

International Institute Of Information Technology Bangalore

Project 2(E)

Decentralized Auction Engine Using Hyperledger-Fabric Blockchain Framework.

Submitted To:
Prof. Shrisha Rao

Submitted By:
Mayank Senani(MT2017067)
Mayank.Senani@iiitb.org
Sharad Mahajan(MT2017103)
Sharad.Mahajan@iiitb.org
Tarun Agrawal(MT2017127)
Tarun.Agrawal@iiitb.org

Project Goal:

To create an “Auction Engine” where buyers, sellers, and auctioneer are all distributed and use the **BlockChain Technology**.

Gap Analysis:

The limitations of current auctions :-

- 1) All the online auction platforms that currently exist are based on one centralized operation.
- 2) **They are not transparent-** Bidders have no way to ensure the origin, authenticity and legitimacy of a higher bid. Only the organizer has this information.
- 3) **They are not open** - Every bidder must use the organizer’s platform for registration, authentication and bidding. So, as there are a huge number of auctions organizers worldwide, a bidder must register many times over. They must therefore manage many different accounts and learn how to use a new and different interface each time.
- 4) **The ecosystem is very limited-** Because no one standard exists, each organizer has developed its own bidder interfaces and tools and even most auction organizers don’t have mobile applications.
- 5) In contrast with the live auction there will be high organizational cost and maintenance cost with limited number of bidders.

Therefore,

Our solution is a **new, open, decentralized** platform that is dedicated to real time Management of any auction, worldwide, whether online or live so that it is accessible to all.

In effect, everyone will be able to connect in order to sell, bid, organize an auction or offer a service to the ecosystem.

- **Scalable, reliable and transparent:** each operation will be registered within the network in a way that is transparent, publicly verifiable and infeasible to falsify.
- **Interoperable:** The bidder will be able to participate in numerous worldwide auctions– using just one interface.

Tools and Technologies:

- 1) Frontend:- HTML,CSS.
- 2) Backend:- Hyperledger Fabric (using Java/Node sdk),Go.
- 3) Amazon EC2 cloud services.
- 4) Github for repository management.

Architecture:

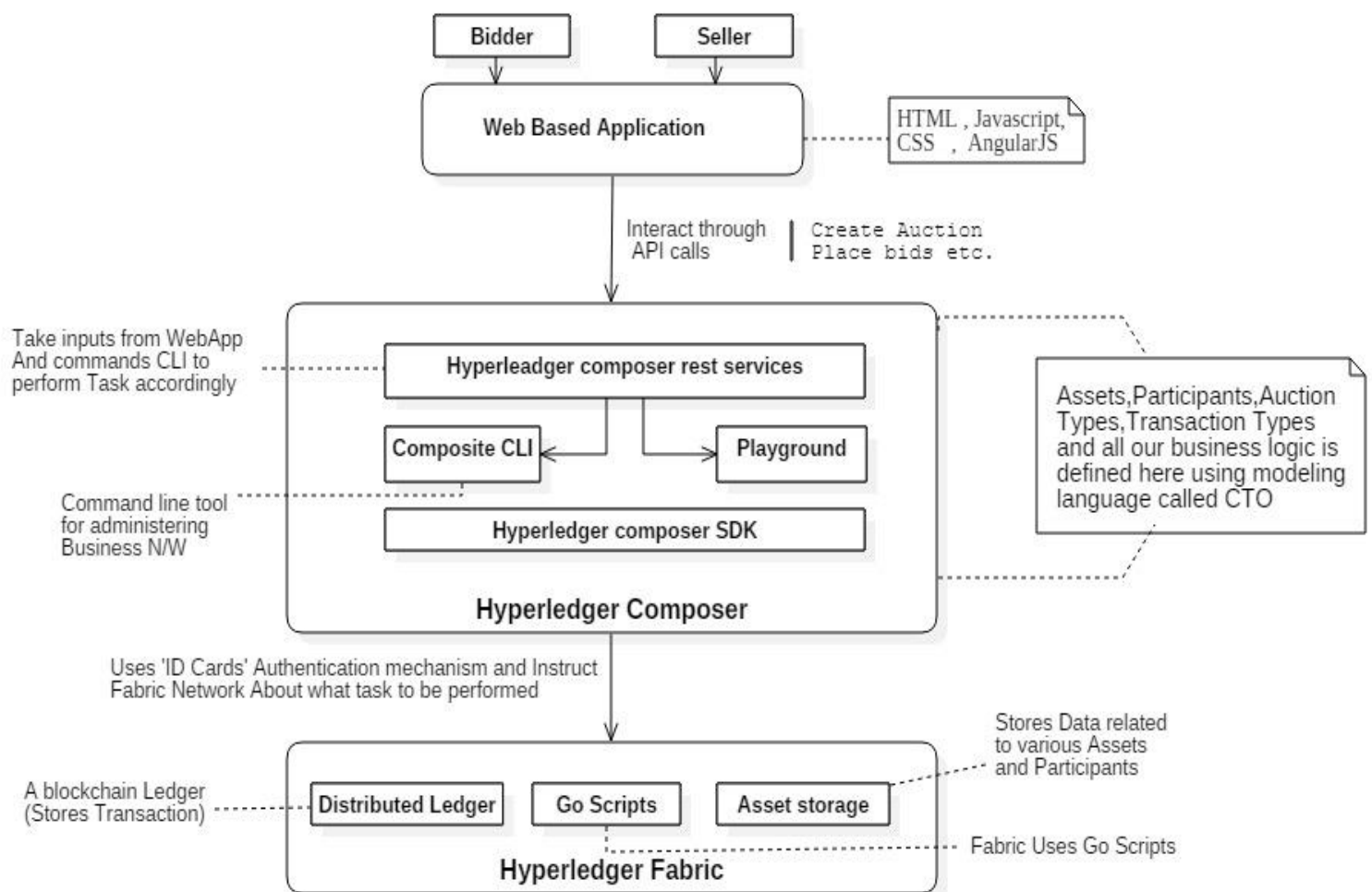


Fig. 1.1 Architecture Of Auction Engine

Use Case:

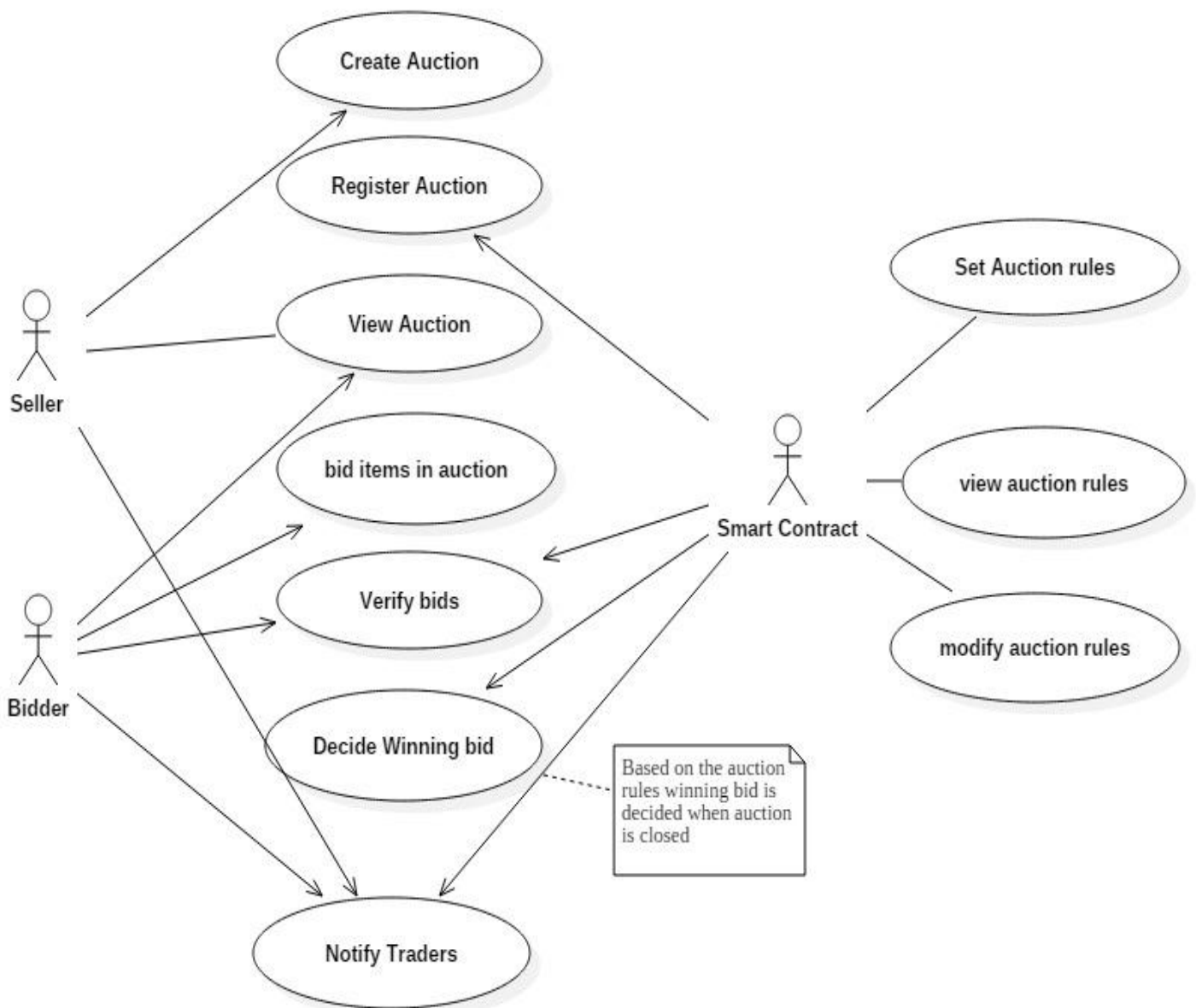


Fig 1.2 Use Case Diagram of Auction Engine

Test Cases:-

Generic:-

- 1-> Auction Type Can be (**English, Dutch, Kth Price, Double Auction, Reverse**).
- 2-> All Participants should belong to Hyper Ledger-Fabric Network.
- 3-> Ensure Money is “**Debited From**” /”**Credited To**” correct recipient.
- 4-> Ensure that participants who are out of Auction once are not allowed bid again.
- 5-> Auction Process should not start until all participants are Connected.
- 6-> After Auction Starts no new participant is allowed to join.
- 7-> Once placed, a bid can't be modified or removed.

For English Auction:-

- 1-> any new **bid** > **lastMaxBid**.
- 2-> Ensure that each participant must have deposited initial deposit amount.
- 3-> if no higher bid arrives within 60 seconds or auction times up, then bidder with highest bid wins and Auction Closes.
- 4-> **Base Amount** should be fixed before start of Auction Process.
- 5-> If no bids received within 60 seconds after start of auction, then Auction is closed and item remains unsold.
- 6-> Selling Amount are equal to last bid placed by winner.

For Dutch Auction:-

- 1-> After every 60 seconds **newSellingPrice** = **currentSellingPrice** – (**x% of currentSellingPrice**).
- 2-> **Base Amount** should be fixed before start of Auction Process.
- 3-> If no one buys within 60 seconds when sell price reaches base amount, then auction closes and item remains unsold.
- 4-> Auction close as soon as items are sold out.
- 5-> First Participant, who is willing to buy at **currentSellingPrice** should be the winner of auction.

For Reverse Auction:-

- 1-> No asks should be accepted after Time Over.
- 2-> **newAsk** <= **currentMinAsk** – (**x % of currentMinAsk**).
- 3-> if no lower Ask arrives within 60 seconds or auction times up, then seller with lowest ask wins and Auction Closes.
- 4-> **Base Amount** should be fixed before start of Auction Process.
- 5-> If no asks received within 60 seconds after start of auction, then Auction is closed and item remains unsold.
- 6-> Settlement Amount is equal to last ask placed by winner.

For Kth Price Auction:-

- 1-> any new **bid** > **lastMaxBid**.
- 2->Ensure that each participant must have deposited initial deposit amount.
- 3->if no higher bid arrives within 60 seconds or auction times up, then bidder with highest bid wins and Auction Closes.
- 4->**Base amount** should be fixed before start of Auction Process.th
- 5->if no one bids within 60 seconds after start of auction, then auction is closed and item remains unsold.
- 6-> Selling Amount is equal to last **Kth bid** of all the bids placed.
- 7->If items is sold and number of bids are less than k, then selling amount is equal to smallest bid placed.

For Double Auction:-

- 1->order the bidder and sellers based on their bid/ask and establish equilibrium between the bids and asks.
- 2->If **asksValues** > **bidValues** then no trade occurs (the seller wants more than the buyer pays)
- 3->seller puts their asks call and bidder put their bid call in a defined time interval.
- 4->any bids/asks that doesn't satisfy the equilibrium price is dropped.

Roadmap:

| WORK | Date | Progress |
|--|---------------|----------|
| Analyze Project Requirement and study basics of Blockchain | 10th Jan 2018 | |
| Create Architecture and Analysis of Test Cases | 24th Jan 2018 | |
| Environment Setup and Build Basic Fabric Network | 07th Feb 2018 | |
| Using Hyperledger Composer , Build transaction flow | 21st Feb 2018 | |
| Integrate our Business Model with existing network | 07th Mar 2018 | |
| Demo of Auction Engine (English and Dutch) | 21st Mar 2018 | |
| Testing of Current Build Of Auction Engine | 04th Apr 2018 | |
| Webapp Creation and Integration Testing | 14th Apr 2018 | |
| Final Submission | 25th Apr 2018 | |

