaling-ran-bechor/