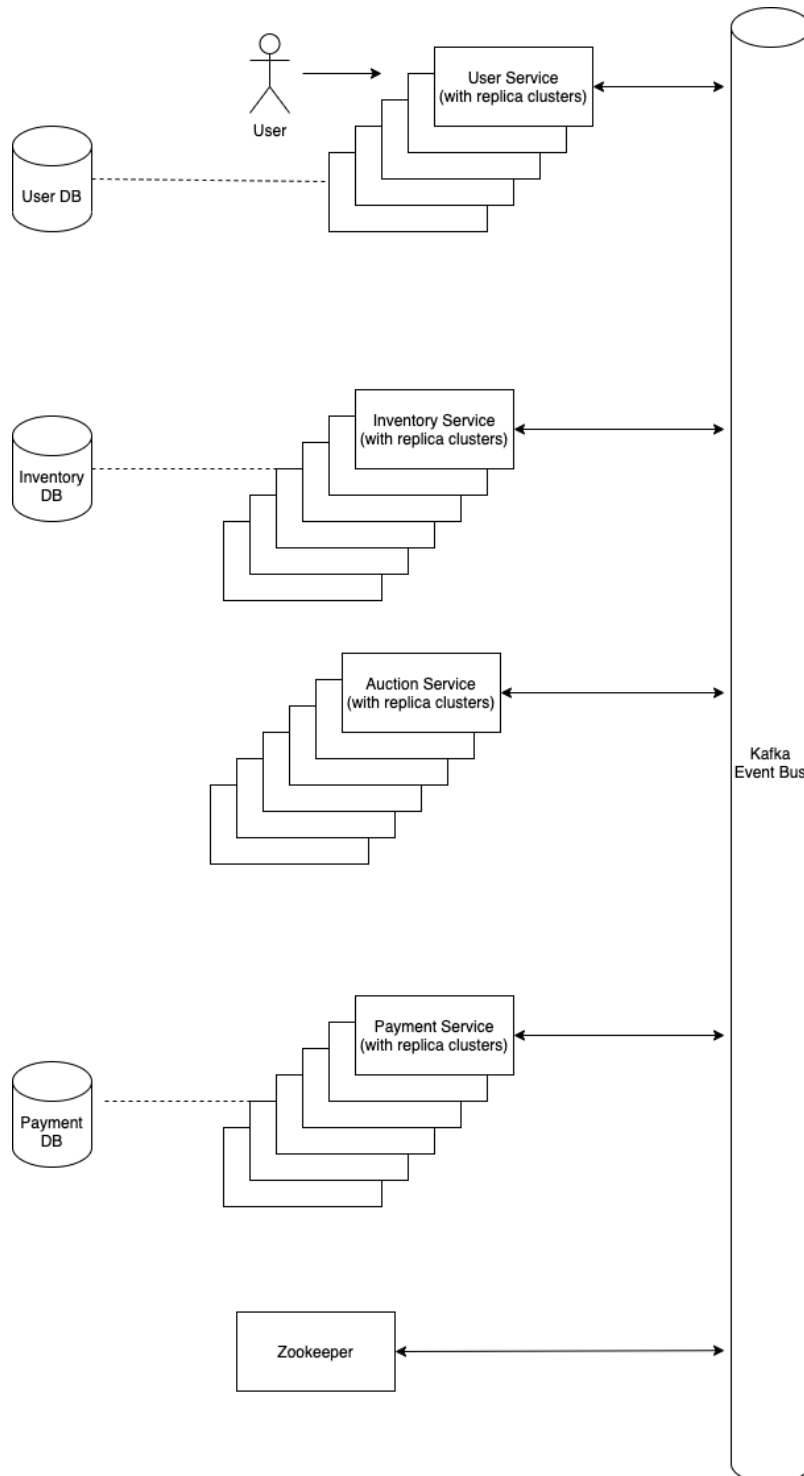


CENG 555 Analysis and Design of Microservice Based Systems

283678013 – Nazım Umut Ekici

262001027 – Oğuzhan Orhan

283078008 – Güliz Akkaya



- User Service
- Inventory Service
- Auction Service
- Payment Service

We separated the system into 4 micro-services. We implemented all of them. Additionally, a zookeeper service is implemented to provide discoverability and health checking for the services.

User Service: Responsible for registration, authentication and authorization of the user, when user attempts to login to system, users' information registered to User Database. Username, password, name and surname info required while registration. Responsible for information exchange between other services and user, like a gateway. (Implemented)

Inventory Service: Holds the information of the fishes and shares that information. Information saved into Inventory Database. (Implemented)

Auction Service: Holds the auction logic. (Implemented)

Payment Service: Holds the payment logic, controls payment method and validates checkout. Saves payment and receipt information into a Payment Database. (Implemented)

Zookeeper: Checks service health and responsible for service discovery.

Implemented Scenerios and Execution Flow Between Services:

1. User register
2. User Login
3. Fish record saving: User -> User service -> Inventory service (Inventory service saves fish info into Inventory DB)
4. List fish records: User -> User service -> Inventory service (gets the fish list from database and return that information to the user service) -> User service -> User
5. Start & Finish Auction: User -> User Service -> Auction Service
6. Make bid: User -> User Service -> Auction Service -> Inventory Service -> User Service -> User
7. Buy product: User -> User service -> inventory service -> auction service -> payment service (validates online payment method and records information into Payment Database, related info return back to user service) -> User Service -> User
8. Get payment records: User -> User service -> Payment service -> User service -> User

Bus Structure:

Kafka is used as an event bus. There are 6 topics on the bus:

- user (6 partitions)
- inventory (6 partitions)
- auction (6 partitions)
- payment (6 partitions)
- user_response (1 partition)
- zookeeper (1 partition)

Each topic **to** services has 6 partitions. Multiple partitions enable load balancing between service instances (6 is chosen rather randomly). user_response topic carries request responses to User services. zookeeper topic relays service statuses to Zookeeper service.