A Bare Metal Marketplace

Sprint 3 Presentation

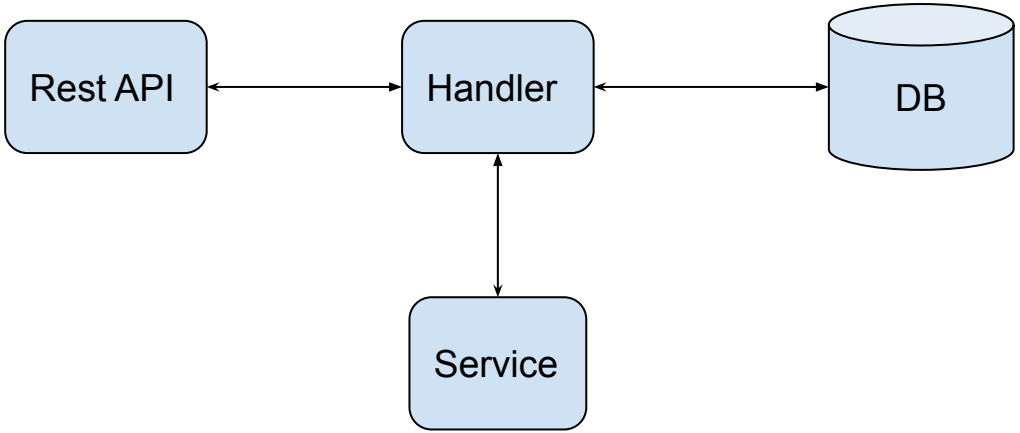
Ayush Upneja | Manan Monga | Haoxuan Jia | Parker Van Roy

Advisors: Sahil Tikale | Jonathan Chamberlain

Work Completed

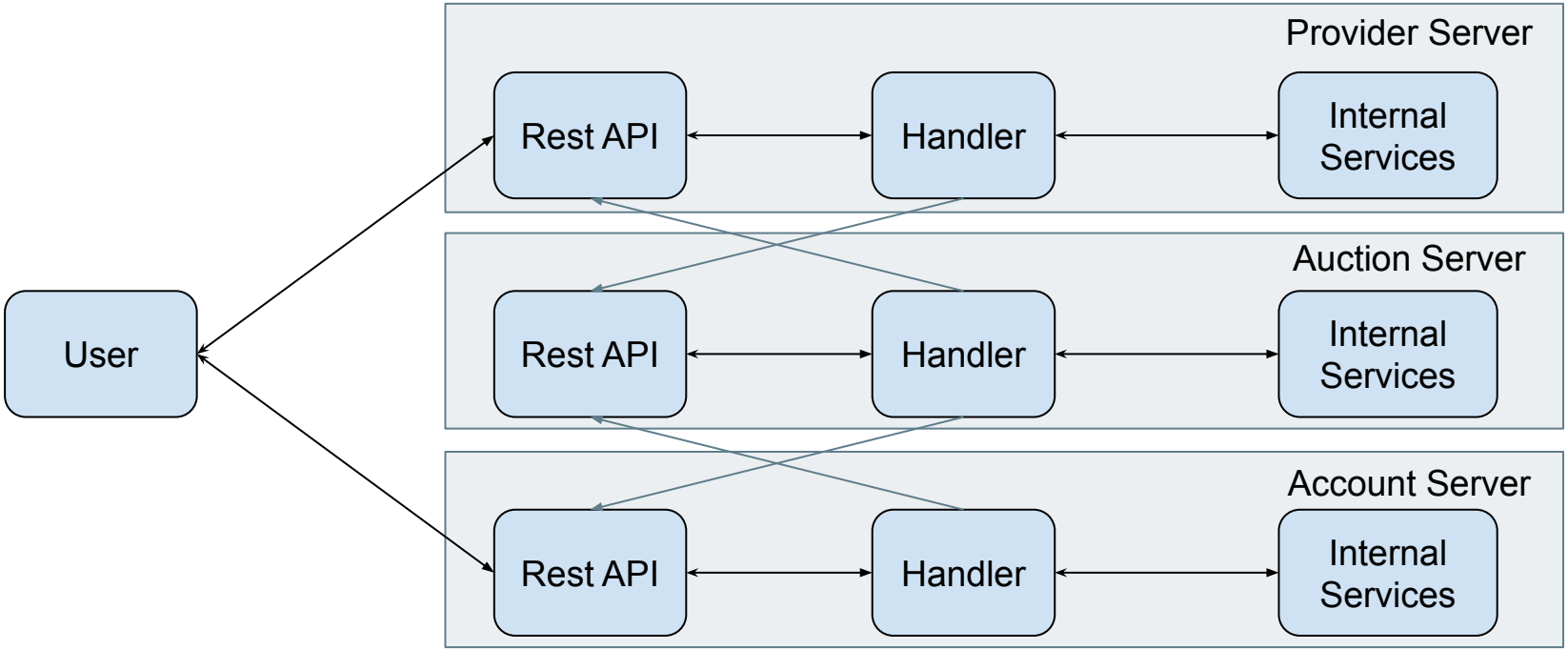
- Designed micro service structure
- Wrote queries using SQLAlchemy
- Wrote Flask Rest APIs
- Connect handler to APIs

Auction MicroService Structure



Developed under Component Abstraction Model

Service Communication Architecture



Auction Engine

- Double blind- second price auction
- Limit number of bids per user to 1
- Highest bid wins but the price is that of second highest bid, if no second bid then reserve price

How bids get matched to offers

- 4 basic functions in the engine
- List of bids created from all bids
- List sorted to get the highest of similar bids
- Highest bid gets matched with offer
- Matched (offer,bid) pair sent to database

SQLAlchemy

- How we are connecting to db and doing queries
- Object relational mapping
- Use query API to interact with tables

Marketplace Service API

Offer

- GET show me offers from everyone
- POST Create a new offer
- GET Show me details of a particular offer.
- DEL Deletes the offer.
- PUT updates only the <Status> field of the offer

Contract

- GET Display data about a particular contract.

contract

- GET List all contracts
- DEL Deletes the particular Contract.
- POST Create a new contract
- PUT updates only the <Status> field of the contract.

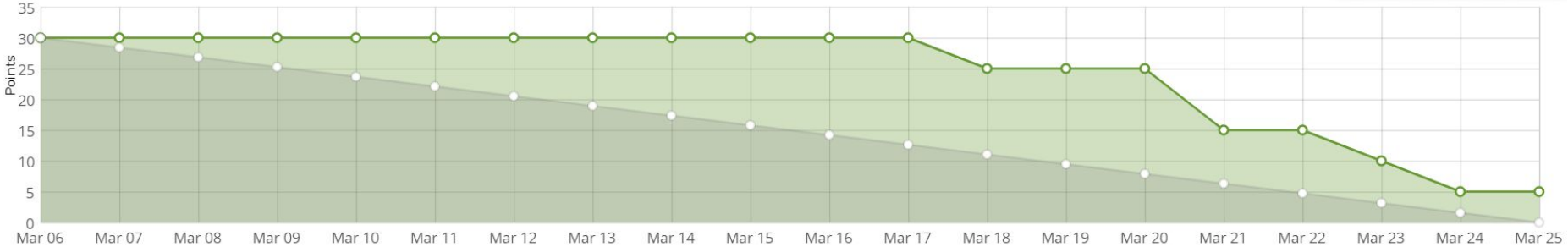
Bid

- POST Create a new bid
- GET Display data about a particular bid.
- GET List all bids
- DEL Deletes the particular bid.

Next Step

- Finish Integration of auction microservice
- Deploy auction microservice
- Implement account and provider microservice

Sprint 3 Burndown Chart



Thank you! Questions?