



A Bare Metal Marketplace

Sprint 2 Presentation


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

Advisors: Sahil Tikale | Jonathan Chamberlain





Project Description(Review)

- ❏ Co-located data center tenants with bare metal machines can rent them out
 - ❏ Provides bare metal nodes to users who need more resources when they want them
 - ❏ Connect an auction system to existing marketplace platform
 - ❏ Match up bids and offers in a double-blind auction
- 

Motivation for Bare-Metal Provisioning

- ❏ Entity installing OS on virtual node has root access
- ❏ You get exact configuration of node you rent
- ❏ Can use custom OS
- ❏ Security issue solved with BOLTED

Co - located Data Centers

- ❑ Use of a separate data center as a shared facility
- ❑ Multiple institutions' data is stored and managed offsite by colocation provider
- ❑ Colocation provider is paid fees for service
- ❑ Machines are owned by institutions

Work Completed

- ❏ Learned about TDD, pytest, microservice
- ❏ Designed auction system structure
- ❏ Created user stories for auction system
- ❏ Designed database schema
- ❏ Set up database with SQLAlchemy

User Stories

Provider Service

As a buyer

I want to be able to bid on new systems before giving up current system

As a seller

I want to be able check average price by entering my configuration

Auction Service

As an operator

I want to match earliest offer first to incoming bids

I want seller to get most money out of offer i.e. match seller with highest bidder for his configuration

I want to be able to see if any offers have been dormant for a while

Account Service

As an admin

I want to create and delete users

I want to be able to take or grant credits

As a buyer and seller

I want to check existing credits and contracts

I want to be able to request for more credits

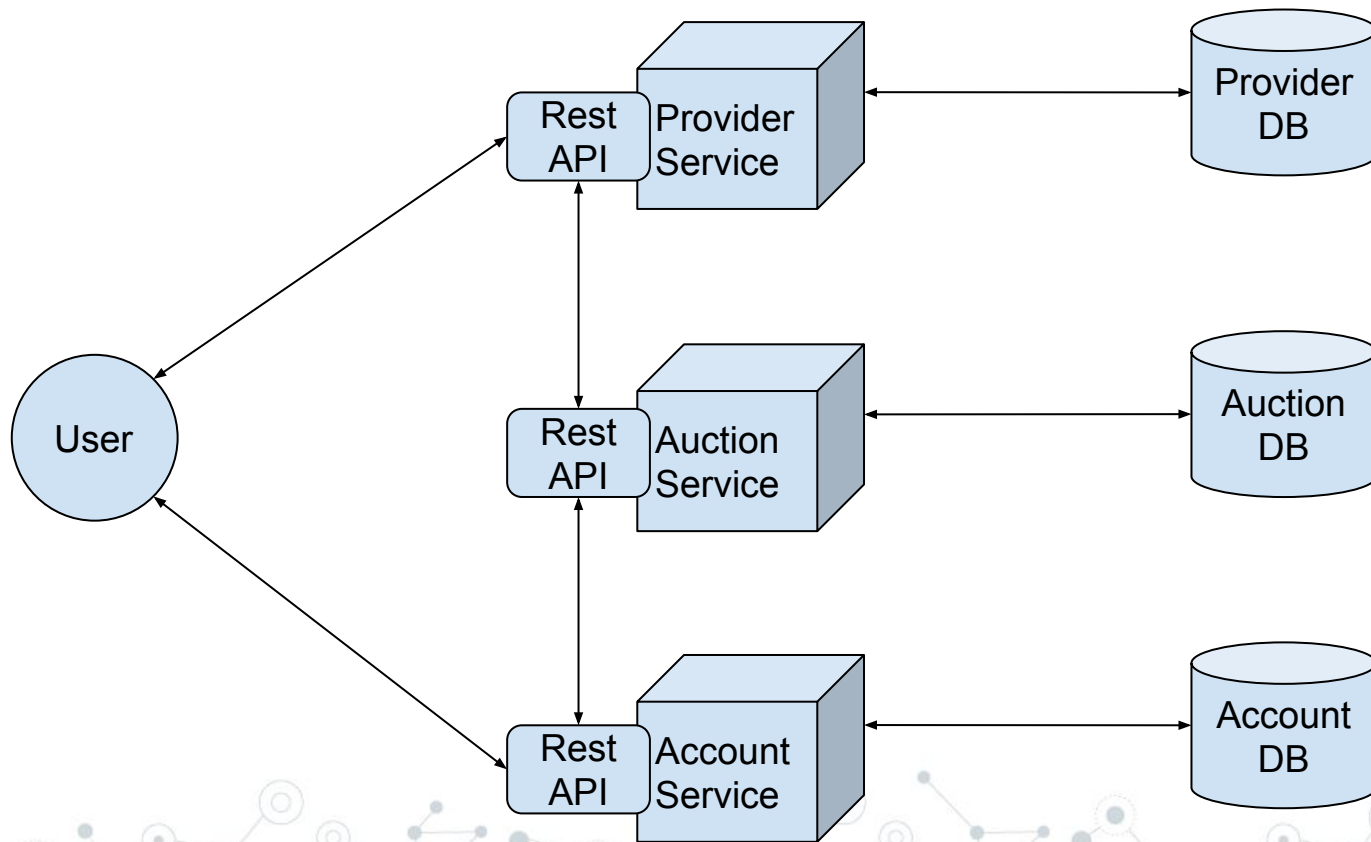
I want to be informed when a contract is made

As an operator

I want to take credits from buyer and give it to seller

I want to confirm contract after checking credits from buyer

Auction System Structure



Database Schema

Provider DB

Users

Bids

Offers

User-bid relation

User-offer relation

Auction DB

Bids

Offers

Contracts

Contract-bid-offer
relation

Account DB

Users

Contracts

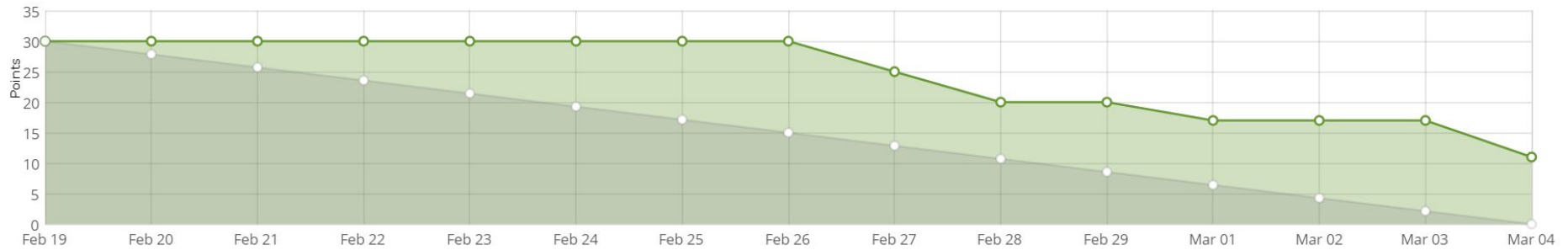
User-contract relation

How to maintain data consistency in microservices?

Next Step

- ❏ Write test cases using pytest
- ❏ Connect to databases
- ❏ Implement auction microservice

Sprint 2 Burndown Chart





Thank you! Questions?

