



Alexandria University

Faculty of Engineering

Computer and Systems Engineering Department

Graduation project submitted in partial fulfillment of the B.Sc.

Degree

**July 2025**

---

## **BitAuction - A Blockchain-based Decentralized Auction System**

### **Authors**

Amr Ahmed

Fareeda Abouzeid

Joseph Shokry

Michael Monir

Mohamed Arous

Omar Tammam

### **Supervisors**

Prof. Dr. Mohamed S. Abougabal

Prof. Dr. Shaimaa Lazem

Dr. Amira Alshazly

## **Abstract**

In recent years, blockchain technology has emerged and brought decentralization, transparency, and immutability to online applications. Auctions have become an important market process, finding wide adoption in many industries due to their efficiency in promoting fair trade. Open-outcry auctions include live and verbal bidding in very dynamic public settings. However, these auctions face problems such as limited accessibility and dependence on centralized parties. Additionally, the maintenance of bid order and timing in a centralized auction system requires belief in some governing body-a requirement that can be avoided by blockchains.

This research endeavor explores the idea of a blockchain-based auction system with the intention to include an open-outcry bidding mechanism. The solution uses blockchain technology to solve scalability challenges evident in traditional decentralized systems while also guaranteeing trust among participants in a manner that is transparent and secure. This distributed, immutable framework overcomes such issues by enabling verifiable, tamper-proof, and transparent bidding processes. This system removes intermediaries by deploying smart contracts for automating key functions that will be necessary in bid validation, determination of winners, and payment settlement. Additionally, it incorporates a distributed mechanism for achieving total ordering of bids, ensuring accurate bid timing and synchronization, which are critical for maintaining fairness in an open-outcry auction.

In order to solve the challenge of scalability, consensus mechanisms will be introduced that allow the system to handle high transaction throughput while supporting global participation. The precision of timing in bids is ensured by trusted timestamps obtained from external Network Time Protocol (NTP) services. This project adds distributed nodes to the timestamps system, allowing for distributed retrieval and verification of trusted timestamps.

Depending on our progress during the second semester, we also plan to explore the implementation of a sealed-bid auction mechanism. We plan to minimize transaction costs during the bidding phase using off-chain solutions.

## Acknowledgment

First and foremost, we want to thank Allah for his ceaseless blessings and constant presence, without which we would not have succeeded in this scientific endeavor.

We sincerely appreciate the inspiration, support, and encouragement we received from our mentors, **Prof. Dr. Mohamed S. Abougabal**, **Prof. Dr. Shaimaa Lazem**, and **Dr. Amira AlShazly**, during this project. We thank them for their constant guidance, and constructive criticism, and for introducing us to scientific methodologies for producing good-quality work.

We would like to express our sincere gratitude to all the professors who have played a role in shaping us into who we are today, especially **Dr. Ahmed Kosba**. His unwavering dedication to helping us improve and his deep instruction in scientific and engineering thinking have been invaluable. He worked tirelessly for our success, and we would not be where we are without his guidance and support.

Finally, we would like to extend our heartfelt thanks to our friends and families. We are deeply grateful for their unwavering support and for being with us every step of the way.

## **Declaration**

We declare that no part of the work referred to in this report has been submitted in support of an application for another degree or qualification of this or any other University or Institution of learning.

Amr Ahmed: .....

Fareeda Abouzeid: .....

Joseph Shokry: .....

Michael Monir: .....

Mohamed Arous: .....

Omar Tammam: .....

# Table of Contents

<b>Abstract.....</b>	<b>I</b>
<b>Acknowledgment.....</b>	<b>II</b>
<b>Declaration.....</b>	<b>III</b>
<b>Table of Contents.....</b>	<b>IV</b>
<b>List of Acronyms.....</b>	<b>V</b>
<b>List of Figures.....</b>	<b>VI</b>
<b>List of Tables.....</b>	<b>VII</b>
<b>Chapter 1</b>	
<b>Introduction.....</b>	<b>1</b>
1.1 Introduction.....	2
1.2 Motivation and Objectives.....	2
1.3 Contributions.....	3
1.4 Organization of the Report.....	3
1.5 Conclusion.....	4
<b>Chapter 2</b>	
<b>Background.....</b>	<b>5</b>
2.1 Introduction.....	6
2.2 Auction Background.....	6
2.2.1 Auction Systems.....	6
2.2.2 Types of Auctions.....	7
2.3 Blockchain Background.....	10
2.3.1 Blockchain Technology.....	10
2.3.2 Consensus Mechanism.....	12
2.3.3 Blockchain Architecture[1].....	15
2.3.4 Cryptographic Primitives in Blockchains.....	17
2.3.5 Types of Blockchains.....	19
2.3.6 Blockchains impact on different fields.....	21
2.3.7 Smart Contracts.....	23
2.4 Conclusion.....	25
<b>Chapter 3</b>	
<b>Related Work.....</b>	<b>26</b>
3.1 Introduction.....	28
3.2 Related Work.....	29
3.2.1 Related Work (Published papers).....	29
3.2.2 Related Work (Applications).....	31
3.3 Need to Extend Related Work.....	33
3.3.1 Open-Source Development - (3.2.2.1).....	33
3.3.2 Development using the SPI model (3.2.1.3).....	33
3.3.3 Support for Open-Outcry auction (3.2.1.4).....	33

3.3.4 Integrate with a decentralized system (3.2.1.6, 3.2.1.12).....	33
3.3.5 Auction privacy protection (3.2.1.13).....	34
3.3.6 Transaction ordering and synchronization (3.2.1.15).....	34
3.3.7 Smart contract update (3.2.1.16).....	34
3.3.8 Auction payment with cryptocurrency (3.2.1.17).....	34
3.3.9 Auction scalability solutions (3.2.1.8).....	35
3.3.10 Multi chain compatibility.....	35
3.3.11 Off-Chain data handling (3.2.1.9).....	36
3.3.12 Integration with oracles for real-world data.....	36
3.3.13 Conduct evaluation for the system.....	36
3.3.14 Use FastFabric as an underlying system.....	36
3.3.15 Support dynamic live auction (3.2.2.15).....	37
3.3.16 Support personalized recommendations (3.2.1.19).....	37
3.3.17 Support price prediction on auction items (3.2.1.19).....	37
3.3.18 Develop auction user-friendly interface.....	37
3.3.19 Design and implement a sealed bid auction. (3.2.1.4).....	37
3.3.20 Optimize the bidding stage by applying off-chain operations. (3.2.1.9).....	38
3.3.21 Compare and test the effect of this added feature (3.2.1.18).....	38
3.4 Scope of Work.....	38
3.5 Future Work.....	39
3.5 Conclusion.....	40
<b>Chapter 4</b>	
<b>Software Development Process and Implementation.....</b>	<b>41</b>
4.1 Introduction.....	42
4.2 Software Process Improvement (SPI).....	42
4.2.1 What is SPI?.....	42
4.2.2 Project Management Process (PMP).....	43
4.2.3 Product Development Process (PDP).....	44
4.2.4 Peer Review Process (PRP).....	46
4.2.5 Configuration Management Process (CMP).....	46
4.2.5 Quality Assurance Process (QAP).....	47
4.3 Software Requirements Specifications (SRS).....	47
4.3.1 Scope of the Software.....	47
4.3.2 System Specifications.....	48
4.4 Software Design and Architecture.....	51
4.4.1 Work Overview.....	51
4.4.2 Hyperledger Network.....	53
Smart Contract Architecture.....	54
Data Storage and State Management.....	54
Supporting Components.....	54
Security Features.....	54
4.4.3 Database Design.....	56
4.4.4 Sequence Diagrams.....	57

4.5 Platforms and Tools.....	59
4.5.1 Front-end Tools.....	59
4.5.2 Back-end Tools.....	60
4.5.3 Blockchain tools and platforms.....	60
4.5.4 Version Control.....	61
4.6 Implementation.....	61
4.6.1 Chaincode.....	61
4.6.2 Features.....	62
4.7 Conclusion.....	65
<b>Chapter 5</b>	
<b>Results and Discussion.....</b>	<b>67</b>
5.1 Introduction.....	68
5.2 Results.....	68
5.2.1 Hyperledger Explorer[37].....	68
5.2.2 NTP servers Testing.....	69
5.2.3 Performance Testing.....	69
5.2.4 Results Discussion.....	71
5.2.5 Recommendations.....	72
5.3 conclusion.....	72
<b>Chapter 6</b>	
<b>Conclusion and future work.....</b>	<b>73</b>
6.1 Introduction.....	74
6.2 Project conclusion.....	74
6.3 Contribution of the project.....	74
6.4 Future Work.....	75
<b>References.....</b>	<b>76</b>
<b>Appendix A.....</b>	<b>79</b>
Summary of BitAuction Research Papers.....	79
A.1 Summary of [1].....	80
A.2 Summary of [2].....	84
A.3 Summary of [3].....	89
A.6 Summary of [6].....	93
A.7 Summary of [7].....	96
A.8 Summary of [8].....	99
A.9 Summary of [9].....	108
A.10 Summary of [10].....	112
A.11 Summary of [11].....	115
A.12 Summary of [12].....	119
A.18 Summary of [18].....	127
A.19 Summary of [19].....	131
<b>Appendix B.....</b>	<b>138</b>
B.1 Implementation Plan.....	139
<b>Appendix C.....</b>	<b>141</b>

Sealed-bid optimization.....	141
Explanation of Sealed-bid Auction Optimization.....	142
C.1 Offchain Protocol.....	142
C.2 Auction onchain protocol.....	143
<b>Appendix D.....</b>	<b>146</b>
UI samples of BitAuction.....	146



## List of Acronyms

Meaning	Term/Symbol	Page
Asymmetric Encryption	AE	<a href="#">18</a>
Commitment Schemes	CS	<a href="#">18</a>
Decentralized Finance	DeFi	<a href="#">25</a>
Delegated Proof of Stake	DPOS	<a href="#">11</a>
Distributed Hash Table	DHT	<a href="#">16</a>
Elliptic Curve Digital Signature Algorithm	ECDSA	<a href="#">16</a>
First-Price Sealed-Bid	FPSB	<a href="#">8</a>
Hyperledger Fabric	HLF	<a href="#">15</a>
InterPlanetary File System	IPFS	<a href="#">34</a>
Network Time Protocol	NTP	<a href="#">I</a>
Peer-to-peer	P2P	<a href="#">19</a>
Practical Byzantine Fault Tolerance	PBFS	<a href="#">14</a>
Proof of Authority	POA	<a href="#">16</a>
Proof of Stake	POS	<a href="#">11</a>
Proof of Work	POW	<a href="#">11</a>
Secure Multiparty Computation	MPC	<a href="#">17</a>
Software Process Improvement	SPI	<a href="#">32</a>
Vickrey-Clarke-Groves	VCG	<a href="#">10</a>
Zero-Knowledge Proofs	ZKP	<a href="#">18</a>

## List of Figures

<i>Figure 2.1 Dimensions for Classifying Auction Models</i> .....	7
<i>Figure 2.2 Blockchain Architecture layers</i> .....	15
<i>Figure 3.1 The blockchain trilemma</i> .....	35
<i>Figure 4.1: Work Overview</i> .....	51
<i>Figure 4.2: Hyperledger Fabric Network</i> .....	54
<i>Figure 4.3: Database ERD</i> .....	55
<i>Figure 4.4: Relational schema</i> .....	56
<i>Figure 4.5: Sequence Diagram 1</i> .....	57
<i>Figure 4.6: Sequence diagram 2</i> .....	58
<i>Figure 4.7: Auction contract flow</i> .....	61
<i>Figure 4.8: Hyperledger Fabric Explorer</i> .....	65

**List of Tables**

*Table 2.1 Comparison of Consensus Mechanisms* .....14

*Table 3.1 Literature Review*.....28

*Table 3.2 Applications Review*.....30

Table 5.2.2.1 Results of baseline tests .....70

Table 5.2.2.1 Results of baseline tests .....70

Table 5.2.2.1 Results of baseline tests .....70

*Table B.1 Implementation Plan*.....103

# **Chapter 1**

## **Introduction**

## 1.1 Introduction

Auction systems have seen remarkable evolution, progressing from traditional in-person events to online platforms and, more recently, to blockchain-based systems.<sup>[1]</sup> Traditional auctions often required participants to gather at a specific physical location, which limited accessibility and incurred high operational costs.<sup>[2]</sup> Although online auctions expanded participation and reduced costs, they brought new challenges, including centralization, lack of transparency, cybersecurity risks, and high transaction fees.<sup>[2]</sup> In this chapter, an overview of the motivation underlying the proposed work is presented and our contributions to addressing the identified challenges are outlined.

## 1.2 Motivation and Objectives

The motivation for this project lies in addressing the limitations and vulnerabilities of existing auction systems, whether traditional or online. Traditional auctions face constraints such as limited accessibility, high operational costs, and dependence on centralized authorities, leading to concerns about trust and fairness.<sup>[1]</sup> While online auction systems improve accessibility, they remain prone to issues like data manipulation, fraud, and centralization risks.<sup>[1]</sup>

A blockchain-based auction system presents a compelling alternative to these persistent issues by leveraging decentralization, transparency, and enhanced security.<sup>[1]</sup> Blockchain's immutable ledger ensures that all bids are visible and verifiable, fostering trust among participants.<sup>[1]</sup> Additionally, smart contracts automate auction processes, reducing the risks of fraud and unauthorized changes.<sup>[1]</sup> However, scalability remains a significant challenge in blockchain-based systems due to the blockchain trilemma: the trade-off between scalability, security, and decentralization.<sup>[2]</sup>

The open-outcry auction format, in particular, is underexplored in the current literature due to its inherent complexity and synchronization requirements<sup>[3]</sup>, leaving a gap in practical implementations. Scalability has consistently been an issue for both traditional and blockchain-based auctions, limiting their ability to handle high transaction volumes efficiently.<sup>[3]</sup>

## 1.3 Contributions

This project addresses these challenges by proposing a scalable blockchain-based auction platform that leverages the strengths of blockchain technology to resolve issues of trust, transparency, and performance.

Key contributions of this project include such as.

The first five points represent fundamental features present in any blockchain-based auction system, offering core benefits like transparency, security, cost efficiency, global accessibility, and fraud resistance—ensuring trust and fairness through verifiable, decentralized mechanisms.

- **Decentralization and Transparency:** ensuring all bids are visible and verifiable through an immutable ledger to foster trust among participants.
- **Enhanced Security:** employing smart contracts to automate auction processes and reduce the risks of fraud and unauthorized changes.
- **Cost Reduction:** minimizing transaction fees by eliminating intermediaries, making the system more cost-effective.
- **Global Accessibility:** allowing participants to securely join auctions from anywhere in the world, independent of centralized entities.
- **Fraud Prevention:** leveraging immutable records and automated rule enforcement to reduce fraudulent activities, such as fake bids or manipulated outcomes.
- **Scalability Improvements:** exploring innovative methods to overcome scalability challenges and support open-outcry bids, filling the gap in existing literature and practical implementations.

Through these contributions, the project aims to design a robust auction platform that successfully balances scalability, security, and decentralization while supporting the complex requirements of open-outcry auction formats.

## 1.4 Organization of the Report

The report is organized to reflect the progressive development of the BitAuction project. Background information on relevant concepts is presented in Chapter Two.

Section 2.2 outlines auction systems, detailing their components, classification methods, and limitations. Section 2.3 shifts to blockchain technology, discussing its architecture, consensus mechanisms, cryptographic tools, and applications, with a focus on smart contracts.

Section 3.2 reviews related work, summarizing prior research and applications while identifying gaps, such as scalability and privacy challenges. Section 3.3 highlights how this project addresses these gaps with innovations like open-cry auctions and enhanced privacy. Section 3.4 defines the project scope, and Section 3.6 concludes by summarizing the chapter and linking to the implementation details in subsequent sections.

## **1.5 Conclusion**

This chapter has provided a comprehensive overview of the motivation, objectives, and contributions of the proposed blockchain-based auction platform, addressing the limitations of traditional and online auction systems. By leveraging the decentralized and transparent nature of blockchain, the project aims to overcome persistent issues such as trust deficits, high costs, and security vulnerabilities. Key contributions include decentralization, enhanced security, fraud prevention, and scalability improvements, particularly in the context of open-outcry auctions.

Through these innovations, the project seeks to fill gaps in existing literature and practical implementations, creating a robust, cost-effective, and globally accessible auction platform. The report is structured to guide the reader through the conceptual foundations, related work, and the technical implementation, beginning with a detailed background in the next chapter.

## **Chapter 2**

### **Background**



## **2.1 Introduction**

A brief introduction to BitAuction and the evolution of auction systems was provided in the previous chapter. This chapter delves deeper into the essential background of auction systems and blockchain technology.

Section 2.2 offers a comprehensive comparison of the different types of auction systems across multiple dimensions, such as the bidding process, the number of items involved, and the quantity of auctioned units.

Section 2.3 provides foundational knowledge about blockchain technology, covering its core principles, key features, and associated challenges. It also explores the different types of consensus mechanisms, blockchain architecture, various blockchain classifications, the concept of smart contracts, and the necessary cryptographic primitives that underpin blockchain systems.

## **2.2 Auction Background**

### **2.2.1 Auction Systems**

An auction is a process of buying and selling goods or services. This process involves offering items or goods for bidding, waiting for bids to be accepted, and then selling goods to the highest bidder under the supervision of an auctioneer.<sup>[2]</sup>

Normal auctions consist of four elements such as.<sup>[1]</sup>

- Seller: who owns and wants to sell the objects.
- Two or several bidders: who want to buy the objects via the auction.
- Auction Object: object traded between the seller and the buyer(s).
- Auctioneer: an intermediary agent who hosts and controls the auction process.

## 2.2.2 Types of Auctions



Figure 2.1 Dimensions for Classifying Auction Models

Auction models can be classified from different dimensions such as.<sup>[1]</sup>

1. Bidding process
2. Number of items
3. Roles of buyers/sellers
4. Bidding participants

### 2.2.2.1 Classification of Auction Models by Bidding Process<sup>[1]</sup>

There are 2 main categories for auction models depending on the bidding process

- **Open-Outcry:** a bidder's activities are transparent and visible to all bidders.
- **Sealed-Bid:** bidders submit their bids to the auctioneer privately, and the bids are only known by the auctioneer until the auction ends.

Typical open-outcry auctions and sealed-bid auctions are summarized as follows.

- **English Auction** (also called open-outcry ascending-price auction). In an English auction, the price begins low and rises as buyers submit their bids until no bidder offers a higher bid. The auctioneer then declares the end of the auction, and the winner is the highest bidder, who is required to purchase the item at their winning bid. This

type of auction requires all bids to be transparent and allows bidders to place multiple bids.<sup>[1]</sup>

- **Dutch Auction** (also called open-outcry descending-price auction or clock auction). In a Dutch auction, the auctioneer begins by announcing a high asking price and then progressively lowers it until a buyer is willing to accept the current price.<sup>[1]</sup>
- **First-Price Sealed-Bid (FPSB) Auction** (also called blind auction). In an FPSB auction, all bidders submit sealed bids to the auctioneer simultaneously, and the highest bidder wins and pays their bid. Other bidders' bids will not be revealed during the auction until a winner is determined. Therefore, bidders do not compete openly with each other.<sup>[1]</sup>
- **Vickrey Auction** (also called second-price sealed-bid auction). It is similar to an FPSB auction but with a different payment mechanism. After all bidders submit sealed bids to the auctioneer, the highest bidder still wins but only pays the second-highest bid.<sup>[1]</sup>

#### 2.2.2.2 Classification of Auction Models by Number of Items<sup>[1]</sup>

Auction models can also be classified based on the number of items being auctioned, dividing them into single-item auctions and multi-item auctions. While the auction models discussed previously primarily handle the sale of a single item, there are cases where selling multiple items simultaneously is more efficient. Multi-item auctions are further categorized as follows.

- **Homogeneous Auctions:** all items being auctioned are identical.
- **Heterogeneous Auctions:** the items differ from one another.<sup>[1]</sup>

#### Combinatorial Auctions (Multi-Lot Auctions)

Combinatorial auctions are a form of multi-item auction that involves selling heterogeneous items together. Bidders can place bids on specific bundles or combinations of items rather than on individual items. This model is particularly useful when bidders assign non-additive values to combinations of items, meaning the value of a group of items exceeds the sum of their values. However, while combinatorial auctions allow bidders to express more complex preferences, they introduce significant challenges related to mechanism design and

computation. A key challenge is the winner determination problem, which is NP-hard and computationally intensive due to the vast number of possible item combinations.<sup>[1]</sup>

### **Multi-Unit Auctions**

Multi-unit auctions are used to sell multiple homogeneous items simultaneously. These auctions can be classified into two types based on the payment mechanism such as.

1. **Pay-as-Bid Auction** (Discriminatory Price Auction): winning bidders pay the exact amount of their respective bids for each unit they secure.
2. **Uniform Price Auction** (Clearing Price Auction): all winning bidders pay the same unit price, determined by the auction outcome, regardless of their individual bid amounts.

In multi-unit auctions, bidders may have an incentive to bid less than their true valuations to reduce costs, which can result in inefficient allocation of items.<sup>[1]</sup>

#### **2.2.2.3 Classification of Auction Models by the Roles of Buyers/Sellers**

Auction models can also be classified based on the roles of buyers and sellers into forward auctions and reverse auctions.

- **Forward Auction:** also known as a seller-determined auction, this is the conventional model where a single seller offers products to multiple potential buyers (bidders). All previously discussed auction models fall into this category.<sup>[1]</sup>
- **Reverse Auction:** in contrast to forward auctions, a reverse auction (also called a buyer-determined auction or procurement auction) reverses the roles of buyers and sellers. In this model, a single buyer seeks goods or services, and multiple sellers compete by placing bids to win the contract. The buyer typically specifies the requirements for the goods or services, and sellers submit their bids based on their offerings.<sup>[1]</sup>

Reverse auctions are particularly well-suited for procurement processes carried out by governments, corporations, and organizations, as they encourage competition among sellers to offer the most competitive price. However, a notable drawback of reverse auctions is that they do not require sellers to disclose the specific costs associated with their bids. As a result, buyers might select the lowest-priced bid only to encounter substandard products or poor service quality.

#### 2.2.2.4 Classifying auction models depending on the bidding participants

Auction models can also be categorized based on the participants involved in the bidding process into single-sided and double-sided auctions.

- **Single-Sided Auction:** this approach is common in traditional auctions, such as forward and reverse auctions, where either sellers or buyers participate as bidders. However, single-sided auctions may struggle to accommodate additional participants, particularly in large-scale settings.<sup>[1]</sup>
- **Double Auction:** also referred to as a double-sided auction, this model extends the conventional auction framework by allowing many-to-many interactions between participants. In a double auction, multiple sellers submit offers while multiple buyers place bids. The auctioneer, or market institution, determines a market-clearing price that matches supply with demand.<sup>[1]</sup>

Several market-clearing mechanisms have been developed for double auctions, including the following.

- The average mechanism,
- The Vickrey-Clarke-Groves mechanism (VCG),
- The trade reduction mechanism, and
- McAfee's mechanism.

Double auctions are particularly well-suited for marketplaces where multiple buyers and sellers interact, such as stock exchanges. However, managing double auctions can be challenging, especially when dealing with heterogeneous items that have multiple attributes. This complexity arises from the significant execution time and computational costs involved.

## 2.3 Blockchain Background

### 2.3.1 Blockchain Technology

Blockchain technology is a decentralized, distributed ledger system that securely records transactions across multiple nodes within a network. This innovative approach ensures data integrity, transparency, and security without the need for a central authority. Each transaction is organized into blocks, which are chronologically linked to form a "chain" of blocks, hence the term "blockchain".

Blockchain has revolutionized various industries, including finance, supply chain, healthcare, and more, by offering a trustless and immutable method for recording and verifying transactions.

## **How Blockchain Works**

### **1. Transaction Creation**

The process begins when a user initiates a transaction. This transaction is broadcast to a peer-to-peer network comprising numerous nodes. Each node receives the transaction request, ensuring wide distribution across the network.

### **2. Validation**

The nodes in the network validate the transaction using consensus mechanisms. One commonly used mechanism is Proof of Work (PoW), where nodes (referred to as miners) solve complex mathematical puzzles to verify the authenticity of the transaction. Other mechanisms include Proof of Stake (PoS) and Delegated Proof of Stake (DPoS), which rely on different validation criteria.

### **3. Block Formation**

Once validated, the transaction is grouped with other verified transactions into a block.

Each block contains

- A timestamp.
- A list of transactions.
- A cryptographic hash of the previous block.

These blocks are linked together chronologically, forming a continuous chain. This linkage ensures that altering any block requires changes to all subsequent blocks, providing robust security against tampering.

## **Key Features of Blockchain**

### **1. Decentralization**

Unlike traditional systems that rely on a central authority, blockchain distributes data across all nodes in the network. This eliminates single points of failure and enhances system reliability.

### **2. Transparency**

Blockchain provides a transparent ledger where all participants can view transaction history. This fosters trust among users and minimizes the risk of fraud.

### **3. Immutability**

Once data is recorded on the blockchain, it becomes virtually impossible to alter. The cryptographic hashing ensures that any modification is immediately detectable.

### **4. Security**

The decentralized structure, coupled with advanced cryptographic techniques, ensures a high level of security, protecting data from unauthorized access and cyber threats.

## **Challenges and Limitations**

Despite its advantages, blockchain faces several challenges such as.

- **Scalability:** processing large volumes of transactions remains a bottleneck for many blockchain networks.
- **Energy Consumption:** consensus mechanisms like Proof of Work require significant computational power, raising environmental concerns.
- **Regulatory Uncertainty:** governments and regulators are still grappling with how to govern blockchain-based systems.

### **2.3.2 Consensus Mechanism**

Blockchain consensus mechanisms are protocols that ensure agreement on the state of the blockchain across distributed nodes without requiring mutual trust. These mechanisms are critical for maintaining the performance, integrity, and fault tolerance of distributed ledgers. This section discusses common approaches to blockchain consensus mechanisms, including their principles, advantages, challenges, and vulnerabilities.

### 2.3.2.1 Proof of Work (PoW) Consensus Mechanism

Proof of Work (PoW) is the first and most widely recognized consensus mechanism, utilized by Bitcoin and other permissionless blockchains.

- **Mechanism:** miners compete to solve complex mathematical problems using cryptographic hashing. The first miner to find a valid solution is granted the right to append the next block to the blockchain.
- **Security:** the high energy consumption associated with PoW makes attacks costly. Compromising the network requires over 50% of the hashing power, which is economically prohibitive.
- **Challenges:** PoW faces scalability issues due to slow transaction times and significant energy costs, which hinder its ability to process a high volume of transactions efficiently.

### 2.3.2.2 Proof of Stake (PoS) Consensus Mechanism

Proof of Stake (PoS) is an energy-efficient alternative to PoW, leveraging economic logic to achieve consensus.

- **Mechanism:** validators are chosen based on the size and duration of their cryptocurrency holdings ("stake"). Participants with larger stakes have a higher likelihood of being selected to validate blocks.
- **Security:** while PoS reduces computational requirements, it introduces a centralization risk, as large stakeholders may dominate the network and potentially collude.
- **Challenges:** long-term security concerns include wealth concentration and the potential for exploitation by wealthy participants.

### 2.3.2.3 Delegated Proof of Stake (DPoS) Consensus Mechanism

Delegated Proof of Stake (DPoS) introduces a representative democracy to the consensus process, building on the principles of PoS.

- **Mechanism:** stakeholders vote for a small group of delegates who are responsible for validating transactions. Votes are weighted based on the amount of cryptocurrency held. Delegates can lose their deposits if they act dishonestly.



- **Efficiency:** DPoS improves scalability by requiring fewer nodes to reach consensus, enabling faster transaction processing.
- **Challenges:** the voting process introduces centralization risks, as large stakeholders can dominate delegate selection.

#### 2.3.2.4 Practical Byzantine Fault Tolerance (PBFT) Consensus Mechanism

Practical Byzantine Fault Tolerance (PBFT) is designed to ensure reliability even in the presence of malicious nodes, making it suitable for permissioned networks.

- **Mechanism:** a leader node proposes a block, which other nodes validate. Consensus requires agreement from at least two-thirds of the nodes. A new leader can be elected if the current leader fails.
- **Efficiency:** PBFT is energy-efficient and achieves fast consensus, making it ideal for environments where network participants are pre-selected and trusted to some degree.
- **Challenges:** PBFT requires a minimum of  $3m + 1$  nodes to operate, which can limit decentralization. Additionally, the leader node serves as a single point of failure if compromised.

#### 2.3.2.5 Comparison of Consensus Mechanisms

A comparison of the discussed consensus mechanisms is presented in the table below:

Table 2.1 Comparison of Consensus Mechanisms

Mechanism	Energy Consumption	Scalability	Consensus Type	Vulnerability	Example
PoW	High	Low	Permissionless	Attack with $> 50\%$ hashing power	Bitcoin
PoS	Low	Medium	Permissionless	Collusion by richest nodes	Peercoin
DPoS	Low	High	Elected Delegates	Collusion among delegates	Bitshares

PBFT	Low	Low	Permissioned	> 1/3 dishonest nodes	Hyperledger Fabric (HLF)
------	-----	-----	--------------	-----------------------------	-----------------------------

The table highlights the trade-offs between energy consumption, scalability, consensus type, vulnerabilities, and examples of blockchains using these mechanisms. Each protocol addresses specific challenges based on the intended application and network requirements.

### 2.3.3 Blockchain Architecture<sup>[1]</sup>

Blockchain systems are often modeled using a layered architecture, which abstracts typical blockchain technologies and functional components into six layers from bottom to top: data, network, consensus, incentive, contract, and application layers. The bottom three layers are considered the fundamental elements of a blockchain, while the top three represent extended elements.

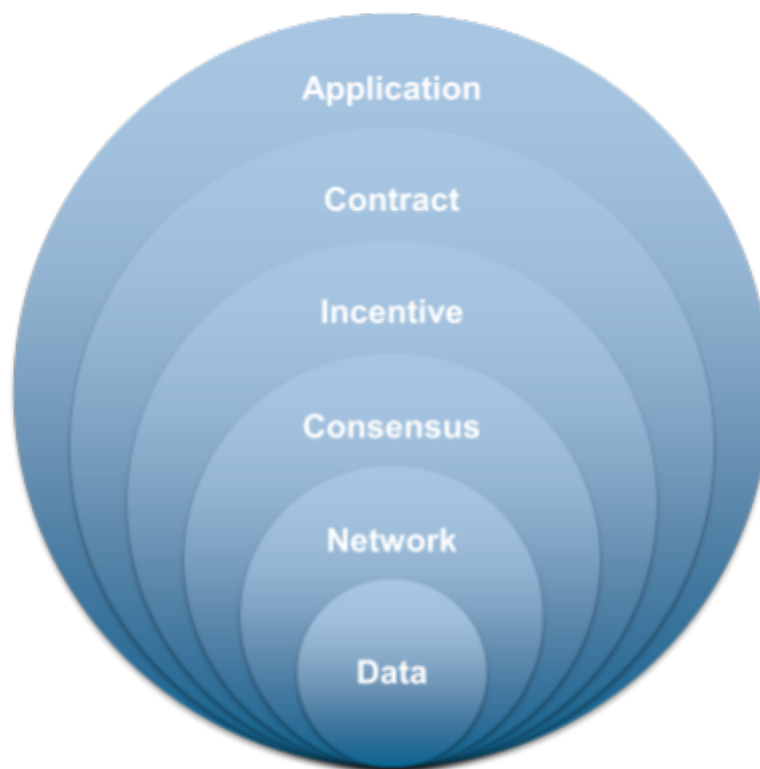


Figure 2.2 Blockchain Architecture layers

### **2.3.3.1 Data Layer**

The data layer defines the schema, structure, and storage of all information within the blockchain. Blockchains use a "chained blocks" data structure as their backbone. Each block contains multiple transactions and essential metadata, such as version, hash, nonce, timestamp, and Merkle root, stored in the block header. The blocks are cryptographically linked using algorithms like encryption, digital signatures, and hashing, creating a tamper-proof database. For instance, Bitcoin employs double iterative SHA-256 for hashing, while Ethereum uses KECCAK-256. Both Bitcoin and Ethereum utilize the Elliptic Curve Digital Signature Algorithm (ECDSA) for transaction signatures.

### **2.3.3.2 Network Layer**

This layer handles the protocols for connecting nodes and validating data transfers across them. Blockchain nodes operate in a peer-to-peer (P2P) network, where all peers maintain the system collaboratively without a central authority. Depending on the type of P2P network—structured or unstructured—different communication protocols are employed. Bitcoin uses a gossip-based protocol for peer selection and state exchange, propagating new transactions to neighboring nodes for validation. Valid transactions are stored for processing, while invalid ones are rejected. Ethereum, on the other hand, uses the Kademlia distributed hash table (DHT) protocol for P2P communication.

### **2.3.3.3 Consensus Layer**

The consensus layer is the core of a blockchain system, defining protocols and algorithms for decentralized nodes to agree on blockchain updates. Proof of Work (PoW) is the most widely recognized consensus algorithm, with alternatives like Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), Proof of Elapsed Time (PoET), Proof of Authority (PoA), and Raft also gaining attention.

### **2.3.3.4 Incentive Layer**

The incentive layer implements mechanisms to motivate participants to validate data and maintain the system. These mechanisms typically include block rewards and transaction fees. For example, Bitcoin rewards miners with 6.25 Bitcoins for successfully mining a valid block, along with transaction fees from the included transactions. This layer is essential in permissionless blockchains. In permissioned blockchains, incentive mechanisms are often optional since participants are preselected organizations.

### 2.3.3.5 Contract Layer

The contract layer introduces decentralized programming paradigms to blockchain systems, initially popularized by Ethereum's smart contract technology. A smart contract is a self-executing program running on the blockchain, enabling innovative applications beyond cryptocurrencies. Variants of smart contracts, such as chaincodes and transaction processors, are used in platforms like Hyperledger Fabric and Sawtooth.

### 2.3.3.6 Application Layer

The application layer provides APIs and programming models for developing specific blockchain applications. Initially associated with cryptocurrency, blockchain has evolved with the advent of smart contract technology. Decentralized applications are now widely adopted across various industries, showcasing significant market potential.

## 2.3.4 Cryptographic Primitives in Blockchains

Cryptographic primitives play a crucial role in ensuring the privacy, security, and integrity of data stored on the blockchain. Since all data on a blockchain must remain public to ensure traceability, verifiability, and immutability, cryptographic techniques are used to hide sensitive user data. This section explores key cryptographic primitives and their applications in blockchain systems.

### 2.3.4.1 Secure Multiparty Computation (MPC)

Secure Multiparty Computation (MPC) is a cryptographic protocol designed to enable multiple parties to jointly compute a function while keeping their inputs private.

- **Mechanism:** MPC distributes the computation of an objective function across multiple parties without revealing the individual data of participants. This ensures privacy while achieving a common computational goal.
- **Applications:** MPC can be used in scenarios like sealed-bid auctions, where it ensures that only the winning bid and the bidder's identity are revealed at the conclusion of the auction. This maintains the confidentiality of all other bids while providing a verifiable outcome.
- **Advantages:** MPC is a robust solution for collaborative computations in trustless environments, ensuring data security and integrity.

#### 2.3.4.2 Zero-Knowledge Proofs (ZKP)

Zero-Knowledge Proofs (ZKP) are cryptographic methods that allow one party to prove knowledge of a specific value to another party without revealing the actual value itself.

- **Mechanism:** the prover demonstrates to the verifier that a statement is true without exposing any underlying data. For example, a ZKP can verify that a bid is within a valid range without disclosing the actual bid amount.
- **Applications:** ZKP is widely used in privacy-preserving blockchain transactions, such as verifying bids in auctions or authenticating transactions without revealing sensitive information.
- **Advantages:** this approach significantly enhances user privacy while ensuring compliance with system rules and constraints.

#### 2.3.4.3 Commitment Schemes (CS)

Commitment Schemes are cryptographic protocols that allow users to commit to a value while keeping it hidden, with the option to reveal it later.

- **Mechanism:** a participant submits a commitment to a value using cryptographic techniques, ensuring that the value cannot be altered after the commitment is made. Elliptic curves are often employed in commitment schemes to ensure the hiding and binding properties of the committed values.
- **Applications:** commitment schemes are primarily used to conceal bid prices in auctions, ensuring fairness while maintaining bidder privacy. The commitment can later be revealed to verify the authenticity of the bid.
- **Advantages:** this ensures both the integrity and confidentiality of sensitive data during the bidding process.

#### 2.3.4.4 Asymmetric Encryption (AE)

Asymmetric Encryption is a cryptographic algorithm that employs a pair of keys: one for encryption (public key) and another for decryption (private key).

- **Mechanism:** data encrypted using the public key can only be decrypted with the corresponding private key, and vice versa. This ensures that sensitive information can be securely transmitted between parties.

- **Applications:** in blockchain-based auctions, bidders encrypt their bids using the auctioneer's public key. Only the auctioneer, possessing the private key, can decrypt and access the bid amounts. This guarantees bid confidentiality while ensuring only authorized access.
- **Advantages:** asymmetric encryption provides a secure method for handling sensitive communications in distributed systems, protecting data from unauthorized access.

### 2.3.5 Types of Blockchains

Blockchain technology encompasses various types that cater to different use cases and organizational needs. Understanding the characteristics of these types helps in determining the appropriate blockchain system for specific applications. Below are the primary types of blockchains: permissioned, permissionless, and hybrid.

#### 2.3.5.1 Permissioned Blockchain

A permissioned blockchain is a closed system where access is restricted to authorized participants only. These blockchains are often employed in environments where privacy, security, and controlled participation are paramount.

##### Key Features

- **Access Control:** only pre-approved participants can read, write, or validate transactions.
- **Consensus Mechanism:** utilizes efficient algorithms like **Practical Byzantine Fault Tolerance (PBFT)** and **Raft** to detect and isolate malicious nodes.

##### Example

- **Hyperledger:** a blockchain framework with implementations such as **Fabric** and **Sawtooth** that support enterprise applications.

##### Pros

- **Improved Performance:** transactions are processed faster due to the limited number of participants.
- **Enhanced Security:** controlled access minimizes the risk of unauthorized activity.

## Cons

- **Reduced Decentralization:** the reliance on a central authority for participant approval may diminish trust.
- **Limited Transparency:** access restrictions can hinder full transparency across the network.

### 2.3.5.2 Permissionless Blockchain

A permissionless blockchain is an open network where anyone can join, participate, and validate transactions without needing prior authorization. This type of blockchain is widely known for its use in cryptocurrencies.

## Key Features

- **Decentralization:** anyone with an internet connection can participate in the network.
- **Consensus Mechanism:** often employs resource-intensive algorithms like **Proof of Work (PoW)** to establish trust among anonymous users.

## Example

- **Bitcoin:** the first and most prominent cryptocurrency that operates on a permissionless blockchain.
- **Ethereum:** a blockchain known for its smart contract capabilities and decentralized applications.

## Pros

- **Full Decentralization:** no central authority governs the network, ensuring transparency.
- **Public Accessibility:** open participation fosters inclusivity and innovation.

## Cons

- **High Energy Consumption:** algorithms like PoW require significant computational power.
- **Slower Transactions:** processing times can be delayed due to the high number of participants.

### 2.3.5.3 Hybrid Blockchain

A hybrid blockchain combines elements of both permissioned and permissionless blockchains, offering customizable levels of privacy and decentralization. This type is ideal for scenarios requiring both public accessibility and private data control.

#### Key Features

- **Customizable Privacy:** allows specific transactions or data to remain private while other activities are open to public verification.
- **Dual Functionality:** supports both public and private operations within a single framework.

#### Example

- **Aergo:** a blockchain platform that combines public chains with private sidechains for enhanced flexibility.

#### Pros

- **Flexibility:** balances privacy and accessibility, making it suitable for regulated industries.
- **Data Control:** organizations can choose what information to share publicly and what to keep private.

#### Cons

- **Complex Design:** integrating public and private functionalities increases system complexity.
- **Regulatory Challenges:** navigating compliance requirements for hybrid systems can be difficult.

### 2.3.6 Blockchains impact on different fields

Blockchain technology has found diverse applications across various industries, providing innovative solutions to long-standing challenges. Below are key areas where blockchain is making a significant impact.



#### 2.3.6.1. Finance

Blockchain technology has revolutionized the financial sector by enabling decentralized transactions. Cryptocurrencies such as Bitcoin and Ethereum provide secure, peer-to-peer methods of transferring value without relying on intermediaries. This enhances security, reduces transaction fees, and minimizes the risk of fraud.

##### Key Benefits

- **Enhanced Security:** transactions are recorded on an immutable ledger, reducing the risk of data breaches.
- **Fraud Prevention:** the decentralized nature and cryptographic techniques make fraud exceedingly difficult.

#### 2.3.6.2 Healthcare

In healthcare, blockchain secures sensitive patient data and ensures secure sharing of health information between providers. By creating a transparent and tamper-proof system, blockchain addresses critical concerns around privacy and data security.

##### Key Benefits

- **Data Protection:** blockchain's encryption ensures patient information remains confidential.
- **Interoperability:** facilitates seamless data sharing across different healthcare systems.

#### 2.3.6.3 Voting

Blockchain technology is being explored for its potential to create secure, tamper-proof voting systems. By providing an auditable and transparent trail of votes, it ensures the integrity of the electoral process.

##### Key Benefits

- **Tamper-Proof Voting:** prevents election fraud by securing votes on an immutable ledger.
- **Transparency:** enables voters to verify their votes while maintaining anonymity.

#### 2.3.6.4 Intellectual Property

Blockchain helps protect intellectual property rights by recording ownership and usage on an immutable ledger. Smart contracts can automate royalty payments, ensuring creators are compensated fairly.

#### Key Benefits

- **Ownership Protection:** verifiable records of ownership deter infringement.
- **Automated Payments:** smart contracts simplify royalty distribution and ensure timely payments.

#### 2.3.7 Smart Contracts

Smart contracts are tamper-proof, self-executing programs that run on blockchain networks. These contracts automate and enforce business rules and logic agreed upon by participating entities, ensuring that agreements are honored without the need for intermediaries. By executing pre-defined conditions encoded in secure code, smart contracts act as autonomous software agents capable of managing digital agreements.

#### General Concept of Smart Contracts

Smart contracts operate on the principle of "if-then" logic. When specific conditions are met, the contract executes the corresponding actions. For example, in a supply chain use case, a smart contract might release payment to a supplier automatically upon confirmation of goods delivery. This automation reduces the need for manual intervention and ensures transparency and efficiency.

#### Gas and Computational Costs

Smart contracts function as digital agreements embedded within blockchain networks, executing automatically when predetermined conditions are met. Unlike traditional contracts that require manual oversight, smart contracts ensure that the agreed terms are enforced programmatically. For instance, in a supply chain, a smart contract might release payment upon verifying delivery conditions, streamlining operations and minimizing disputes.

These contracts are written in code and deployed on the blockchain, where their execution is transparent and immutable. By removing the need for intermediaries, smart contracts provide an efficient and secure method for conducting transactions across various domains, from financial services to logistics and beyond.

- **Gas:** measured in "gwei" (a fraction of Ether), gas is a critical component of executing smart contracts on public blockchains.
- **Optimizations:** developers often optimize smart contracts to minimize gas costs, making transactions more cost-effective for users.

### Examples of Coins and Platforms Supporting Smart Contracts

1. **Ethereum:** the pioneer of smart contract functionality, Ethereum uses its programming language, Solidity, to enable developers to create decentralized applications and execute complex smart contracts.
2. **Binance Smart Chain:** a blockchain that supports smart contracts with low transaction costs and faster processing times.
3. **Cardano:** a blockchain platform that offers secure and scalable smart contract functionality using its Plutus programming language.
4. **Hyperledger Fabric:** although permissioned, Hyperledger Fabric allows for the development of "chaincodes," which serve as smart contracts tailored for enterprise use cases.
5. **EOS:** known for its Delegated Proof of Stake (DPoS) consensus, EOS facilitates high-speed and low-cost smart contract executions.

### Advantages of Smart Contracts

1. **Automation:** smart contracts eliminate the need for intermediaries by automating processes, reducing human error and delays.
2. **Transparency:** all terms and conditions are visible to the participating parties, ensuring clarity and reducing disputes.
3. **Cost Efficiency:** by removing intermediaries and automating workflows, smart contracts lower transaction costs.
4. **Security:** smart contracts are cryptographically secure and immutable once deployed, preventing unauthorized modifications.

5. **Versatility:** they can be applied across diverse industries, including finance, supply chain, healthcare, and real estate.

### **Drawbacks of Smart Contracts**

1. **Immutability:** while immutability is a strength, it can also be a drawback. Errors in smart contract code cannot be corrected without deploying a new contract.
2. **Scalability:** on public blockchains, executing complex smart contracts can be resource-intensive, affecting transaction speed and scalability.
3. **Gas Costs:** high gas fees, especially during network congestion, can make smart contract execution expensive.
4. **Security Risks:** poorly written or audited smart contracts can be exploited by attackers, leading to significant financial losses (e.g., the 2016 DAO hack on Ethereum).

### **Applications of Smart Contracts**

Smart contracts are versatile and have found applications across various domains such as.

1. **Decentralized Finance (DeFi):** automating lending, borrowing, and trading processes without intermediaries.
2. **Supply Chain Management:** tracking goods and automating payments upon delivery.
3. **Healthcare:** managing patient records and automating insurance claim processes.
4. **Real Estate:** streamlining property transactions and escrow agreements.

Smart contracts are transforming industries by enabling trustless, automated, and secure transactions. While they offer numerous advantages, careful consideration of their limitations and thorough auditing of the underlying code is essential to ensure robust implementation.

## **2.4 Conclusion**

This chapter provided a comprehensive foundation on the various auction models and the fundamentals of blockchain technology, offering the necessary background to understand the project and the concepts discussed in the subsequent chapters.

## **Chapter 3**

### **Related Work**



## 3.1 Introduction

The previous chapter explained the necessary scientific background to understand the problem of existing auction systems and the objective of the BitAuction project which aims to solve these problems. This chapter provides a detailed analysis of the foundational research and practical implementations relevant to the auction system proposed in this project. It begins by reviewing related work in Sections 3.2.1 and 3.2.2, including academic papers that underpin the theoretical framework and applications that illustrate real-world implementations. Following this, a critical assessment of the gaps in existing work highlights the need for further advancements in Section 3.3, paving the way for the scope of this project and the points which will be focused on in Section 3.4. The chapter concludes with an exploration of potential extensions and future directions that could enhance the system's performance, scalability, and applicability in Section 3.5. This structured approach aims to establish a clear context for the project while emphasizing its contribution to the broader domain of auction systems.

## 3.2 Related Work

### 3.2.1 Related Work (Published papers)

This table provides a comparative analysis of various published papers, highlighting their features and characteristics across multiple dimensions.

Table 3.1 Literature Review

Papers		[1]	[2]	[3]	[6]	[8]	[9]	[10]	[11]	[12]
1. Year		23	23	24	21	21	20	23	18	19
2. Q-level		Q1	Q2	Q2	N/A	Q1	Conf	Q2	Q2	Q1
3. Software Model		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4. Auction Model	English reverse	✓	✓	✗	✗	✓	✗	✗	✗	✗
	Open-bid	✓	✓	✓	✗	✗	✗	✗	✗	✗
	Sealed-bid	✓	✓	✗	✓	✗	✓	✓	✓	✗
5. Secure		✓	✓	✓	✓	✓	✓	✓	✓	✗
6. Decentralized		✓	✓	✗	✓	✓	✓	✓	✗	✗
7. Smart Contracts		✓	✓	✓	✓	✓	✓	✓	✓	✗
8. Scalable		✗	✗	✗	✗	✗	✗	✓	✗	✗



9. Off-chain Calculations		✗	✓	✗	✗	✗	✗	✓	✗	✗
10. Rate of transactions		N/A	N/A	500 TPS	15 TPS	15 TPS	N/A	10k TPS	15 TPS	N/A
11. Consensus Algorithm	PoW	✓	✓	✗	✓	✓	✗	N/A	✓	✗
	PoS	✓	✓	✗	✗	✗	✗	N/A	✗	✗
	DPoS	✓	✓	✗	✗	✗	✗	N/A	✗	✗
	DBFT	✓	✓	✗	✗	✗	✓	N/A	✗	✗
	Raft	✗	✗	✓	✗	✗	✗	N/A	✗	✗
12. Blockchain Type	permissionless	✓	✓	✗	✓	✓	✗	✗	✓	✗
	Permissioned	✓	✓	✓	✗	✗	✓	✓	✗	✗
	Bitcoin	✗	✓	✗	✗	N/A	N/A	N/A	✗	✗
	Ethereum	✓	✓	✗	✓	N/A	N/A	N/A	✓	✗
	Hyperledger Fabric	✗	✗	✓	✗	N/A	N/A	N/A	✗	✗
13. Mixing Privacy Service		✓	✗	✗	✗	✗	✗	✗	✗	✗
14. Hybrid Blockchain		✓	✓	✗	✗	✗	✗	✓	✗	✗
15. Ordering & Fairness		✗	✓	✓	✓	N/A	✓	✓	✓	✗
16. Smart Contract Updates		✓	✗	✗	✗	✗	✗	✗	✗	✗

17. Payment with Cryptocurrency	✗	✓	✗	✓	✗	✗	N/A	✓	✗
18. Energy Efficient Consensus Models	✗	✓	✓	✗	✗	✗	N/A	✗	✗
19. Initial Price Prediction Support	✗	✗	✗	✗	✗	✗	✗	✗	✓
20. Personalized Recommendations	✗	✗	✗	✗	✗	✗	✗	✗	✓

### 3.2.2 Related Work (Applications)

This table summarizes various auction applications, detailing their technical characteristics, features, and blockchain-related capabilities.

Table 3.2 Applications Review

Auctions	[13]	[14]	[15]	[16]	[17]
1. Open source	✓	✗	✗	✓	✗
2. Year	2022	2017	2020	2020	2002
3. License	Apache 2.0	MIT	N/A	Apache 2.0	N/A
4. Source code availability	✓	✓	✗	✓	✗
5. Support Blockchain	✓	✓	✓	✓	✗
6. Language	JS	TS	TS & JS	Go	N/A
7. Frontend Framework	React	N/A	React	N/A	N/A
8. Backend Framework	Reach	N/A	N/A		N/A

9. Support off chain		✗	✓	N/A	✓	✓
10. Auction model	Open Outcry	✓	✓	N/A	✓	✓
	Sealed bid	✓	✗	✗	✗	✗
11. Support anonymity		✓	N/A	N/A	N/A	✗
12. Most common Currency type		ETH, Algo	ETH	ETH	STARS	N/A
13. Generic auction objects		✗	✗	✗	✗	✓
14. Blockchain technology		Algorand	Solidity	Solidity	Cosmwasm	N/A
15. Support live auction		✗	✓	N/A	✓	✓
16. Arabic Support		✗	✗	✗	✗	✗

### **3.3 Need to Extend Related Work**

A close study for tables 3.2.1 and 3.2.2 suggests the need for the following extensions of related work, where each point references its corresponding entry from the related work tables either academic papers (3.2.1) or applications (3.2.2).

#### **3.3.1 Open-Source Development - ([3.2.2.1](#))**

Create an open-source blockchain based auction platform to encourage community contributions and collaboration especially on open-cry auction by publishing the code on GitHub ensuring proper documentation and licensing for transparent and widespread usage, This model fosters trust among users and stakeholders by providing full access to the system's codebase.<sup>[3]</sup>

#### **3.3.2 Development using the SPI model ([3.2.1.3](#))**

Utilize the Software Process Improvement (SPI) model which enhances the efficiency and quality of the development process. Using SPI ensures that auction systems are designed with optimized workflows and iterative feedback. This model identifies and mitigates bottlenecks, aligning system development with management goals. It also promotes adaptability, allowing developers to incorporate changes dynamically. Overall, SPI results in a more robust and maintainable auction system.

#### **3.3.3 Support for Open-Outcry auction ([3.2.1.4](#))**

Implement an open-outcry auction which involves live bidding, requiring real-time communication and synchronization. Implementing this format digitally demands robust systems capable of handling high concurrency. Open-outcry auctions increase user engagement by offering a dynamic and interactive bidding process. The current literature on open-outcry auctions is limited compared to sealed approaches and this is due to the difficulty of time synchronization problem.<sup>[3]</sup>

#### **3.3.4 Integrate with a decentralized system ([3.2.1.6](#), [3.2.1.12](#))**

Decentralized systems provide transparency, immutability, and security by leveraging blockchain technology. Integration ensures that auction processes are trustless, reducing reliance on central authorities. This fosters user confidence and expands the reach of auction

platforms globally. Utilize Hyperledger as a blockchain solution for the auction system, leveraging its permissioned nature to ensure secure and private transactions. Although most research papers utilize Ethereum smart contracts, Hyperledger is more suitable for our case due to its enhanced privacy features and higher transaction throughput and customization.<sup>[3][8]</sup>

### **3.3.5 Auction privacy protection (3.2.1.13)**

Privacy protection mechanisms safeguard sensitive user and transaction data during auctions. Techniques such as zero-knowledge proofs can validate bids without revealing specific details ensuring that participants' identities and financial information remain confidential, or using Mixing Services can prevent users' addresses from being linked to their real identities, but the mixing function is handled centrally, additionally the usage of permissioned Blockchains where nodes within the same channel can share data, but nodes outside of this channel cannot access it.<sup>[1]</sup>

### **3.3.6 Transaction ordering and synchronization (3.2.1.15)**

Design an efficient transaction ordering which ensures fairness in processing bids during high-traffic auctions. Synchronization mechanisms, such as consensus algorithms, maintain consistency across distributed systems. This could be extended on this paper <sup>[3]</sup>, These processes help avoid conflicts or duplications, ensuring accurate and timely bid placement. This functionality is critical for ensuring smooth operations in large-scale auctions.<sup>[3]</sup>

### **3.3.7 Smart contract update (3.2.1.16)**

Smart contracts automate auction processes, but their immutability poses challenges for updates. Implementing upgradeable smart contracts allows for bug fixes and feature enhancements without disrupting existing functionality, as a solution researchers suggest using proxy contract mechanisms in which OpenZippelin provides a range of libraries to handle proxy logic.<sup>[1]</sup>

### **3.3.8 Auction payment with cryptocurrency (3.2.1.17)**

Implement cryptocurrency payment gateways to allow seamless transactions during the auction where cryptocurrency payments require wallets, multi-signature support, and

compliance with regulations. Smart contracts can further automate payment settlements, ensuring transparency and accuracy.<sup>[1]</sup>

### 3.3.9 Auction scalability solutions [\(3.2.1.8\)](#)

Utilizing different Scalability solutions ensuring that auction systems handle increasing user loads efficiently. Techniques like layer-2 scaling, and optimized algorithms improve performance. These solutions mitigate latency and resource constraints during peak bidding periods. A scalable system provides seamless user experiences regardless of participant numbers. Investing in scalability safeguards against potential bottlenecks as the platform grows. The blockchain trilemma concept argues that a blockchain system cannot support the following three operations.<sup>[6]</sup>

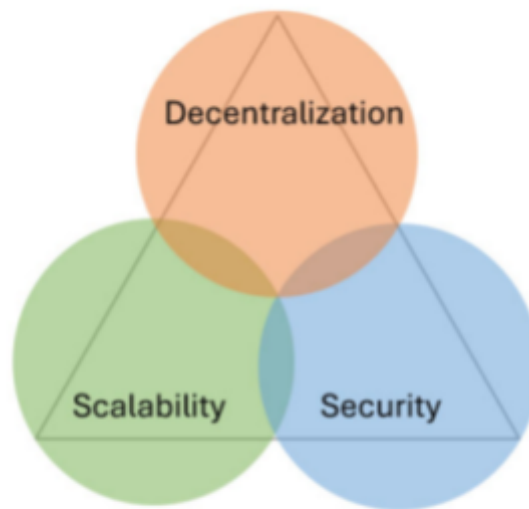


Figure 3.1 The blockchain trilemma

### 3.3.10 Multi chain compatibility

Multi-chain compatibility enables auctions to operate across multiple blockchain networks. It minimizes congestion and prevents one large or complex auction from slowing down the entire system. It allows users to interact with assets and participants from different chains seamlessly.<sup>[8]</sup>

### **3.3.11 Off-Chain data handling [\(3.2.1.9\)](#)**

Off-chain data handling reduces on-chain storage costs and improves processing speed. Using databases or decentralized storage solutions like InterPlanetary File System (IPFS) can manage non-critical data efficiently. This approach ensures that only essential transactions are recorded on the blockchain. Off-chain mechanisms also provide more flexibility for large datasets. Balancing on-chain and off-chain operations enhances system performance and scalability.<sup>[8]</sup>

### **3.3.12 Integration with oracles for real-world data**

Oracles bridge the gap between blockchain systems and real-world data. They provide reliable feeds for variables like auction item appraisals, exchange rates, or event timings. By integrating oracles, auction systems can operate dynamically and contextually. It ensures that decisions and processes are based on accurate and timely information. Trustworthy oracles are crucial for automating smart contract conditions and actions.<sup>[8]</sup>

### **3.3.13 Conduct evaluation for the system**

Evaluate the performance of the blockchain layer, focusing on transaction throughput, latency, and gas costs under varying levels of user activity, and the application's overall performance, including response times, resource utilization, and scalability, to ensure smooth user experiences during high-load scenarios. Testing should include real-world scenarios to validate scalability and robustness. Comprehensive evaluations enhance the system's competitiveness and user experience.<sup>[3]</sup>

### **3.3.14 Use FastFabric as an underlying system**

FastFabric, a blockchain framework, offers high throughput and low latency for auction systems. Its modular architecture supports customizability and scalability. Using FastFabric ensures faster transaction processing and consensus times. It is well-suited for applications requiring real-time operations, like auctions. Leveraging this framework can enhance system performance and adaptability. FastFabric supports up to 20,000 transactions per second, making it an ideal choice for the auction system, where a high volume of transactions needs to be processed quickly and efficiently.<sup>[19]</sup>

### **3.3.15 Support dynamic live auction [\(3.2.2.15\)](#)**

A dynamic interface engages users with real-time updates and interactive features. Tools like live bid tracking, countdown timers, and visual analytics enhance the user experience. Ensuring responsiveness across devices is crucial for accessibility. Such interfaces can include personalized views for bidders and auctioneers. A well-designed interface fosters participation and satisfaction in live auctions.

### **3.3.16 Support personalized recommendations [\(3.2.1.19\)](#)**

Personalized recommendations suggest auction items based on user preferences and behaviors. Machine learning algorithms analyze past activities to tailor these suggestions. This feature increases user engagement and the likelihood of successful bids. Recommendation systems can also enhance the discoverability of items. Personalization creates a more user-centric auction experience.<sup>[12]</sup>

### **3.3.17 Support price prediction on auction items [\(3.2.1.19\)](#)**

Price prediction tools estimate item values based on historical and market data. Classical machine and deep learning models can identify patterns and provide accurate price ranges. This helps bidders make informed decisions and sellers set realistic expectations. Predictive analytics add transparency and efficiency to auction processes. Implementing this feature positions the platform as data-driven and user-friendly.<sup>[12]</sup>

### **3.3.18 Develop auction user-friendly interface**

A user-friendly interface simplifies navigation and interaction within the auction system. Clear layouts, intuitive design, and accessibility features ensure inclusivity. Features like guided tutorials and tooltips assist new users. The interface should also cater to diverse device types and screen sizes. A well-designed interface boosts user retention and satisfaction.

### **3.3.19 Design and implement a sealed bid auction. [\(3.2.1.4\)](#)**

A sealed bid auction is a type of auction where bidders submit their bids confidentially, without knowing the bids of others. The auctioneer collects all bids, determines the winner, and announces the results without revealing individual bids publicly. The implementation involves designing a mechanism where bids are encrypted or hashed to ensure privacy, and only the auctioneer has access to the full details during evaluation.



Additionally, a smart contract could be used to automate the auction process, ensuring transparency and tamper-proof results.

### **3.3.20 Optimize the bidding stage by applying off-chain operations. [\(3.2.1.9\)](#)**

To enhance the efficiency of the bidding stage, computations can be moved off-chain. Instead of each bidder submitting their bid directly on-chain (which results in N separate transactions, leading to higher gas fees and slower processing), bidders communicate their bids to the auctioneer in the commitment stage. The auctioneer aggregates these bids and submits a single commitment to the blockchain, reducing the number of on-chain writes to just one. This approach minimizes blockchain usage while maintaining integrity through cryptographic commitments, ensuring that the auctioneer cannot tamper with or alter the submitted bids.

### **3.3.21 Compare and test the effect of this added feature [\(3.2.1.18\)](#)**

This step involves analyzing the performance improvements introduced by the off-chain optimization. Key metric (only the gas fees) should be compared between the optimized sealed bid auction and a traditional on-chain implementation. The test would demonstrate how reducing n writes to a single write impacts cost-effectiveness and scalability. Additionally, testing ensures that the optimized method preserves fairness and security, validating the cryptographic commitments against tampering or fraudulent behavior. Quantitative results from these tests would showcase the level of optimization achieved and its practical benefits.

## **3.4 Scope of Work**

After a close study of table 3.2.1 and 3.2.2, It was agreed on to use the paper [3] as a foundation stone for implementing the open-cry auction system that uses blockchains. The scope of work covers and references points from the need to extend related work, the rest is left for future work.

1. Open Source [\(3.3.1\)](#)
2. Documentation using the SPI model [\(3.3.2\)](#)
3. Support for open-cry auctions [\(3.3.3\)](#)
4. Integration with a decentralized system [\(3.3.4\)](#) [\(3.3.5\)](#)

- Use the Hyperledger Fabric blockchain to integrate it with our auction system.
  - Build smart contracts (chaincode) on the Hyperledger Fabric to enforce auction rules (3.3.4)
5. Design an efficient transaction ordering system in the ordering service in Hyperledger Fabric (3.3.6)
    - Use a trusted external API to get timestamps for each bid.
    - Use the timestamps to order bids inside the system.
  6. Use off-chain data storage like IPFS to store auction related data outside the blockchain (3.3.11)
    - Store metadata of the auction on the IPFS.
  7. Support dynamic live auction interface ensuring low latency and real-time updates (3.3.15)
  8. Design an intuitive user interface (3.3.18)
  9. Improve scalability by using multichain solutions, assigning each chain to a specific subset of auction items. (3.3.8)
  10. Evaluate the system for both layers, blockchain and the application (3.3.13)
  11. Design another sealed bid auction smart contract that optimizes the bidding phase. (3.3.19)
    - This solution depends on off-chain calculations to reduce gas costs.
  12. Compare the sealed bid auction optimization to on-chain only solutions.(3.3.20)
  13. Compare and test the effect of the off-chain optimization for the bidding phase added feature (3.3.21)

## 3.5 Future Work

Future work aims to enhance the auction system by addressing scalability, integration, and user-centric features to ensure robust functionality and seamless operation in real-world environments.

14. Auction payment with cryptocurrency: The integration of cryptocurrency as a payment method will enable secure and decentralized transactions. (8)
15. Try promising auction scalability solutions to address potential bottlenecks in high-demand scenarios. (9)
16. Integration with Oracles for Real-World Data: Enables the auction system to incorporate real-time data from external sources, such as dynamic pricing, currency

exchange rates, and provide users with accurate and relevant information during auctions. (12)

17. Use FastFabric as an Underlying System: Leveraging FastFabric as the foundational blockchain system will enhance the overall efficiency of the auction platform. (14)
18. Support Personalized Recommendations: Enhance the user experience by suggesting auction items based on user preferences and browsing history. (16)

### **3.5 Conclusion**

In conclusion, this chapter has presented a comprehensive overview of related work, spanning academic literature and practical applications, and identified critical gaps that necessitate further innovation. The defined scope of work addresses these gaps, focusing on advancing the auction system through the integration of cutting-edge technologies and methodologies. Collectively, the insights from this chapter establish a solid foundation for the project while outlining pathways for continued exploration and refinement.

## **Chapter 4**

# **Software Development Process and Implementation**

## 4.1 Introduction

In chapter 3, a discussion about the related work and related applications of using blockchain to build a secure auction system. In this chapter, The software development process of the project will be shown which is the Software Process Improvement (SPI) model in Section 4.2, next we will discuss the requirements of our project in Section 4.3, the solutions design in Section 4.4 and the tools used in in Section 4.5, then we will show the main implementation features for our application in Section 4.6 and the results in Section 4.7, Finally, the chapter is concluded in Section 4.8

## 4.2 Software Process Improvement (SPI)

### 4.2.1 What is SPI?

The Software Process Improvement (SPI) model was developed by the Software Engineering Competence Center (SECC) to assist small and medium-sized enterprises in enhancing the quality of their products to meet international standards. This is achieved by adopting contemporary software development processes and practices in a cost-effective manner.

SPI encompasses five key processes: project management, product development, peer review, quality assurance, and configuration management. These processes are implemented across four phases of a project:

- **Initiation:** Focuses on defining the project scope, identifying stakeholders, and creating a preliminary project management plan.
- **Planning:** Involves developing a detailed project plan that includes a schedule, estimates of size, effort, and cost, as well as a risk management plan.
- **Execution, Monitoring, and Control:** Entails executing the project plan, managing resources, tracking progress, and making necessary adjustments.
- **Closing:** Involves finalizing the project and resolving any remaining issues.

Each SPI process involves specific tasks for each phase and generates certain work products (outputs). Since this project is academic rather than commercial, not all SPI processes were applied. We adapted the SPI model to suit the nature of the project, and this chapter provides a summary of the adapted processes we utilized[22].

#### 4.2.2 Project Management Process (PMP)

The project management process is a methodical approach to planning, organizing, and managing resources to achieve specific project objectives. This framework helps in defining project activities, monitoring progress, and implementing corrective actions when necessary. By adhering to a structured process, project managers can enhance the likelihood of successfully meeting the project's goals and objectives.

For our blockchain auction project, we created four key work products:

- **Minutes of Meetings (MoM):** After each meeting, we prepared MoM documents that recorded the meeting's location and time, attendees, main discussion points, action items to be completed by the next meeting, and the agreed-upon location and time for the next meeting.
- **Process-Activity-Task Matrix (PATM):** We developed a PATM that outlined the main processes involved in project development. Each process was broken down into smaller units called activities, which were further subdivided into specific tasks.
- **Weekly Time Sheets:** This sheet detailed the tasks each team member worked on every week and the number of hours spent on each task. This approach facilitated tracking the collective contributions and managing the project's progress effectively.
- **Implementation Plan:** We prepared a comprehensive Implementation Plan that detailed each task's estimated start and end dates, assigned responsibilities, and completion status. This plan not only outlined the timeline for each task but also specified who was responsible for each task and included a progress percentage to track the completion of each task.

### 4.2.3 Product Development Process (PDP)

The product development process encompasses all engineering activities that transform customer requirements into a final product through a series of steps. This process operates under the supervision of the project management process and is supported by peer review, quality assurance, and configuration management processes.

The product development process includes numerous procedures such as requirements planning, requirements elicitation, requirements analysis, requirements development, requirements validation, requirements acceptance, requirements management, development planning, architecture designing, detailed designing, implementation, component testing, integration testing, system testing, acceptance testing, and finally, product release.

Process	Activity	Task
<b>Security Affairs (SA)</b>	User Authentication & Access Control (UAAC)	Implement Web3 wallet-based user registration and login
		Conduct smart contract and system-wide security audits
<b>Bidders' Affairs (BA)</b>	Bidding Process (BP)	Implement bid placement validation logic enforcing auction rules
		Enable real-time WebSocket-based updates for live bidding
		Notify bidders in real-time about bid status and auction end
<b>Creator of Auction Affairs (CAA)</b>	Auction Management (AM)	Develop clean and intuitive web UI for auction creation
		Allow creators to define rules like starting bid, duration, minimum increment
		Notify auction creators of key events

		Store auction metadata off-chain using database
<b>Blockchain Affairs (BCA)</b>	Smart Contract Execution (SCE)	Implement auction logic as chaincode in Hyperledger Fabric
		Secure on-chain wallet and fund transfer mechanisms
<b>Notification Affairs (NA)</b>	Auction & Bidding Notifications (ABN)	Notify users before auction expiration or at timeout
		Notify seller and winning bidder of payment confirmation
<b>Ordering Service Affairs (OSA)</b>	Timestamp Synchronization (TS)	Integrate trusted time source using NTP and oracles
		Securely store timestamps for each auction and bid event
		Ensure bid ordering and fairness based on stored timestamps

Table 4.1: Process-Activity-Task Matrix

In the development of this project, several procedures were implemented to ensure the high quality of the final product, resulting in the following outputs:

- **Software Requirements Specifications (SRS):** This document outlines the functional and non-functional requirements of the blockchain auction project. It serves as a detailed guideline for the development team, ensuring that all customer needs and expectations are clearly defined and met. The SRS includes use cases, user stories, and specific requirements for performance, security, and usability.
- **Architecture and Detailed Design:** This output includes the architectural design of the system, detailing the overall structure and how the different components interact with each other. The detailed design further breaks down each component, providing specific information about the design patterns, data structures, and algorithms to be used for the blockchain auction system.



- **Test Plan:** This plan outlines the testing strategy and procedures to ensure that the final product meets the specified requirements and is free of defects. It includes various types of testing, such as component testing, integration testing, system testing, and acceptance testing for both the blockchain network and web application components.

#### **4.2.4 Peer Review Process (PRP)**

The peer review process aims to detect and remove defects from work products early in the development cycle through a systematic examination by the author's peers. In our blockchain auction project, in addition to the timesheets detailing each team member's tasks as discussed in the project management process, we implemented peer review documents.

These documents listed each task, identified the task owner, specified the peer reviewer, and documented any problems detected by the peer reviewer. This peer review process facilitated thorough scrutiny of our work, enabling us to promptly identify and address any issues in both the smart contract development and web application implementation.

By leveraging everyone's knowledge, we enhanced the quality of our work and minimized the likelihood of encountering problems later on, particularly important given the security-critical nature of blockchain applications.

#### **4.2.5 Configuration Management Process (CMP)**

The configuration management process ensures a secure infrastructure for the entire project and the software development life cycle by storing all evolving work products in a controlled environment.

In our blockchain auction project, several primary tools were utilized for configuration management:

- **GitHub:** This platform was employed for configuration management and version control of the project source code, including smart contracts, web application code, and deployment scripts. It allowed us to track changes, collaborate on code development, and maintain a history of revisions across all project components.

- **Jira:** Used for issue tracking and project management, allowing the team to efficiently prioritize and resolve issues throughout the development process, particularly for tracking blockchain-specific bugs and smart contract vulnerabilities.
- **Confluence:** Served as our documentation repository, storing all project-related documents, meeting notes, and technical specifications in a centralized location accessible to all team members.
- **Google Drive:** We used Google Drive as a backup storage solution for all project-related documents, maintaining all versions of these documents. This facilitated easy access and sharing of project documentation among team members, ensuring everyone had access to the most up-to-date information.

#### 4.2.5 Quality Assurance Process (QAP)

The quality assurance process involves testing, validation, and verification methods to ensure that the software meets defined quality standards and functions as intended. This process helps identify and eliminate defects, ensuring the system is reliable, consistent, and aligned with the project's requirements.

The quality assurance process for this project was overseen by our academic supervisors: Prof. Dr. Mohamed S. Abougabal, Prof. Dr. Shaimaa Lazem, and Dr. Amira Alshazly. Their role was to ensure that our work aligned with the methodology they provided and met the academic standards of our university. By holding us to a high standard—often treating us as master's level students—they helped elevate the quality of our work. Their continuous guidance and feedback played a key role in maintaining rigor and improving the overall outcome of the project.

### 4.3 Software Requirements Specifications (SRS)

#### 4.3.1 Scope of the Software

- **Blockchain Integration:** Implement Hyperledger Fabric blockchain technology to create a secure and decentralized system for conducting transparent and tamper-proof auctions.
- **Security and Transparency:** Prioritize the security and transparency of auction processes by leveraging the cryptographic features and decentralized structure

of blockchain to prevent fraud, bid manipulation, and ensure fair auction outcomes.

- **Immutability and Tamper-Proofing:** Utilize blockchain's immutable nature to ensure that auction records, bids, and transactions are tamper-proof, maintaining the integrity of the entire auction process.
- **Real-time Bidding System:** Implement a transparent and accountable auction system through the decentralized nature of blockchain, providing a clear audit trail of all bids, transactions, and auction activities.
- **Two-Stage Bidding Mechanism:** Develop an innovative bidding process that combines privacy during bid creation with transparency during bid submission, ensuring fair competition while preventing bid sniping.
- **Smart Contract Integration:** Utilize smart contracts for automated auction execution, transaction settlement, and escrow services, reducing the need for intermediaries and ensuring trustless operations.
- **Timestamp Synchronization:** Implement precise timestamp ordering through NTP-based oracles to ensure fair bid sequencing and prevent timing-based manipulation.
- **Digital Asset Focus:** Design the platform exclusively for digital assets and NFTs, providing a specialized environment for blockchain-based collectibles and digital goods.
- **Contributions to Decentralized Commerce:** Provide significant insights and contributions to the discourse on utilizing blockchain technology to revolutionize online auctions, contributing to the advancement of decentralized commerce.

## **4.3.2 System Specifications**

### **4.3.2.1 User Requirements**

- As a user, I want to easily authenticate using my Web3 wallet with just a few simple steps. I should feel confident that my blockchain identity is securely connected to the platform.
- As a seller, I want to create auctions for my digital assets and NFTs with customizable parameters including starting bid, reserve price, and auction duration.

- As a seller, I want the assurance that my auction rules are enforced automatically through smart contracts, ensuring no manipulation or unauthorized changes can occur.
- As a buyer, I want to participate in real-time auctions where I can see all bids as they happen, ensuring complete transparency in the bidding process.
- As a buyer, I want to create private bids initially and then submit them publicly when I'm ready, giving me strategic control over my bidding approach.
- As a buyer, I want the ability to withdraw my bids before the auction ends if I change my mind, with automatic refund of my deposited funds.
- As a user, I want to view the complete history of all my auction activities, including bids placed, auctions won, and transaction records.
- As a user, I want real-time notifications about auction events, bid status updates, and transaction confirmations through the platform's messaging system.
- As a user, I want to search and filter auctions based on categories, price ranges, and auction status to easily find items of interest.

#### **4.3.2.2 Functional Requirements**

1. User Authentication and Access Control
  - Users must authenticate securely using Web3 wallets (MetaMask, WalletConnect)
  - The system must implement role-based access control for sellers and buyers
  - Support for blockchain-based identity verification
2. Auction Management
  - Sellers can create auctions with customizable parameters
  - Support for digital assets and NFT auctions only
  - Automatic enforcement of auction rules through smart contracts
  - Prevention of auction modifications after creation
3. Two-Stage Bidding Process
  - Create Bid: Private bid creation with encrypted bid details

- Submit Bid: Public bid submission with timestamp verification
- Bid withdrawal functionality before auction completion
- 4. Blockchain Integration
  - Utilize Hyperledger Fabric for secure transaction processing
  - Implement smart contracts for automated auction execution
  - Ensure immutable storage of all auction data and transactions
- 5. Transaction Settlement
  - Automatic asset transfer upon auction completion
  - Smart contract-based escrow services
  - Automated refund processing for unsuccessful bidders
- 6. Timestamp Synchronization
  - Integration with NTP-based oracles for accurate timestamping
  - Deterministic bid ordering using transaction ID hashing
  - Prevention of timing-based bid manipulation
- 7. Real-time Communication
  - WebSocket-based real-time bid updates
  - Live auction status notifications
  - Instant transaction confirmations

#### **4.3.2.3 Non-Functional Requirements**

1. Security
  - End-to-end encryption for all bid transactions
  - Smart contract security audits before deployment
  - Fraud detection and prevention mechanisms
  - Protection against bid manipulation and sniping
2. Performance
  - Bid updates must reflect within 1 second
  - Smart contract execution time under 1 second
  - Support for up to 1000 concurrent auctions
  - Handle up to 1000 bids per auction
3. Scalability

- Design system to accommodate increasing numbers of users and auctions
  - Efficient resource utilization for blockchain operations
  - Optimized smart contract execution
4. Usability
    - Intuitive user interface for both technical and non-technical users
    - Clear auction participation instructions
    - Responsive design for desktop and mobile devices
  5. Reliability
    - 99.9% system uptime availability
    - Robust error handling and recovery procedures
    - Blockchain-based data redundancy and backup
  6. Compliance
    - Adherence to blockchain security best practices
    - Compliance with digital asset trading regulations
    - Regular security updates and patches
  7. Interoperability
    - Seamless integration with popular Web3 wallets
    - Compatibility with major blockchain networks
    - Support for various digital asset standards (ERC-721, ERC-1155)
  8. Maintainability
    - Modular architecture for easy updates and bug fixes
    - Comprehensive logging and monitoring systems
    - Detailed documentation for system maintenance and troubleshooting

## 4.4 Software Design and Architecture

### 4.4.1 Work Overview

The workflow diagram illustrates the complete process flow and decision points within the BitAuction system, showing how different components interact from auction creation through final settlement. The diagram demonstrates the systematic progression of activities including user authentication, auction setup, two-stage bidding process, real-time monitoring, and

automated settlement, ensuring a streamlined and organized workflow for conducting secure and transparent blockchain-based auctions.

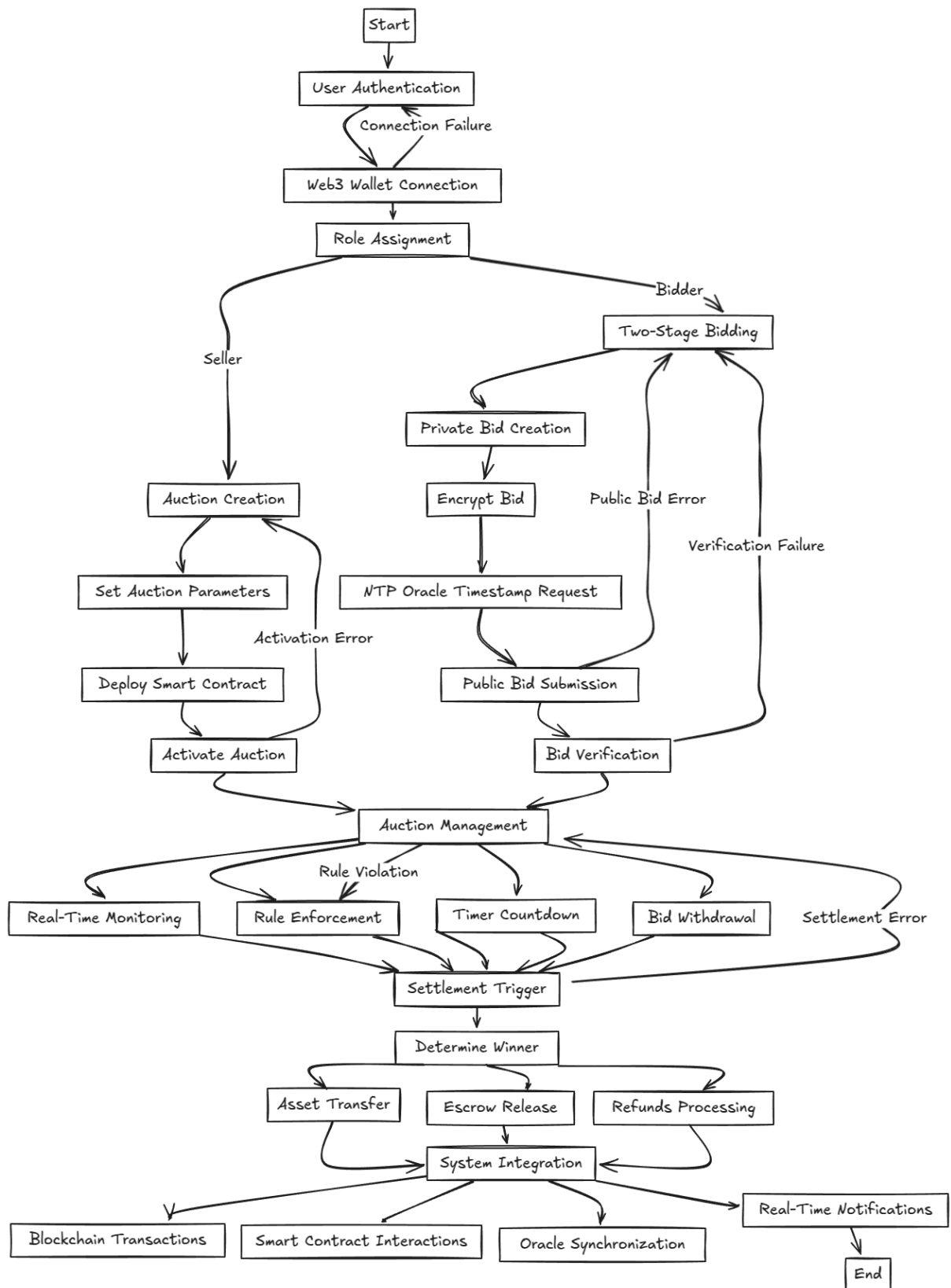


Figure 4.1: Work Overview



## 4.4.2 Hyperledger Network

The implementation of our auction system utilizes a Hyperledger Fabric network, designed to facilitate secure and transparent auction processes across multiple participating organizations. The network is structured to support decentralized bidding mechanisms with tamper-proof timestamp verification and fair auction settlement procedures.

### 4.4.2.1 Key Components

#### Organizations

- **Organization 1-4:** Four peer organizations, each capable of performing multiple roles within the auction ecosystem:
  - **Auction Creation:** Any organization can create auctions and act as sellers
  - **Bidding Participation:** All organizations can participate as bidders in auctions
  - **Transaction Endorsement:** Each organization maintains peers responsible for endorsing transactions and maintaining the distributed ledger
  - **Time Oracle Services:** All organizations provide tamper-proof timestamp services through NTP synchronization for bid ordering and auction deadline enforcement
- **Orderer Organization:** Facilitates the ordering service, ensuring consensus and proper sequencing of transactions across the auction network.

#### Channels

- **Primary Channel (mychannel):** The unified communication channel where all auction-related activities occur, including:
  - Auction creation and management
  - Bid submission and settlement
  - Time oracle timestamp verification services
  - Inter-organizational communication and data exchange

## Smart Contract Architecture

- **Auction Smart Contract:** Core chaincode implementing auction lifecycle management with functions for:
  - Auction creation with seller identity verification
  - Bid submission with price validation
  - Timestamp recording through oracle integration
  - Auction settlement with highest bid determination
- **Time Oracle Chaincode:** Specialized chaincode providing NTP-synchronized timestamps for ensuring fair bid ordering and auction deadline enforcement.

## Data Storage and State Management

- **World State Database:** Stores auction records, bid information, and participant data with composite key indexing for efficient querying.
- **State-Based Endorsement:** Dynamic endorsement policies ensuring only authorized organizations can modify specific auction states.

## Supporting Components

- **Transaction ID-Based Bid Tracking:** Unique transaction identifiers ensure bid integrity and prevent double-spending attacks.
- **Deterministic Timestamp Shuffling:** CRC32-based encoding and seeded randomization for fair timestamp selection in multi-endorser scenarios.
- **Composite Key Management:** Efficient bid indexing using auction ID and transaction ID combinations for optimized data retrieval.
- **Client Identity Verification:** X.509 certificate-based authentication ensuring secure participant identification and authorization.

## Security Features

- **State-Based Endorsement Policies:** Dynamic policy assignment ensuring only auction sellers can end their auctions.
- **Bid Validation:** Comprehensive price validation and bidder authorization checks preventing malicious bidding.

- **Timestamp Integrity:** Oracle-based timestamp verification preventing bid time manipulation attacks.

This robust network architecture ensures auction integrity, participant privacy, and transparent settlement while maintaining decentralized governance across all participating organizations. The integration of external time oracles provides additional security guarantees for time-sensitive auction operations.

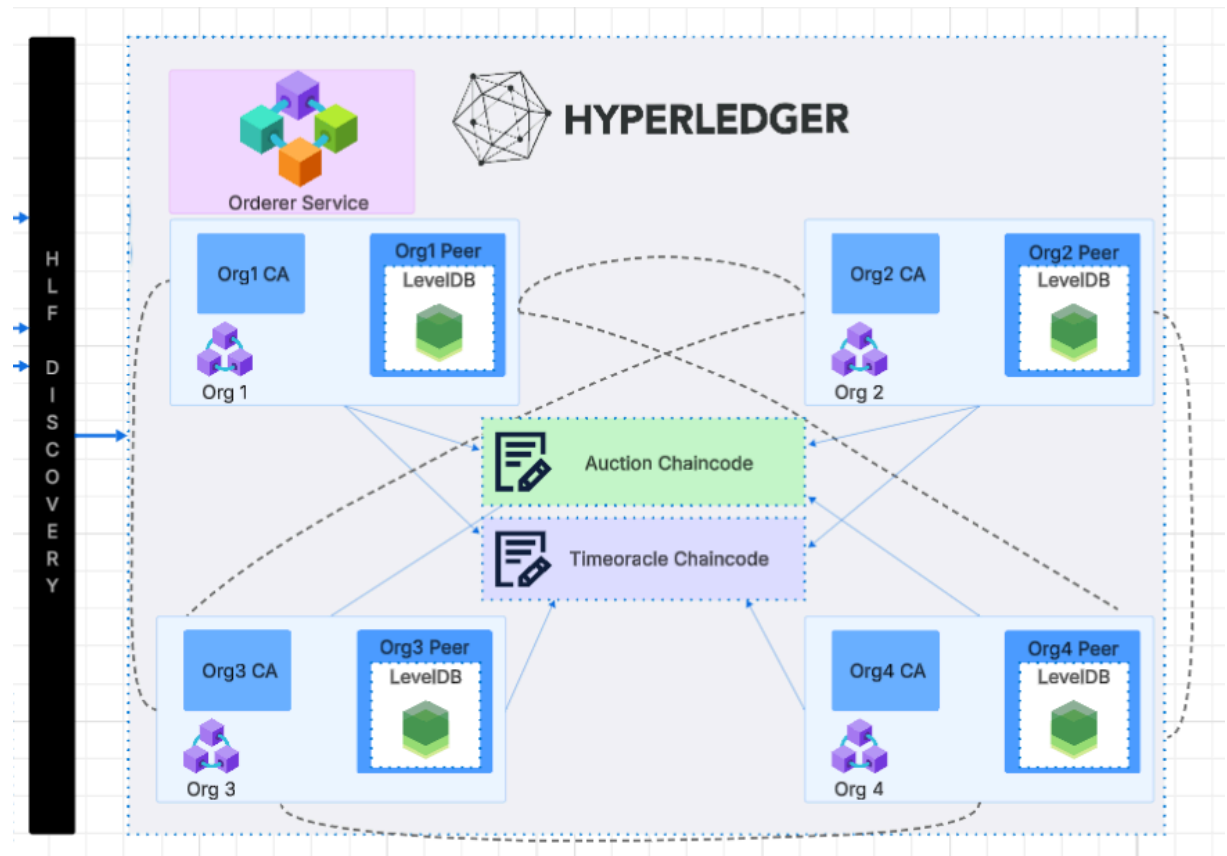


Figure 4.2: Hyperledger Fabric Network

### 4.4.3 Database Design

Our database is structured using PostgreSQL, reflecting a relational design following traditional ER (Entity-Relationship) model principles. The database schema is designed to support the blockchain-based auction system while maintaining data integrity and efficient querying capabilities. The diagram illustrates the relationships and data fields for various entities, such as users, auctions, bids, notifications, and other key components within the decentralized auction system.

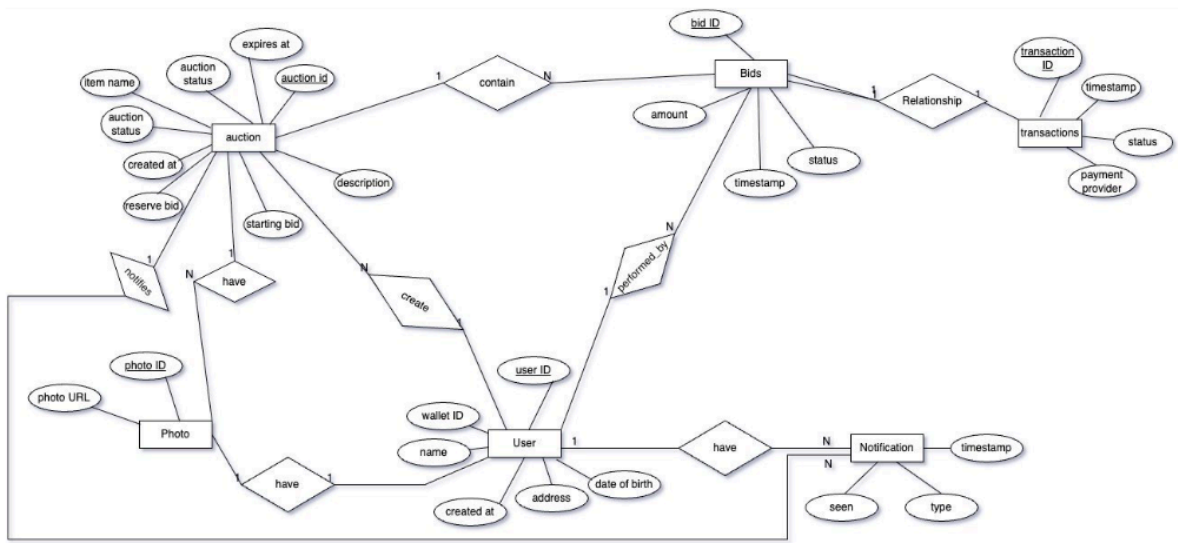


Figure 4.3: Database ERD

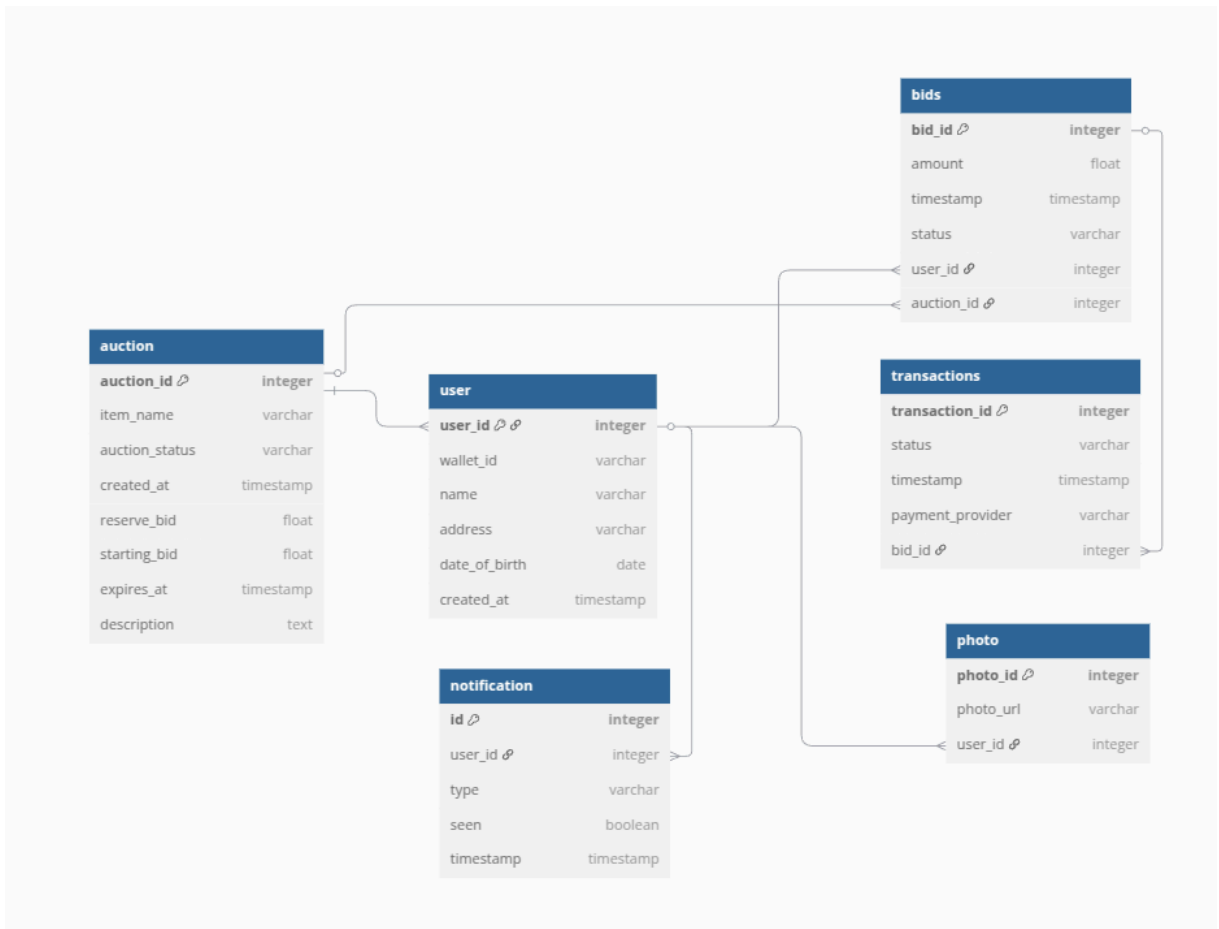


Figure 4.4: Relational schema

#### 4.4.4 Sequence Diagrams

Sequence diagrams are a type of UML diagram used to illustrate how objects interact in a particular sequence over time. They provide a visual representation of the flow of messages between various components within a system, highlighting the order in which these interactions occur. A sample of the sequence diagrams is shown here..

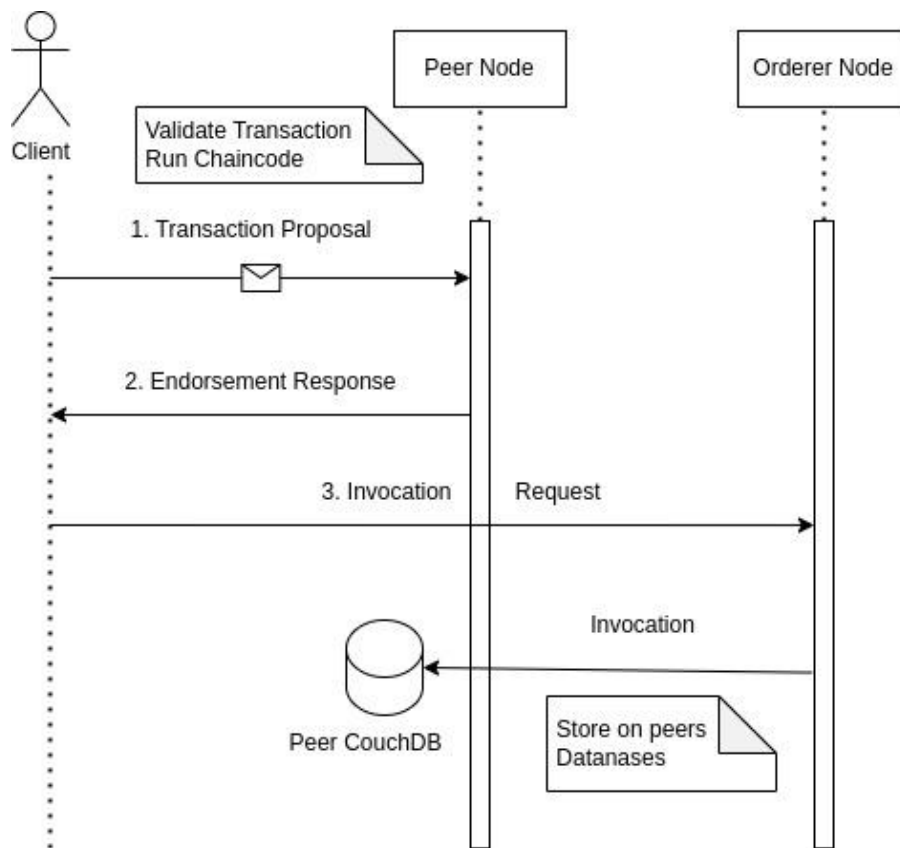


Figure 4.5: Sequence Diagram 1

This diagram shows how a client submits a transaction: it sends a proposal to a peer, gets endorsement, then sends it to the orderer, which commits it to the ledger via peers.

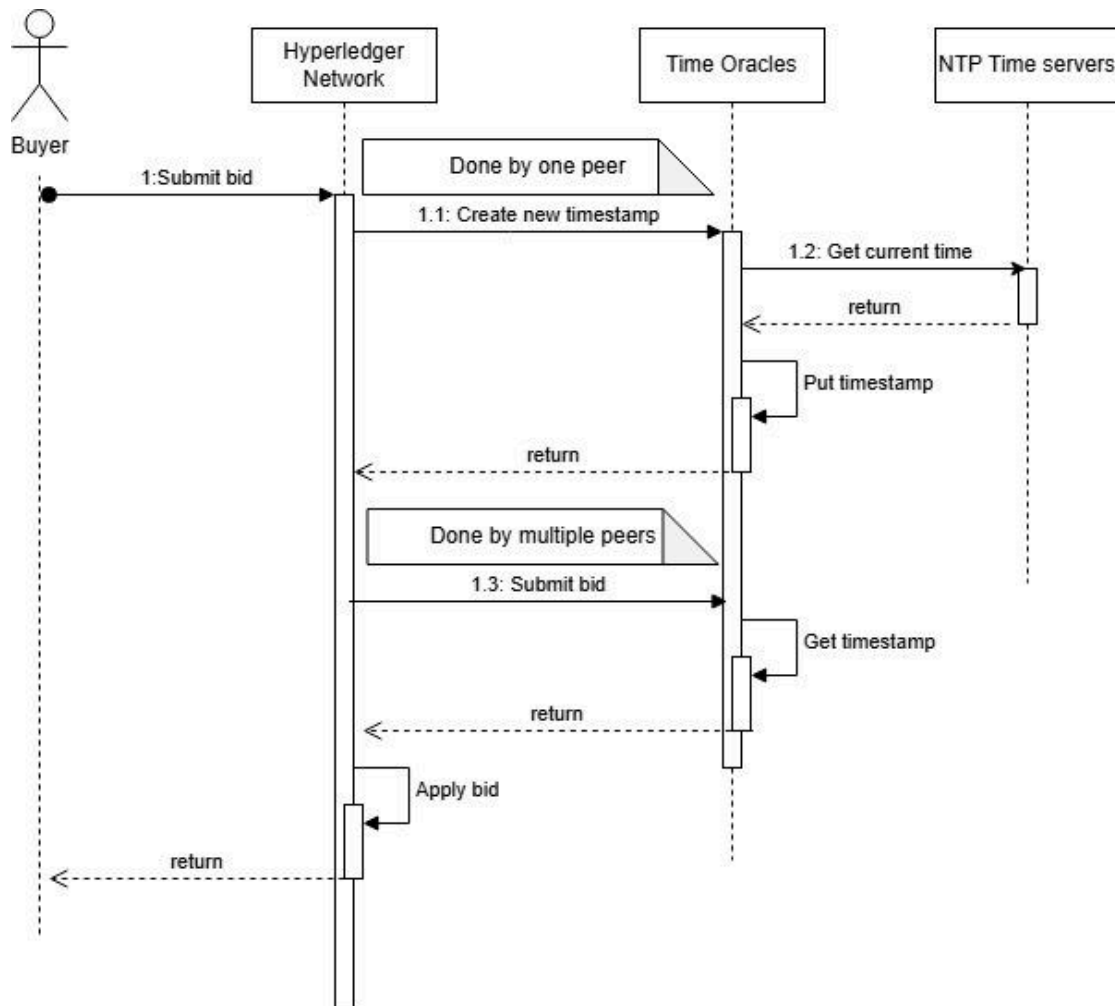


Figure 4.6: Sequence diagram 2

This diagram describes how a buyer submits a bid: a peer gets a trusted timestamp from time servers, then multiple peers validate and apply the bid with that timestamp.

## 4.5 Platforms and Tools

### 4.5.1 Front-end Tools

#### 4.5.1.1 React.js[23]

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on components. Maintained by Meta (formerly Facebook) and a community, it enables efficient rendering by using a virtual DOM and supports development of single-page, mobile, or server-rendered applications[24].

## **4.5.2 Back-end Tools**

### **4.5.2.1 NestJS[25]**

NestJS (or simply Nest) is a free and open-source, server-side Node.js framework built with TypeScript. Inspired by Angular, it uses a modular architecture to support OOP, FP, and FRP styles, and is released under the MIT License[26].

### **4.5.2.2 PostgreSQL[27]**

PostgreSQL (or Postgres) is a free and open-source object-relational database system that emphasizes extensibility and SQL compliance. It supports ACID transactions, advanced features like foreign keys, views, triggers, and persists complex workloads reliably[28].

### **4.5.2.3 Redis[29]**

Redis (Remote Dictionary Server) is an open-source, in-memory data structure store, often used as a database, cache, and message broker. It supports various data structures and offers low-latency operations thanks to its in-memory architecture[30].

### **4.5.2.4 WebSocket[31]**

WebSocket is a standardized computer communications protocol providing full-duplex communication channels over a single TCP connection. Named in RFC 6455 (2011), it enables persistent, bidirectional messaging without HTTP polling[32].

## **4.5.3 Blockchain tools and platforms**

### **4.5.3.1 Hyperledger Fabric[33]**

Hyperledger Fabric is an open-source, permissioned blockchain framework, started in 2015 by the Linux Foundation. It is a modular, general-purpose framework that offers unique identity management and access control features, which make it suitable for a variety of industry applications such as track-and-trace of supply chains, trade finance, loyalty, and rewards, as well as clearing and settlement of financial assets[34].



## 4.5.4 Version Control

### 4.5.4.1 Git[35]

Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers who are collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different computers)[36].

## 4.6 Implementation

### 4.6.1 Chaincode

The blockchain-based auction system operates through a comprehensive six-stage process as depicted in Figure 4.7. The workflow initiates with auction creation, where sellers define auction parameters including item details, time limits, and starting conditions through the CreateAuction function. Subsequently, participants submit their bids using the Bid and SubmitBid functions, with each bid being cryptographically secured and timestamped. The system then validates submitted bids through smart contract verification, ensuring bid authenticity and compliance with auction rules. Concurrently, timestamp verification occurs through integration with the time oracle chaincode, providing tamper-proof chronological ordering of bid submissions. As the auction reaches its predetermined time limit, the EndAuction function is triggered, automatically closing the bidding process and preventing further submissions. Finally, the winner selection mechanism analyzes all valid bids, determining the highest bidder while considering timestamp priority for tie-breaking scenarios, ultimately completing the transparent and automated auction process with immutable results recorded on the blockchain ledger.

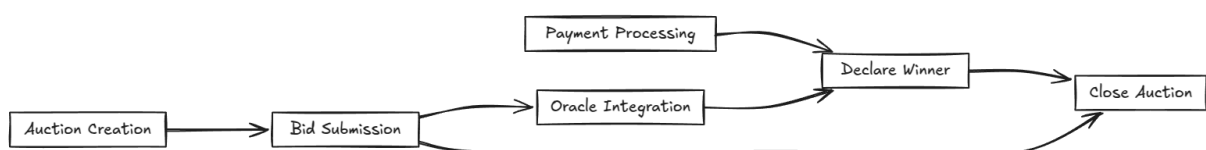


Figure 4.7: Auction contract flow

## 4.6.2 Features

### 1. Registration and Authentication System

**Description:** The registration system in our Auction app caters to four main types of organizations: Org1, Org2, Org3, and Org4. Each organization represents a different participant in the blockchain-based auction network, ensuring secure and decentralized auction management through Hyperledger Fabric integration.

**Organization Registration:** Organizations can register users through a comprehensive authentication flow that integrates with Hyperledger Fabric's Certificate Authority (CA). Each organization maintains its own wallet and identity management system, ensuring secure and isolated user credentials.

- **Org1-Org4 Users:** Users from each organization register by providing their credentials through the Fabric CA enrollment process. This ensures that only verified participants can access the auction network and perform transactions.

**Session Management:** To ensure secure and efficient session management, we utilize Hyperledger Fabric's identity and wallet system. User identities are managed through cryptographic certificates issued by the Certificate Authority, providing enterprise-grade security and blockchain-based authentication.

**Multi-Organization Support:** The system supports seamless interaction between users from different organizations, enabling cross-organizational bidding and auction participation while maintaining security boundaries.

**Comparison with Other Apps:** Unlike traditional auction applications, our system offers a comprehensive blockchain-based registration system that ensures immutable user identities and transactions. The integration with Hyperledger Fabric provides enterprise-grade security and decentralization that traditional centralized auction platforms cannot offer.

**UI Samples:** For a visual representation of the registration process, please refer to the UI sample screenshots in Appendix D figures D.1 and D.2.

### 2. Auction Management System

**Description:** The auction management system allows users to create, manage, and participate in auctions across the blockchain network. The system supports real-time bidding with automated monitoring and notifications.

**Auction Creation:** Sellers can create auctions by providing item details, descriptions, time limits, and optional picture URLs. Each auction is automatically distributed across all participating organizations in the network, ensuring transparency and accessibility.

**Auction Features:**

- **Item Management:** Sellers can specify detailed item descriptions and attach images
- **Time Limit Control:** Configurable auction duration with automatic timeout detection
- **Multi-Organization Visibility:** Auctions are visible across all network organizations
- **Status Tracking:** Real-time status updates (open/ended) with blockchain verification

**Auction Termination:** Auctions can be ended manually by sellers or automatically when time limits expire. The system ensures that auctions cannot be ended prematurely and validates all termination conditions through smart contract logic.

**Comparison with Other Apps:** Our blockchain-based auction system provides immutable auction records and transparent bidding processes that traditional auction platforms cannot guarantee. The decentralized nature ensures no single point of failure or manipulation.

**UI Samples:** For a visual representation of the auction management interface, please refer to the UI sample screenshots in Appendix D figures D.3, D.4, D.5, D.6.

### **3. Real-Time Bidding System**

**Description:** The bidding system enables secure, real-time participation in auctions with automated validation and cross-organizational endorsement.

**Bidding Process:** Users can place bids on active auctions with automatic validation through smart contracts. The system uses a two-phase commit process: initial bid submission followed by multi-organizational endorsement to ensure consensus across the network.

**Bid Features:**

- **Real-Time Validation:** Immediate bid validation through blockchain smart contracts

- **Highest Bid Tracking:** Automatic tracking and updates of current highest bids
- **Cross-Organization Endorsement:** Multi-party validation ensuring bid integrity
- **Timestamp Integration:** NTP-synchronized timestamps for accurate bid ordering

**Bid Monitoring:** The system continuously monitors all active auctions every 2 seconds, detecting bid updates and broadcasting changes to all connected clients through WebSocket connections.

**Comparison with Other Apps:** The blockchain-based bidding system ensures bid immutability and prevents manipulation, providing transparency that traditional auction platforms lack. The multi-organizational endorsement process adds an extra layer of security and consensus.

#### 4. Real-Time Notification System

**Description:** The notification system provides instant updates to users about auction events, bid changes, and auction outcomes through WebSocket connections and persistent notification storage.

**WebSocket Integration:** Real-time bidirectional communication between server and clients enables instant notifications for bid updates, auction endings, and timeouts. The system supports multiple concurrent client connections with user-specific targeting.

##### Notification Types:

- **Bid Updates:** Instant notifications when new highest bids are placed
- **Auction Endings:** Automatic notifications to winners and sellers when auctions conclude
- **Timeout Alerts:** Proactive notifications to sellers when auctions exceed time limits
- **System Events:** Real-time updates about auction status changes

**Persistent Notifications:** All notifications are stored in a PostgreSQL database with tracking for seen/unseen status, ensuring users don't miss important auction events even when offline.

**Redis Pub/Sub Integration:** The system uses Redis for scalable message broadcasting, enabling efficient real-time communication across multiple server instances and client connections.

**Comparison with Other Apps:** Our integrated notification system provides comprehensive real-time updates with persistent storage, ensuring users stay informed about all auction activities. The combination of WebSocket and database storage offers reliability that basic notification systems cannot match.

## 4.7 Conclusion

In this chapter, we provided a comprehensive overview of our implementation and the engineering decisions that shaped our solution. We detailed the software model used for project management, the system requirements derived from our problem definition, and the reasoning behind the tools and technologies we selected. We addressed core questions and challenges, explained our architectural choices, and walked through the algorithms and protocols we developed. Overall, this chapter presents the technical foundation of our project and highlights the key steps we took to bring our system to life.



## **Chapter 5**

### **Results and Discussion**

## 5.1 Introduction

Evaluating the performance of blockchain-based systems is crucial for understanding their real-world behavior, especially under different workload conditions. In this phase of our project, we carried out a series of experiments to assess both the performance and throughput of the blockchain layer. Additionally, we tested the NTP servers to evaluate the reliability and uniqueness of the timestamps retrieved from them.

## 5.2 Results

### 5.2.1 Hyperledger Explorer[37]

As part of the implemented solution, we integrated **Hyperledger Explorer**, an essential blockchain visualization tool that offers real-time monitoring and comprehensive insights into the state of the network. Through this integration, we were able to:

- **Monitor Live Transactions:** Track all submitted transactions across the network in real time, ensuring transparency and traceability of each auction bid.
- **Inspect Block Details:** Access detailed information for every block, including block height, transaction count, hash values, and timestamps, providing a complete audit trail for system operations.
- **Analyze Network Topology:** View an overview of all active peers, ordering nodes, and deployed chaincodes (smart contracts), enabling verification of network components and configurations.

This setup significantly enhances the transparency and accountability of our auction system. It allows authorized participants to independently verify system operations, ensures that all bids are securely and immutably recorded on the blockchain, and eliminates the possibility of tampering or unauthorized data manipulation. As a result, the integrity and trustworthiness of the auction process are fully maintained.



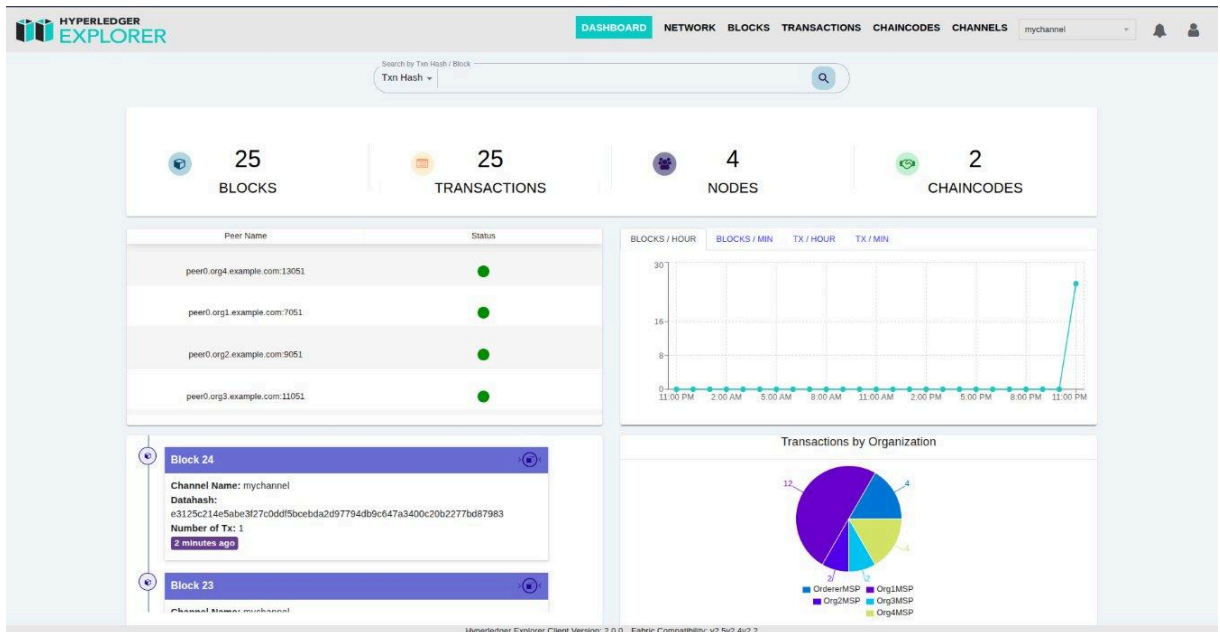


Figure 4.8: Hyperledger Fabric Explorer

### 5.2.2 NTP servers Testing

To evaluate the reliability and uniqueness of timestamps retrieved from the NTP servers, we conducted a dedicated performance test simulating concurrent requests. The test was performed using **JMeter**, configured with **200 concurrent threads**, a **40-second ramp-up period**, and a total test duration of **10 minutes**. During this period, the system retrieved **2269 timestamps** from the NTP servers.

The results demonstrated the precision and resolution of the timestamping process. Across multiple consecutive retrievals, the time differences between successive timestamps ranged from approximately **118 microseconds (118,082 ns)** to **621 microseconds (621,054 ns)**.

The test confirms that, even under significant load with 200 parallel users, the NTP servers consistently provide high-resolution timestamps with minimal variance, ensuring uniqueness and suitability for applications requiring precise time synchronization.

### 5.2.3 Performance Testing

Conducted using JMeter to evaluate system performance under load, Invoked Operations are in those javascript files that connect to the Hyperledger network:

- **Bid.js** and **SubmitBid.js**: perform some bids (include read and write operations).

- **QueryAllBids.js**: Retrieves all the bids for a certain auction.
- **TestWriteData.js**: Create an object and save it in the state (ledger) of the blockchain.

Our Testing environment were conducted on a single machine using the Hyperledger network including the following:

- 4 Organizations.
- 4 Peer nodes and 1 Orderer node.
- 2 Chain codes deployed on each peer node.

Blockchain configurations have several parameters that could be configured and tuned such as:

- Max Message Count: Controls how many transactions are grouped into a block.
- Block Timeout: Defines how long the system waits before generating a block, even if it's not full.
- Block Size: Determines the maximum block size in bytes.

Testing our baseline performance which incorporates writing a simple block into the ledger of the block chain.

Table 5.2.2.1 Results of baseline tests

Metrics	TestWriteData.js
# Samples	1,513
Avg. Response Time	19, 655 ms
Error Rate	0.00 %
Throughput	4.9 / sec

Table 5.2.2.2 Results of creating and submitting bids

Metrics	Bid.js	SubmitBid.js	Total
# Samples	750	702	1452
Avg. Response Time	20,381 ms	20,945 ms	20,653 ms
Error Rate	0.00 %	0.00 %	0.00 %
Throughput	2.4 / sec	2.2 / sec	4.6 /sec

Table 5.2.2.3 Results of querying bids

Metrics	QueryAllBids.js
# Samples	18,999
Avg. Response Time	1, 523 ms
Received KBs	34.84 KB / sec
Error Rate	0.00 %
Throughput	63.3 / sec

## 5.2.4 Results Discussion

- Write Operations Performance:
  - Files TestWriteData.js, Bid.js, and SubmitBid.js showed high response time on average; these operations involve writing data to the blockchain ledger, which naturally introduces delays due to consensus mechanisms and state validation.
  - The Throughput for write operations remains low, around 2.2 to 2.4 operations per second, reflecting the expected overhead in write transactions.
- Read Operations Performance:
  - QueryAllBids.js, responsible for retrieving all bids, demonstrated significantly better performance with an average response time of 1.5 seconds.
  - The Throughput for read operations are higher than write operations, around 63.3 operations per second, indicating that read-heavy workloads are processed efficiently compared to write operations.
- The system exhibits acceptable performance for read-intensive operations, but write operations remain a bottleneck, typical for permissioned blockchain networks like Hyperledger Fabric.
- These results establish a baseline for evaluating any subsequent improvements or optimizations to the system.

### **5.2.5 Recommendations**

- Block Size Tuning by adjusting batch size and batch timeout for better batching and reduced block generation delay.
- Network Configuration Tuning by increasing the number of Orderer Nodes for better fault tolerance and performance (subject to hardware availability).
- Testing on high-performance machines recommended for accurate production-grade benchmarking.

## **5.3 conclusion**

In this chapter, we looked at the results of testing the system. We discussed both throughput and latency, and explored ways to improve them. We also explained the need to run the system on a more capable machine, and why the current numbers didn't meet our expectations.

## **Chapter 6**

### **Conclusion and future work**

## 6.1 Introduction

In the previous chapters, the project details were discussed thoroughly. The motivation for the project, the scientific background needed, the literature review, the need to extend the related work, the scope of work, the requirements and design, the followed development process, the implementation details and results all are discussed. In this chapter, the Conclusion, contribution of the project, and the proposed future work will be discussed.

## 6.2 Project conclusion

In this project, we developed *BitAuction*, a decentralized auction platform built on top of blockchain technology, aiming to introduce transparency, trust, and automation into the bidding process. From the initial design to the full implementation, we tackled key challenges in building a secure, tamper-proof system while maintaining a user-friendly experience for both bidders and sellers.

In conclusion, *BitAuction* demonstrates how blockchain can be effectively utilized to reshape traditional auction systems. By integrating smart contracts, distributed consensus, and real-time bid tracking, the platform ensures fairness and integrity without reliance on a central authority. This project not only showcases the potential of decentralized applications (dApps) but also serves as a scalable blueprint for future systems that require high levels of trust and automation.

## 6.3 Contribution of the project

After going through all the phases of this project, in the following section, a discussion of the project's contribution will be detailed.

- **Open Source Availability:** The entire system is made publicly accessible on GitHub to promote transparency, collaboration, and further research.
- **Adopted SPI Model:** The system was developed using the Software Process Improvement (SPI) model, ensuring a systematic and iterative development process.
- **Implemented Open-Cry Auction Model:** A transparent and competitive auction model was adopted, allowing all participants to view and respond to real-time bids.
- **Live Auction Interface:** Developed a dynamic, real-time interface that allows users to observe and participate in live auctions seamlessly.

- **User-Centric UI Design:** Designed an intuitive and accessible user interface to enhance usability and engagement across a wide user base.
- **Time Tie-Breaker Protocol:** Implemented a robust tie-breaking mechanism based on synchronized timestamps from multiple Network Time Protocol (NTP) servers to ensure fairness and accuracy.

## 6.4 Future Work

Future work was initially discussed and outlined in Section 3.5 during our research. The following points were later identified during the implementation phase.

- **Adopt Network Time Security (NTS)** instead of NTP to obtain cryptographically secure and tamper-resistant timestamps for submitted bids, enhancing the integrity of the tie-breaking mechanism.
- **Fraud Detection and Attack Simulation:** Conduct comprehensive tests to evaluate the system's resistance to fraudulent activities. This will require implementing digital currency mechanisms and payment flows, which were beyond the scope of the current version but are essential for enabling realistic fraud scenarios.
- **Consensus Algorithm Benchmarking:** Perform a detailed comparison between our current consensus mechanism and more resource-intensive alternatives such as Proof of Work (PoW), with a focus on latency, energy efficiency, and scalability under varying workloads.
- **Incorporate Cristian's Algorithm:** Integrate Cristian's algorithm into the time synchronization module to estimate and account for network latency when retrieving trusted timestamps from NTP servers, improving fairness and accuracy.
- **Dynamic Time Limit Extension:** Enhance the auction logic by allowing automatic time limit extensions when bids are placed shortly before the auction's scheduled end, ensuring fair competition and mitigating the effects of last-second bidding.
- **Decentralization of Backend Responsibilities:**
  - Currently, the backend acts as a trusted intermediary between clients and the blockchain. This introduces a centralized component into an otherwise decentralized architecture. A significant area for improvement is eliminating this trust dependency by enabling clients to directly interact with the blockchain network. This would involve architectural changes to support

features like live auctions and real-time notifications in a decentralized manner, potentially through decentralized pub/sub mechanisms or client-side SDKs with cryptographic verification.

- **Exploring RAFT-based Ordering:**

- The current design uses a custom time service to support ordering in the blockchain network. A promising alternative is the use of the RAFT consensus protocol for ordering services. RAFT provides crash fault tolerance and strong consistency, which may align better with production-grade blockchain networks. However, one issue that must be considered is that RAFT determines block creation based on the orderer's local view of time and incoming transactions. This can lead to fairness issues: a client may send a transaction earlier, but due to network delays, it might be included in a later block than a transaction that arrived later but experienced less delay. Future work should explore mitigations for this issue beside timestamp-based transaction sorting.



## References

- [1]: Shi, Z., De Laat, C., Grosso, P., & Zhao, Z. (2022). Integration of blockchain and auction models: a survey, some applications, and challenges. *IEEE Communications Surveys & Tutorials*, 25(1), 497–537. <https://doi.org/10.1109/comst.2022.3222403>
- [2]: Chiquito, E., Bodin, U., & Schelén, O. (2023). Survey on decentralized auctioning systems. *IEEE Access*, 11, 51673–51688. <https://doi.org/10.1109/access.2023.3279914>
- [3]: Chiquito, E., Bodin, U., Schelén, O., & Monrat, A. a. A. (2024). Digitalized and Decentralized Open-Cry Auctioning: key properties, solution design, and implementation. *IEEE Access*, 12, 64686–64700. <https://doi.org/10.1109/access.2024.3395791>
- [4]: Wust, K. and Gervais, A. (2018) “Do you need a blockchain?”, 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), pp. 45–54. doi:10.1109/cvcbt.2018.00011.
- [5]: Inpher, Inc. (2024b, February 20). What is Secure Multiparty Computation? - SMPC/MPC Explained | Inpher. <https://inpher.io/technology/what-is-secure-multiparty-computation/>
- [6]: Li, Honglei, and Weilian Xue. "A Blockchain-Based Sealed-Bid e-Auction Scheme with Smart Contract and Zero-Knowledge Proof." *Security and Communication Networks* 2021 (2021): Article ID 5523394. <https://doi.org/10.1155/2021/5523394>.
- [7]: Tan, S., & Heng, S. (2022). Secure cryptographic E-Auction system. *International Journal of Technology*, 13(6), 1222. <https://doi.org/10.14716/ijtech.v13i6.5827>
- [8]: Omar, I. A., Hasan, H. R., Jayaraman, R., Salah, K., & Omar, M. (2021). Implementing decentralized auctions using blockchain smart contracts. *Technological Forecasting and Social Change*, 168, 120786. <https://doi.org/10.1016/j.techfore.2021.120786>
- [9]: Qusa, H., Tarazi, J., & Akre, V. (2020, February). Secure e-auction system using blockchain: UAE case study. In 2020 Advances in Science and Engineering Technology International Conferences (ASET) (pp. 1-5). IEEE.
- [10]: Minaei, M., Le, D. V., Kumaresan, R., Beams, A., Moreno-Sanchez, P., Yang, Y., ... & Zamani, M. (2023). Scalable Off-Chain Auctions. *Cryptology ePrint Archive*.
- [11]: H. S. Galal and A. M. Youssef, “Verifiable sealed-bid auction on the Ethereum blockchain,” in *Proc. Int. Conf. FC, Nieuwpoort, Curaçao*, Feb. 2018, pp. 265–278.
- [12]: Chow, V. (2020). Predicting auction price of vehicle license plate with deep recurrent neural network. *Expert Systems with Applications*, 142, 113008.
- [13]: "0xAuction", [Online]. Available, link: Apostrophe-Corp/0xAuction: An NFT market place with real time onchain notifications, [Accessed: 12/1/2024]

- [14] "Opensea," [Online]. Available: <https://opensea.io/>, [Accessed: 12/1/2023].
- [15] "Rarible," [Online]. Available: <https://rarible.com/>, [Accessed: 12/1/2023].
- [16] "stargaze," [Online]. Available: <https://github.com/public-awesome/stargaze>, [Accessed: 12/1/2023].
- [17] "Liveauctioneers," [Online]. Available: LiveAuctioneers: Online Auctions for Arts, Antiques & Collectibles, [Accessed: 12/1/2023].
- [18]: Pop, C., Prata, M., Antal, M., Cioara, T., Anghel, I., & Salomie, I. (2020). An Ethereum-based implementation of English, Dutch, and First-price sealed-bid auctions. 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), <https://doi.org/10.1109/ICCP51029.2020.9266180>
- [19]: Gorenflo, C., Lee, S., Golab, L., & Keshav, S. (2020). FastFabric: Scaling hyperledger fabric to 20 000 transactions per second. International Journal of Network Management, 30(5), e2099.
- [20]: Proxies - OpenZeppelin Docs. (n.d.). <https://docs.openzeppelin.com/contracts/4.x/api/proxy>
- [21]: J. B. Awotunde, R. O. Ogundokun, R. G. Jimoh, S. Misra, and T. O. Aro, "Machine learning algorithm for cryptocurrencies price prediction," in Artificial Intelligence for Cyber Security: Methods, Issues and Possible Horizons or Opportunities. Cham, Switzerland: Springer, 2021, pp. 421–447.
- [22]: Software Engineering Competence Center (SECC). SPIG Product Suite Handbook V1.2, Software Process Improvement Guide (SPIG). Information Technology Industry Development Agency (ITIDA), Ministry of Communications and Information Technology (MCIT), Giza, Egypt, 2010.
- [23]: React.js official documentation. <https://reactjs.org/>, [Accessed: 7/02/2025].
- [24]: Wikipedia contributors. React(JavaScript library). [https://en.wikipedia.org/wiki/React\\_\(JavaScript\\_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library)), [Accessed: 7/02/2025].
- [25]: NestJS official documentation. <https://docs.nestjs.com/>, [Accessed: 7/02/2025].
- [26]: Wikipedia contributors. NestJS. <https://en.wikipedia.org/wiki/NestJS>, [Accessed: 7/02/2025].
- [27]: PostgreSQL official documentation. <https://www.postgresql.org/docs/>, [Accessed: 7/02/2025].
- [28]: Wikipedia contributors. PostgreSQL. <https://en.wikipedia.org/wiki/PostgreSQL>, [Accessed: 7/02/2025].

- [29]: Redis official documentation. <https://redis.io/docs/>, [Accessed: 7/02/2025].
- [30]: Wikipedia contributors. Redis. <https://en.wikipedia.org/wiki/Redis>, [Accessed: 7/02/2025].
- [31]: MDN Web Docs. WebSockets. [https://developer.mozilla.org/en-US/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API), [Accessed: 7/02/2025].
- [32]: Wikipedia contributors. WebSocket. <https://en.wikipedia.org/wiki/WebSocket>, [Accessed: 7/02/2025].
- [33]: Hyperledger fabric documentation. <https://hyperledger-fabric.readthedocs.io/en/release-2.5/>, [Accessed: 7/02/2025].
- [34]: Amazon Web Services. What is hyperledger fabric?, [Accessed: 7/02/2025].
- [35]: Git documentation. <https://git-scm.com/doc>, [Accessed: 7/02/2025].
- [36]: Wikipedia contributors. Git. <https://en.wikipedia.org/wiki/Git>, [Accessed: 7/02/2025].
- [37]: Hyperledger Explorer documentation. <https://blockchain-explorer.readthedocs.io/en/main/introduction.html>, [Accessed: 7/02/2025].

## **Appendix A**

### **Summary of BitAuction Research Papers**

## A.1 Summary of [1]

**Title:** Integration of Blockchain and Auction Models: A Survey, Some Applications, and Challenges

*Engineering Research Paper*

*Question–Answer Form*

**Authors:** Zeshun Shi, Cees de Laat, Paola Grosso, and Zhiming Zhao

**Published in:** 16 November 2022

### **What is your take-away message from this paper?**

This paper highlights the synergy between blockchain and auction models, emphasizing how blockchain's decentralization, transparency, and security can improve auction processes, and how auction models can enhance blockchain's operational efficiency. The paper comprehensively surveys existing blockchain-auction integration efforts, outlines applications, and identifies challenges and future directions in this field.

**What is the motivation for this work (both people problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

#### **Motivation:**

- Centralized auctions suffer from high costs, inflexibility, trust issues, and single points of failure.
- Blockchain systems face challenges in scalability, miner incentives, transaction fee mechanisms, and resource allocation.

**Research Question:**

- How can blockchain and auction models be effectively integrated to address these issues?

**Why non-trivial?**

- Centralized systems are inherently opaque and insecure, while blockchain-based systems require complex incentive mechanisms and scalable solutions.

**Previous Solutions and Gaps:**

- Centralized auction platforms were efficient but lacked transparency and fairness.
- Blockchain systems addressed decentralization but were not optimized for specific auction processes.

**What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work?****How does it represent an improvement? How is the solution achieved?****Proposed Solution:**

- A two-way integration where blockchain provides a trustworthy platform for auctions, and auctions enhance blockchain's mechanisms (e.g., miner selection, transaction fee models).

**Why It Will Work:**

- Blockchain ensures secure, immutable, and automated auctions. Auction models introduce dynamic pricing and allocation for blockchain incentives.

**Improvements:**

- Reduces intermediary costs, increases transparency, and creates efficient incentive mechanisms.

**Implementation:**

- Use of smart contracts for auction automation, blockchain for decentralized transaction validation, and custom auction models for blockchain-based systems.

**What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?****Evaluation:**

- The paper reviews various blockchain-based auction applications, such as energy trading, wireless communication, and service allocation.

- It provides experimental results (e.g., simulation studies) demonstrating efficiency and fairness in blockchain-enabled auction systems.
- The authors discuss the practicality of integrating auction models into blockchain layers like incentive mechanisms and consensus protocols.

**What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

**Good Idea?**

- Yes, the integration leverages the strengths of both blockchain and auctions.

**Flaws:**

- Limited exploration of challenges in scaling auction-based solutions for large blockchain networks.
- Some proposed methods rely on idealized assumptions about user behavior and system capabilities.

**What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

**Authors' Contributions:**

- A comprehensive survey of blockchain and auction model integration.
- Taxonomy and classification of applications.
- Identification of open research challenges and future directions.

**My Opinion:**

- The taxonomy provides valuable insights for categorizing use cases. The emphasis on real-world applications is a significant strength.

**What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?****Authors' Directions:**

- Enhancing scalability and privacy of blockchain-auction systems.
- Developing more efficient consensus mechanisms using auction models.

**My Suggestions:**

- Investigating hybrid blockchain architectures for auctions.
- Exploring machine learning integration for adaptive auction strategies.

**What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

- How can blockchain scalability be improved for handling large-scale auction activities?
- What are the trade-offs between privacy and transparency in blockchain auction systems?
- Can auction-based models be standardized for integration across different blockchain platforms?



## A.2 Summary of [2]

**Title:** Survey on Decentralized Auctioning Systems [2]

*Engineering Research Paper*

*Question–Answer Form*

**Authors:** Eric Chiquito et al.

**Published in:** May 25th, 2023

### **What is your take-away message from this paper?**

The take-away message from this survey is that decentralized auctioning systems, especially blockchain-based ones, offer promising alternatives to traditional centralized auction models. These systems aim to address critical concerns such as trust, transparency, and security by distributing control and removing the need for third-party authorities. However, they also present new scalability, privacy, and conflict resolution challenges. The survey underscores that while blockchain technologies are valuable for creating decentralized, tamper-proof records, non-blockchain approaches like distributed hash tables and graph-based systems offer additional flexibility, though often at the cost of weaker consistency and reliability. Ultimately, the survey advocates a careful balance between decentralization and essential functional and quality requirements to create secure, fair, and scalable auction platforms.

### **What is the motivation for this work (both people problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

**Motivation:** The motivation for this research stems from two interconnected issues *people's trust* and *technical limitations* in centralized auction systems. Current online auction platforms, such as eBay, rely on a centralized authority to validate transactions and manage disputes. This reliance leads to several trust-related problems: participants may question the fairness and security of transactions, especially since centralized entities can create high transaction fees and potentially misuse participant data. Additionally, technical limitations such as single points of failure, limited transparency, and vulnerability to cyber-attacks pose significant challenges to the reliability and efficiency of these systems.

**Distillation into Research Question:** The core question becomes, “How can decentralized systems enable fair, transparent, and trustless auctions without a central authority while still providing scalability, security, and effective dispute resolution?”

**Non-trivial Solution:** The people problem lacks a trivial solution because eliminating a central authority requires developing a trustless environment in which participants can be confident of transaction integrity without a governing body. Without a central administrator to resolve disputes, ensure fairness, or authenticate users, decentralized auctions must find innovative ways to establish and maintain trust among participants.

**Previous Solutions and Their Shortcomings:** Prior solutions have included cryptographic mechanisms, peer-to-peer (P2P) systems, and basic blockchain models. Cryptographic solutions safeguard transaction confidentiality but are computationally expensive, while P2P networks distribute control yet struggle with maintaining data consistency and security. Blockchain is limited by scalability and long transaction times in permissionless (public) models and has partial decentralization in permissioned (private) models.

**What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

**Proposed Solution:** The authors propose a decentralized auction system model that leverages blockchain technologies as well as non-blockchain methods like distributed hash tables (DHT) and graph-based systems. By distributing trust across multiple nodes, blockchain ensures data immutability, transparency, and prevents tampering. Smart contracts are incorporated to automate and enforce auction rules, ensuring fair transaction handling without human intervention.

**Why It's Believed to Work:** Blockchain's transparency, immutability, and decentralized nature are key reasons for its suitability in auction settings. By removing the need for a third-party intermediary, blockchain can reduce transaction fees and increase security. The proposed model represents an improvement by providing a secure, tamper-proof system for tracking bids and finalizing transactions, and using smart contracts to automate conflict-free bidding processes.

**Improvements Over Previous Models:** This solution improves traditional and prior decentralized models by removing a single point of failure, increasing transparency, and preventing fraud through consensus mechanisms. Moreover, integrating smart contracts enables secure and autonomous dispute resolution within the system.

**How the Solution is Achieved:** The system is designed to operate across decentralized nodes, with blockchain technology serving as the core for recording transactions and maintaining audit trails. Additional mechanisms, like off-chain processing, reduce computational overhead, and distributed hash tables facilitate efficient data lookup. Together, these elements create a more robust, trustless system that can handle complex auction scenarios.

**What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

**Author's Evaluation:** The authors evaluate the solution by comparing various decentralized auction systems, focusing on how well they meet key requirements like correctness, fairness, confidentiality, and scalability. They argue that blockchain-based systems, due to their secure consensus protocols, provide strong foundations for decentralized auctions. However, they note that blockchain is less suited for real-time transactions, particularly open-bid auctions that require fast bid validation.

**Supporting Evidence:** The survey provides evidence through a detailed analysis of each approach's strengths and weaknesses, highlighting real-world applications and experimental data where available. For instance, permissioned blockchains are shown to offer improved scalability but with limited decentralization, while permissionless blockchains enhance transparency but are resource-intensive and slow. The authors also discuss experimental systems that employ smart contracts for managing auction rules autonomously.

**What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

**Analysis:** The idea of a decentralized auction system is compelling, particularly as it addresses trust issues inherent in centralized models. The use of blockchain for transparency is a strong approach, as is the inclusion of smart contracts to automate processes and reduce human intervention.

**Flaws:** One flaw in the approach is the scalability limitation, especially with permissionless (public) blockchain networks, where transaction speeds and costs increase with the number of participants. Privacy is also a concern since blockchain's transparent nature could compromise bid confidentiality. Additionally, dispute resolution in physical asset transactions may still require centralized arbitration, which challenges the decentralized model.

**Interesting/Controversial Ideas:** The most intriguing aspect is the balance between decentralization and the need for some level of centralized control for adjudicating disputes, especially when physical assets are involved. The concept of off-chain transactions to improve scalability, combined with on-chain records for auditability, represents a practical compromise that may prompt further discussion and development.

**What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

**Contributions:** The paper makes significant contributions by:

- **Identifying Functional Needs:** It defines the essential features and quality requirements for decentralized auctions, such as fairness, confidentiality, and scalability.
- **Comparing Decentralized Models:** It thoroughly compares blockchain and non-blockchain approaches, illustrating each model's strengths and limitations.
- **Highlighting Knowledge Gaps:** The authors identify gaps, such as bid serialization and the lack of efficient, decentralized conflict resolution methods.
- **Proposing Hybrid Approaches:** The paper suggests integrating blockchain with off-chain mechanisms to improve efficiency, which could inspire future research.

In my opinion, the paper provides a well-rounded foundation for developing decentralized auctions by consolidating key ideas and paving the way for hybrid solutions.

**What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

**Future Directions:**

- **Hybrid Solutions:** Future research could explore hybrid models that combine on-chain and off-chain methods to enhance both scalability and privacy.
- **Advanced Smart Contracts:** Enhancing smart contracts with legal language (e.g., Ricardian contracts) could make them more applicable to real-world disputes, especially for physical assets.
- **Efficient Consensus Mechanisms:** Developing consensus protocols that maintain security while supporting real-time transactions could address performance limitations.
- **Improved User Authentication:** Exploring methods for decentralized user verification that do not compromise privacy would support broader application to various types of auctions.

**What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

**Remaining Questions:**

1. How can decentralized auction systems manage both privacy and transparency effectively in competitive environments?
2. What legal standards or frameworks are required to enforce smart contract-based auctions, especially for physical assets?
3. How might a decentralized system verify user identities without compromising the core principle of privacy?
4. How can blockchain systems handle dynamic, high-frequency bidding without causing delays or escalating costs?

### A.3 Summary of [3]

**Title:** Digitalized and Decentralized Open-Cry Auctioning: Key Properties, Solution Design, and Implementation [3]

**Authors:** Chiquito, E et al.

**Published in:** 01 May 2024

*Engineering Research Paper  
Question–Answer Form*

#### **What is your take-away message from this paper?**

The paper focuses on designing and implementing a decentralized open-cry auction system leveraging blockchain technology. Its primary contributions include identifying key quality attributes like verifiability, transaction immutability, time synchronization, and transaction ordering. It proposes a proof-of-concept (PoC) using Hyperledger Fabric, addressing gaps in existing blockchain systems, particularly time synchronization. The integration of an external API for trusted timestamps is highlighted as a key innovation to ensure fairness and transparency.

**What is the motivation for this work (both people problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

#### **People Problem**

Centralized auction platforms (e.g., eBay, Tradera) require trust in a single authority for transaction validity and order. This dependency raises concerns about fairness, trust, and transparency, as central entities may influence outcomes maliciously or inadvertently.

#### **Technical Problem**

Open-cry auctions in decentralized environments face challenges like transaction ordering, time synchronization, and fairness. Blockchain's inherent design limitations—such as lack of native time synchronization—compound these issues.

### **Why the Problem Is Hard**

1. **Transaction Ordering:** Without a centralized coordinator, achieving consensus on the order of bids is non-trivial.
2. **Time Sensitivity:** Accurate bid timestamps are critical, especially for auctions with dynamic deadlines or concurrent transactions.
3. **Scalability:** Handling multiple auctions concurrently without compromising latency and performance remains a challenge.
4. **Blockchain Trilemma:** You cannot achieve decentralization, scalability, and security at the same time.

### **Previous Solutions**

- Focus on sealed-bid auctions, where transaction ordering and time synchronization are less critical.
- Use of centralized services for time management, which compromises decentralization.

**What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work?**

**How does it represent an improvement? How is the solution achieved?**

### **Design Overview**

1. **Blockchain:** Hyperledger Fabric provides verifiability and immutability.
2. **External Time API:** Ensures accurate timestamps using NTP servers.
3. **Smart Contracts:** Automate auction rules, ensuring fairness and consistent execution.

### **Why It Works**

- Integrating trusted timestamps mitigates synchronization issues without relying on a centralized entity.
- Hyperledger Fabric's modularity allows handling auction-specific requirements effectively.
- Smart contracts ensure rule adherence, reducing human error and manipulation risks.

## **Improvements**

This approach minimizes transaction processing delays and ensures fair ordering and validation of bids, addressing critical gaps in prior systems.

**What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

The authors validate their approach through:

1. **Performance Tests:** Analyzing the latency of transactions and time synchronization.
2. **Scalability Experiments:** Evaluating system performance under varying numbers of participants and transactions.

**What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

## **Strengths**

- **Innovative Time Management:** Trusted timestamps enhance fairness in time-sensitive scenarios.
- **Scalability:** could be improved by incorporating offchain solutions and/or distributed timestamp management.
- **Modular Design:** Adaptable to different use cases and blockchain systems.

## **Weaknesses**

- **Scalability Limitations:** Hyperledger Fabric's architecture struggles with high concurrency and global transaction ordering.
- **Dependency on External API:** Centralized time services introduce a potential point of failure, compromising decentralization and fairness.

## **Interesting/Controversial Ideas**

- Fairness is compromised by introducing a central time management entity.
- Dynamic auction deadlines extend fairness but challenge time-sensitive consensus in decentralized systems.
- Permissioned blockchains strike a balance between privacy and transparency, but their trade-offs (e.g., centralization) merit further exploration.



**What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

- **Key Properties Identification:** Defines essential attributes for decentralized open-cry auctions.
- **Proof-of-Concept Implementation:** Demonstrates a solution to time synchronization.
- **Scalability Insights:** Highlights architectural limitations
- **Performance Analysis:** Benchmarks critical parameters like transaction latency and synchronization delays.

**What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

- Explore alternatives to the external API for time synchronization.
- Enhance scalability by optimizing transaction serialization mechanisms. OR by using off-chain solutions.
- Explore decentralized time synchronization methods to replace the external API.

**What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

- Can the external time API be replaced with a fully decentralized mechanism?
- How does Hyperledger Fabric compare to R3 Corda or Ethereum for similar use cases in terms of cost and efficiency?

## A.6 Summary of [6]

**Title:** A Blockchain-Based Sealed-Bid e-Auction Scheme with SmartContract and Zero-Knowledge Proof

*Engineering Research Paper  
Question–Answer Form*

**Authors:** Honglei Li et al.

**Published in:** 19 May 2021

### **What is your take-away message from this paper?**

The paper introduces a blockchain-based sealed-bid e-auction scheme incorporating smart contracts and zero-knowledge proofs. It aims to enhance the security, privacy, and fairness of the bidding process, especially by eliminating the need for a third-party auctioneer, thus addressing common privacy and trust issues in e-auction systems.

### **What is the motivation for this work (both people problem and technical problem), and its distillation into a research question?**

#### **The paper addresses two main issues:**

1. **People Problem:** Traditional e-auctions are vulnerable to privacy breaches, unfairness, and trust issues, especially due to reliance on third-party auctioneers, which may lead to bid tampering or price leakage.
2. **Technical Problem:** The existing approaches, even those using blockchain, often rely on centralized entities or lack sufficient security, privacy, or efficiency.

### **Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

The research question explores how blockchain and cryptographic tools (e.g., smart contracts and zero-knowledge proofs) can enable a fully decentralized, secure, and verifiable sealed-bid e-auction.

### **What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

The authors propose a sealed-bid e-auction scheme using blockchain, smart contracts, Pedersen commitments, and Bulletproofs zero-knowledge proofs:

- **Idea:** Use blockchain for decentralization and smart contracts to automate bidding processes. Pedersen commitments and Bulletproofs ensure bid confidentiality and verifiable outcomes without revealing private bid details.
- **Improvement:** This method eliminates the third-party auctioneer, securing data and reducing transaction costs.
- **Implementation:** Bidders commit their bid prices with Pedersen commitments stored on the blockchain, which are later verified without exposing other bid prices.

**What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

The authors validate their solution through simulation on an Ethereum-like platform, comparing it against other blockchain-based e-auction schemes. Key findings include that the scheme supports decentralized verification, anonymity, and effective bid validation while being less costly due to the absence of a third-party auctioneer. Performance analysis showed that processing time scaled with the number of participants, highlighting potential scalability limitations for large-scale auctions.

**What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

This is a strong approach to resolving long-standing e-auction challenges, notably by removing the auctioneer role entirely. Potential flaws include scalability concerns for large auctions and high computational demands in the verification phase, which could impact user adoption in real-world applications. The decentralized, privacy-focused verification and use of zero-knowledge proofs are compelling yet could generate debate about performance in high-volume environments.

**What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

- **Ideas and Methods:** The paper presents a decentralized auction model for sealed-bid e-auctions, which is innovative in eliminating the need for a third-party auctioneer.

The authors achieve this by combining Pedersen commitments and Bulletproofs zero-knowledge proofs (ZKPs):

- Pedersen commitments ensure that each bid remains private and unchangeable once submitted, protecting the confidentiality of bid amounts throughout the auction process.
- Bulletproofs ZKP enables bidders to verify the validity of the winning bid without revealing other bid amounts, enhancing privacy while providing transparency and fairness in bid verification.
- **Experimental Results:** Demonstrated feasibility and performance of auction processes without third-party control.
- **Software Implementation:** A smart contract prototype was developed and tested on an Ethereum-like platform, demonstrating the practical viability of the system. The smart contract automates the auction steps, including registration, bid commitment, bid opening, verification, and final settlement, ensuring that each phase operates transparently and without the need for centralized control. This prototype shows potential for real-world deployment on blockchain networks like Ethereum.

**What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

The authors suggest optimizing the verification and finish phases to improve scalability. They also plan to test with other blockchain technologies beyond Ethereum to reduce computational requirements. Exploring other cryptographic protocols that balance efficiency and security could further enhance the scheme's applicability in real-world, large-scale auctions (Security and Communication...).

**What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

- How will performance be affected in a large-scale, public blockchain environment?
- What are the implications of gas costs on the auction's feasibility for regular users in a real Ethereum environment?

## A.7 Summary of [7]

**Title:** Secure Cryptographic E-Auction System

*Engineering Research Paper  
Question–Answer Form*

**Authors:** Soo-Chin Tan et al.

**Published in:** 3 November 2022

### **What is your take-away message from this paper?**

integrating asymmetric encryption (RSA) and digital signatures provides a robust and secure foundation for e-auction systems. By encrypting bids with the auctioneer's public key, the system ensures confidentiality, while digital signatures verify the integrity of bids. This approach eliminates reliance on third parties, addressing privacy, fairness, and trust concerns in e-auctions. However, while the system is effective for small to medium-scale auctions, future research is needed to improve scalability, efficiency, and adaptability to emerging cryptographic challenges.

**What is the motivation for this work (both people problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

#### **People Problem:**

- Trust in e-auctions is undermined by privacy, confidentiality, and fairness concerns.

#### **Technical Problem:**

- Existing systems rely on third parties or have vulnerabilities like bid tampering and inefficiency.

#### **Research Question:**

- How to ensure privacy, security, and fairness in e-auctions without relying on third-party entities?

#### **Why not trivial?**

- Balancing security, scalability, and performance is challenging.

**What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

**Core Idea:**

- Use RSA encryption and digital signatures for secure bid submission and verification
- Bidders encrypt bids using the auctioneer's public key, ensuring only the auctioneer can decrypt them with their private key.
- Bids are signed by bidders to ensure integrity.

**Improvements:**

- Eliminates reliance on third parties, secures confidentiality, and enhances fairness.

**What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

**Security Properties:**

- Achieves anonymity, confidentiality, integrity, and fairness.

**Performance:**

- Efficient for small to medium scales (e.g., decryption and verification take ~2580ms).

**Validation:**

- A Java-based prototype demonstrated compliance with security goals.

**What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

**Strengths:**

- Simple, secure, and removes third-party dependency.

**Flaws:**

- Scalability and efficiency challenges for large-scale auctions.

**Interesting Ideas:**

- Using the auctioneer's public key to encrypt bids ensures confidentiality without needing a third-party arbiter, simplifying the trust model.

**Practical Implications:**

- Practical for small to medium-scale applications, particularly where third-party involvement is undesirable. Wider adoption may require further optimizations.

**What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

**Ideas:**

- Integrates RSA encryption and digital signatures in e-auctions.

**Methods:**

- Clear algorithms for secure auction phases.

**Artifacts:**

- Prototype implementation with validated performance.

**What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

- Explore post-quantum cryptography for resilience.
- Optimize performance for large-scale auctions.
- Investigate decentralized decryption mechanisms (e.g., multi-party computation).

**What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

- How does the system handle high-volume?
- What is the cost of scaling the system for larger auctions?

## A.8 Summary of [8]

**Title:** Implementing decentralized auctions using blockchain smart contracts. Technological Forecasting and Social Change

*Engineering Research Paper  
Question–Answer Form*

**Authors:** Omar, I. A., Hasan, H. R., Jayaraman

**Published in:** July 2021

### What is your take-away message from this paper?

- The primary takeaway from this paper is that using blockchain technology, specifically Ethereum smart contracts, can enhance online auctions by enabling a decentralized, transparent, and secure auction system. Traditional online auction platforms rely on central intermediaries, which may lead to issues like data manipulation, privacy concerns, and additional transaction fees. In contrast, a blockchain-based solution eliminates intermediaries, ensures data integrity, reduces costs, and maintains transparency and traceability throughout the auction process.
- The paper presents an Ethereum blockchain-based solution which captures interactions between auctioneers and bidders using an Ethereum smart contract, decentralized storage, and trusted oracles to ensure data integrity, transparency and elimination of intermediaries, also the paper presents algorithms that define the working principles of the proposed blockchain solution .



**What is the motivation for this work (both people problem and technical problem), and its distillation into a research question?**

**1. People Problem:**

- a. People have suffered from the cons of the traditional auction systems that are conducted offline:
  - i. It requires physical presence and that limits participation.
  - ii. Bidders rely on auctioneers or organizers to run the event fairly, but there can be issues like rigged bids or lack of transparency in bidding history.
  - iii. Auctions are bound by time, often taking longer to organize and conduct, limiting the frequency and efficiency of events.
  - iv. Operational costs for venue, auctioneers, and staff can increase fees.
- b. When it comes to online auctions it solves some of the above problems but still have some issues that people suffered from:
  - i. **Centralization and Trust:** Typically controlled by a central authority that holds control over the bidding and transaction process, leading to concerns about data manipulation and lack of transparency.
  - ii. **High Transaction Fees:** Fees can be high, particularly for large-value items, due to intermediary costs (e.g., payment processors).
  - iii. **Security and Fraud:** Cybersecurity threats and risks of fraud are prevalent, with cases of fake listings, fraudulent bids, and bidder identity spoofing.

2. **Technical Problem:** Centralized systems for auctions are prone to single points of failure and lack transparency, which can lead to vulnerabilities such as data tampering and system downtime. Moreover, these systems require extensive resources to ensure that sensitive information remains secure. Blockchain technology, particularly Ethereum's capabilities with smart contracts, offers a solution by providing a decentralized framework that can support automated, tamper-resistant auctions with lower costs and enhanced transparency. However, implementing such a decentralized auction system comes with challenges like integrating **decentralized storage** for documents, ensuring **time-based** functionality, and maintaining economic feasibility through **efficient use of gas fees** in Ethereum.

3. **Research Question:** The research question driving this work is: Can a blockchain-based framework, specifically using Ethereum smart contracts, effectively replace centralized auction systems to provide a secure, transparent, and cost-effective solution for online auctions, while addressing technical requirements like data integrity, decentralized storage, and real-time processing and addressing the new challenges that the blockchain system produces such as scalability?

**Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

- People don't have a trivial solution because in existing auction system there is a main dependence on Centralized Intermediaries to handle transactions, verify bids, and manage user information, This dependence means users must trust the platform to act fairly, maintain data integrity, and prevent tampering or unauthorized access. Unfortunately, such trust is often challenging to enforce or verify.
- Previous solutions that come along with online auction systems is Authentication Protocols, Centralized Data Encryption and Secure Storage, and Third-Party Verification Systems, all these solutions main goal is to provide trust and transparency to the bidders to trust their system.
- These solutions are inadequate because they do not fundamentally address the root problem: centralized control. As long as the platform is centralized, users must trust the platform operators and any third-party auditors or secure storage providers to act fairly and transparently. This trust requirement leads to inherent vulnerabilities, including possible data manipulation, lack of transparency, and inefficiency.

**What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

- The paper proposes a **decentralized auction framework** using blockchain technology, specifically leveraging Ethereum smart contracts, to create a secure, and transparent auction platform.
- **Hypothesis and Idea:** The core hypothesis is that a decentralized auction system, implemented through Ethereum smart contracts, can improve the security, transparency, and efficiency of online auctions by removing the need for intermediaries. Smart contracts allow for the automation of auction processes, such as

bid submissions, verification, and winner determination, which can be transparently recorded on the blockchain.

- **Design Components:**

- **Ethereum Smart Contracts:** These handle the core auction mechanics receiving bids, verifying bid conditions, and determining the auction winner.
- **Decentralized Storage System:** Off-chain data (e.g., bid-related documents) is stored in a decentralized file system like IPFS. This setup minimizes the blockchain's storage requirements while maintaining data accessibility and integrity.
- **Trusted Oracles:** Oracles act as secure data sources that connect the smart contract to external systems, providing reliable information like time constraints for auction start and end times.

- The proposed solution leverages blockchain's features where blockchain ensures that:

- **Data is immutable:** Once a transaction is recorded, it cannot be altered, ensuring bid integrity.
- **Transparency is maintained:** All transactions are publicly viewable, making it easy for participants to verify the auction's fairness.
- **Smart contracts are trustless and automated:** This removes the need for intermediaries, minimizing transaction fees and preventing manipulation.
- Hence the main goal of decentralization, trust, transparency and security are achieved by the inherent features of the blockchain.

- Improvements over existing systems:

- **Elimination of Centralized Intermediaries:** The solution eliminates the need for auction organizers or third-party validators by using smart contracts, reducing trust requirements and costs.
- **Enhanced Security and Data Integrity:** By storing bid history on the blockchain and sensitive documents in decentralized storage, the risk of data tampering or loss is minimized.
- **Cost Efficiency:** Blockchain minimizes transaction fees by removing intermediaries. The solution also uses gas-efficient smart contract functions to manage operational costs effectively.
- **Resilience Against Cyberattacks:** The decentralized nature of blockchain enhances protection against attacks like DDoS, as there is no central server to compromise.

- The solution is achieved by the gained features which are inherited from the blockchain technology and the usage of smart contracts and algorithms are designed to optimize gas costs, enhance security, and ensure process transparency. The paper added testing scenarios validated that the framework's algorithms and smart contract functions performed correctly and efficiently under various auction conditions.

**What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

- The authors evaluate the proposed blockchain-based auction solution through a series of logical arguments, cost analyses, and test scenarios. They provide evidence and artifacts to support the feasibility and effectiveness of the solution, demonstrating that it addresses the shortcomings of traditional, centralized auction systems.
  - **Logical Argument:**
    - **Transparency and Security:** The authors argue that blockchain's immutability and transparency ensure data integrity and trust. By recording each bid and transaction on the Ethereum blockchain, the auction process becomes tamper-resistant and auditable, mitigating risks of fraud and manipulation.
    - **Cost Analysis:** They provide a detailed cost analysis of gas fees associated with different smart contract functions (e.g., bid submission, auction finalization) and shows that the fees are cheap.
  - **Artifacts and Evidence:**
    - **Smart Contract Algorithms:** The authors provide a series of algorithms representing different auction functions, such as seller qualification, bid submission, and auction finalization. These algorithms ensure that the smart contract automatically enforces auction rules and verifies transactions, reducing the need for manual oversight and human error.
    - **Sequence Diagram:** A sequence diagram illustrates interactions between the auction participants and the smart contract, showing how functions are triggered at different stages of the auction. This diagram helps validate that the system's logic aligns with the expected auction process.

- **Proof-of-Concept:** The authors developed and tested a smart contract written in Solidity and deployed it using the Remix IDE, the authors share the smart contract code on GitHub, allowing others to review, test, and potentially expand on the solution.
- **Test Scenarios and Experiments:**
  - The paper explains various test scenarios obtained upon the execution of the smart contract, These tests include checking that only qualified sellers can bid, verifying that only valid bids update the leading bid, and ensuring that bids are correctly rejected after the auction period ends.

**What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

- The proposed solution aligns well with blockchain's strengths, providing an alternative to centralized platforms by removing intermediaries, automating processes through smart contracts, and potentially reducing costs.
- Potential Flaws and Limitations lay in:
  - **Scalability and Performance:** The current Ethereum network has known scalability limitations, with transaction processing speeds of around 15 transactions per second. Although Ethereum 2.0 promises improvements, its full deployment is still uncertain. High-frequency auctions with thousands of bids could face delays or high gas costs due to network congestion.
  - **Gas Cost Variability:** While the authors demonstrate that blockchain costs are lower than centralized platforms under certain conditions, gas prices are volatile. During periods of high network demand, transaction fees can spike, potentially making the solution more expensive or slower than anticipated. This could discourage users if costs vary unpredictably.
- Most Interesting and Controversial Idea is **Automated Enforcement via Smart Contracts:** Using smart contracts to automatically enforce auction rules (e.g., bid deadlines, qualification criteria) is innovative, as it enables self-governing transactions.

- Practical Implications and Feasibility where this solution could be valuable to marketplaces, e-commerce platforms, and niche auction spaces where transparency and cost-efficiency are critical.

**What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

- The paper's contributions are significant in advancing the application of blockchain technology in online auctions, the main contributions from both the authors' perspective and an analysis of their broader impact:
  - **Framework for Decentralized Auctions:** The paper introduces a decentralized auction framework, conceptualizing a system that operates without intermediaries. It outlines how Ethereum smart contracts can be used to conduct and manage auctions, highlighting the elimination of third parties and increasing trust through blockchain's transparency and security features.
  - **Smart Contract Algorithms and Design:** The paper provides algorithms detailing the auction processes, including seller qualification, bid submission, and winner determination. The smart contracts are designed to automatically enforce rules and ensure bids are transparent, verifiable, and tamper-proof.
  - **Use of Decentralized Storage and Trusted Oracles:** The authors propose integrating decentralized storage (e.g., IPFS) to handle bid documents and trusted oracles to manage time-sensitive functions, such as auction start and end times. This addresses blockchain's storage limitations and enhances the flexibility of smart contracts by connecting them with external data sources..
  - **Detailed Cost Analysis and Gas Efficiency:** A cost analysis is provided, showing gas usage and transaction costs for different auction functions on Ethereum. By comparing these costs to traditional auction fees, the authors argue that the decentralized model can be more cost-effective.
  - **Experimental Testing and Proof-of-Concept Implementation:** The authors implement a proof-of-concept smart contract in Solidity and deploy it on the Ethereum blockchain, testing it under various auction conditions. They also publicly share the code, allowing others to evaluate, reproduce, and build on their work and also provide testcase scenarios for different cases in the paper.

- **Security and Resilience Analysis:** The paper discusses the solution's resilience to security threats, including DDoS attacks and Man-in-the-Middle (MITM) attacks, which are mitigated by blockchain's distributed nature.
- **Future Directions:** The authors suggest the framework's adaptability to other types of auctions (e.g., spectrum, government asset sales) and potential enhancements, such as improved gas efficiency with Ethereum 2.0.

**What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

- The future directions for this research involve addressing existing limitations, exploring broader applications, and enhancing the system's scalability, efficiency, and usability:
  - **Scalability Improvements with Ethereum 2.0:** The authors suggest that Ethereum 2.0, with its move to a Proof-of-Stake (PoS) consensus mechanism, will address scalability issues by increasing the network's transaction processing capacity from around 15 transactions per second (TPS) to potentially 100,000 TPS.
  - **Gas Efficiency Optimization:** The authors propose refining the smart contract design to reduce gas costs, as gas fees directly impact user adoption and overall cost-efficiency.
  - **Auction Type Flexibility and Expanded Applications:** The authors mention adapting the framework to support different auction types (e.g., Dutch auctions, Vickrey auctions).
  - **Multi-Chain and Cross-Chain Compatibility:** The authors don't explicitly mention multi-chain compatibility, but this is a next step given the limitations of Ethereum's scalability and gas fees where we could explore creating cross-chain solutions to allow auctions to occur across multiple blockchain networks, enhancing scalability and enabling auctions to utilize different blockchain networks based on cost and efficiency.
  - **Enhanced User Interface and User Experience:** The authors focus primarily on the back-end framework but acknowledge the importance of a user-friendly front-end to make the system accessible to non-technical users.

**What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

- Here are several questions and discussion points I'd raise about this work:
  - **Scalability Concerns:** How do the authors envision this framework handling high-frequency, high-participation auctions? Would the current design handle scenarios with thousands of simultaneous bids, or would it be better suited for smaller auctions until Ethereum 2.0 widely adopted.
  - **Gas Fees and Cost Control:** Given the volatile nature of Ethereum gas fees, how does the model ensure consistent cost efficiency for participants?
  - **Future Auction Types and Flexibility:** How adaptable is this framework to accommodate various auction types, like sealed-bid or Dutch auctions, without requiring extensive modifications?
  - **Cross-Chain Compatibility and Multi-Chain Future:** With the rise of cross-chain compatibility, could this framework be adapted to support multi-chain auctions, allowing users to interact with auctions across different blockchain ecosystems.
  - **Privacy in a Transparent System:** How can privacy be balanced with transparency in this system? While blockchain enables public verification, would sensitive data, such as user identities or bid amounts, be adequately protected?
  - **Integration with Current Auction Ecosystems:** How might this system coexist or integrate with current online auction platforms? Could there be a hybrid model where existing platforms adopt decentralized elements, like smart contracts for transparency, while retaining a centralized interface?



## A.9 Summary of [9]

**Title:** Secure e-auction system using  
blockchain: UAE case study

*Engineering Research Paper  
Question–Answer Form*

**Authors:** Qusa, H., Tarazi, J., & Akre, V.

**Published in:** February 2020

**What is your take-away message from this paper?**

- The paper demonstrates how blockchain technology and smart contracts can be used to address trust and security issues in electronic auctions (e-auctions). By leveraging blockchain's decentralized nature, the proposed system ensures anonymity, fairness, and transparency while overcoming limitations of traditional e-auction systems.

**What is the motivation for this work (both people problem and technical problem), and its distillation into a research question?**

1. **People Problem:** Traditional e-auctions face trust issues among participants (sellers, buyers, and auctioneers), leading to transactional misbehaviors such as:
  - a. Sellers failing to deliver assets.
  - b. Buyers refusing to pay or aborting during auctions.
  - c. Collusion among bidders to manipulate prices.
  - d. When it comes to online auctions it solves some of the above problems but still have some issues that people suffered from:
    - i. **Centralization and Trust:** Typically controlled by a central authority that holds control over the bidding and transaction process, leading to concerns about data manipulation and lack of transparency.
    - ii. **High Transaction Fees:** Fees can be high, particularly for large-value items, due to intermediary costs (e.g., payment processors).
    - iii. **Security and Fraud:** Cybersecurity threats and risks of fraud are prevalent, with cases of fake listings, fraudulent bids, and bidder identity spoofing.
2. **Technical Problem:** Achieving security properties such as anonymity, unlinkability, and confidentiality in traditional designs is complex and requires intermediaries, making the system vulnerable to security breaches.

3. **Research Question:** Can blockchain and smart contracts provide a secure, trustless, and decentralized solution for e-auction systems that meet security requirements?

**Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

- The problem persists due to:
  - The inherent trust gap in traditional e-auction systems.
  - The inability of centralized intermediaries to ensure complete fairness and transparency.
  - Vulnerabilities to attacks like collusion, repudiation, and manipulation in existing systems
- Previous Solutions and Their Inadequacy:
  - **Traditional Secure Distributed Auctions:** Focused on validity and secrecy but failed to address all auction types or ensure full trust.
  - **Web-Based E-Auctions:** Suffered from critical security issues like linkability and coalition risks.
  - **Blockchain-Based Attempts:** Encountered scalability and functional vulnerabilities, such as replay attacks and improper handling of randomness.

**What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

- **Hypothesis:** Blockchain's decentralized nature combined with smart contracts can address e-auction security issues by eliminating intermediaries and enforcing trust through immutable rules.
- **Design:**
  - a. Four Stages:
    - **Initialization:** Auction details and participants' credentials are registered on the blockchain.
    - **Registration:** Bidders obtain unique identities and tickets to participate.
    - **Bidding:** Transactions are processed through smart contracts while maintaining anonymity.

- **Winner Decision:** The highest bid is validated, and the transaction is securely completed.
- **Belief in Effectiveness:** Blockchain inherently provides transparency, immutability, and decentralization, which align with the security requirements of e-auction systems.
- **Improvement:** The system eliminates the need for intermediaries, enhances fairness, and reduces risks of collusion and data leaks.
- **Implementation:** A prototype was developed using IBM Hyperledger Fabric and Composer, focusing on functions like asset registration, bidding, and ownership transfer.

**What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

- **Evidence:** A "Car Auction Network" demo validated the system's functionality.
- **Limitations:**
  - Scalability remains a challenge due to blockchain's computational complexity.
  - Randomness issues may compromise anonymity under certain conditions.
- **Future Work:** Incorporating fuzzy approximation techniques to enhance anonymity and unlinkability..

**What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

- **Strengths:**
  - Innovative application of blockchain for e-auctions.
  - Provides a practical and secure solution for a real-world problem.
- **Weaknesses:**
  - Scalability and randomness challenges need further research.
  - Usability concerns for non-technical stakeholders.
- **Potential:** The solution has broad applicability in government and private sectors, enhancing trust in online transactions.

**What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

- **Authors' Contributions:**

- Blockchain-based architecture for e-auctions.
- Practical prototype demonstrating the feasibility of the approach.

- **My Opinion:**

- The paper introduces a promising methodology for tackling security and trust issues, making it a significant contribution to blockchain and e-commerce research.

**What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

- **Authors' Suggestions:**

- Address scalability through optimization techniques.
- Improve anonymity using cryptographic methods and fuzzy logic.

- **My Suggestions:**

- Explore cross-platform integration for broader adoption.
- Conduct large-scale simulations to analyze system performance under high transaction loads.

**What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

- How can the system handle extreme scalability requirements with thousands of bidders simultaneously?
- What cryptographic methods can be integrated to enhance randomness and anonymity?
- How does the solution compare to existing blockchain-based systems in terms of performance and usability?

- Can this approach be extended to other domains like e-voting or supply chain auctions?

## A.10 Summary of [10]

**Title:** Scalable Off-Chain Auctions.

**Authors:** Minaei, M et al

**Published in:** 24 September 2024

*Engineering Research Paper  
Question–Answer Form*

### 1) What is your take-away message from this paper?

The paper presents a scalable, privacy-preserving off-chain auction protocol leveraging Programmable Payment Channels (PPC) and zkSnarks. It ensures low on-chain complexity  $O(k \log n)$  for  $k$  deviating bidders, maintaining privacy, security, and efficiency in comparison to traditional on-chain auctions.

### 2) What is the motivation for this work (both people problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?

- **People Problem:** Traditional online auctions lack trust due to potential manipulation by auctioneers and exposure of sensitive bidding information.
- **Technical Problem:** Blockchain-based auctions face scalability, cost inefficiency, and privacy concerns due to high on-chain complexity and the transparent nature of blockchain transactions.
- **Research Question:** How to design a scalable, efficient, and privacy-preserving auction mechanism that leverages blockchain's decentralization while avoiding its limitations?

**Why no trivial solution?** Combining privacy (hiding bid details), correctness (ensuring auction fairness), and efficiency (scalable operations) simultaneously is non-trivial due to conflicting requirements like bid comparison versus bid secrecy.

**3) What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

**Key Idea:** A hybrid auction protocol using PPC and zkSnarks to conduct sealed bid auctions predominantly off-chain.

**Why It Works:** Ensures bid privacy, correctness, and scalability through efficient cryptographic constructs.

**Improvements:**

- $O(k \log n)$  complexity for  $k$  deviating bidders.
- Enhanced privacy via zkSnarks, revealing only the winner and winning bid.
- Low on-chain interaction in optimistic scenarios.

**Implementation:**

- Uses Merkle Trees for bid inclusion proofs.
- Employs zkSnarks for verifying the highest bid without revealing others.

**4) What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

**Artifacts:** A proof-of-concept implemented shows a reduced gas costs and high scalability.

**Performance:** Outperforms traditional on-chain solutions in gas cost and bid inclusion.

**5) What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

**Strengths:**

- Innovative use of PPC and zkSnarks for scalability and privacy.
- Introduces scalability unlike other auctions by not even being dependent on the number of bidders, but rather depending on the number of malicious bidders
- In case of one bidder being malicious bidder, auction continues (no wasted work)

**Flaws:**

- No public verifiability

- No deposit is paid, if a bidder is malicious, he can get away by transferring his money to another account before the auctioneer challenges him.
- Centralized: Auctioneer can prevent people from joining the auction

**6) What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

The paper provides a very promising idea of putting all the bids into only one number and using inclusion proofs to say that a bidder is in the auction.

**7) What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

- Find a way of adding public verifiability
- Removing trust from of auctioneer
- Adding a way of handling deposits from bidders
- Adding a way to enforce the auctioneer to accept anyone joining the auction.

**8) What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

- There is no deposit, so what happens if a malicious bidder converts the money to another account, now the auctioneer can't punish him.
- No public verifiability. If a bidder shows invalid reveal and the auctioneer doesn't punish him, no one can make sure of that.

## A.11 Summary of [11]

**Title:** Verifiable sealed-bid auction on the Ethereum blockchain

*Engineering Research Paper  
Question–Answer Form*

**Authors:** H. S. Galal and A. M. Youssef

**Published in:** 5 October 2019

### 1) What is your take-away message from this paper?

This paper presents a cryptographic protocol and smart contract for a **verifiable sealed-bid auction** on the Ethereum blockchain, addressing the challenges of privacy, fairness, and transparency. The solution effectively utilizes **Pedersen commitments** and **zero-knowledge proofs (ZKP)** to ensure privacy and public verifiability, making it suitable for secure financial applications without requiring extensive interactions from bidders.

### 2) What is the motivation for this work (both people problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they Inadequate?

#### People Problem

Traditional sealed-bid auctions rely on trusting the auctioneer, who could collude with bidders to manipulate results. There's a need for decentralized, transparent auction mechanisms, especially in governmental or financial applications where corruption or privacy concerns are paramount.

#### Technical Problem

Ethereum smart contracts inherently lack privacy due to their public nature. This transparency conflicts with the confidentiality required in sealed-bid auctions. Previous attempts either:

1. **Lacked efficiency** due to excessive computational overhead (e.g., secure multi-party computations).
2. **Leaked sensitive data** like bid rankings or participant identities.



## Research Question

How can we design a sealed-bid auction protocol that ensures privacy, transparency, and fairness on the Ethereum blockchain?

**3) What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

## Design and Hypothesis

The paper proposes a smart contract leveraging cryptographic tools:

1. **Homomorphic Commitments** for secure bid submissions.
2. **Non-interactive Zero-Knowledge Proofs (ZKP)** to validate auction outcomes without revealing losing bids.

## Why It's Expected to Work

The design ensures:

- Privacy: Bidders can't infer others' bids.
- Verifiability: ZKPs allow public correctness checks.
- Resistance to collusion: The auctioneer is provably accountable via cryptographic guarantees.

## Achieving the Solution

1. **Deployment:** The contract is initialized with parameters (e.g., time intervals, deposit amounts).
2. **Commitment:** Bidders submit Pedersen commitments to their bids.
3. **Opening:** Bidders reveal encrypted bid data to the auctioneer.
4. **Verification:** The auctioneer uses ZKP to prove the correctness of the winner's selection.
5. **Finalization:** The winner pays, and funds are redistributed.

**4) What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

The authors provides:

- **Logical Proofs:** Correctness is ensured via cryptographic primitives (e.g., Pedersen commitments, ZKP).
- **Cost Analysis:** Gas usage for Ethereum transactions is detailed.
- **Prototype Implementation:** Tested on a private Ethereum blockchain

**5) What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

**Strengths**

- **Feasibility:** Gas costs and prototype testing validate practical deployment.
- **User Simplicity:** Bidders interact minimally, avoiding complex cryptographic operations.

**Weaknesses/Flaws**

- **Auctioneer Trust:** Despite ZKP, auctioneers retain some control, which might not be the best solution for users seeking full decentralization.
- **Cost Scalability:** Gas costs could rise significantly for more bidders or higher bid ranges.

**6) What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

1. Security of losing bids
2. Using encryption of bids by auctioneer's private key
3. Using homomorphic encryption schema

**7) What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

**Author's Suggestions**

- Extend the protocol for full bid privacy (including the winner's bid).
- Investigate mechanisms to eliminate reliance on auctioneers entirely.

**Additional Directions**

- Explore alternative blockchains (eg ZCash).
- Use zk-snark instead of bullet proofs.
- Optimize gas consumption for large-scale auctions.
- Develop hybrid systems combining off-chain computation with blockchain verification.

**8) What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

- Would a fully auctioneer-less design be feasible?
- How can the winner's bid be fully anonymized without sacrificing verifiability?

## A.12 Summary of [12]

**Title:** Predicting auction price of vehicle license plate with deep recurrent neural network

*Engineering Research Paper  
Question–Answer Form*

**Authors:** Chow, V.

**Published in:** 5 October 2019

**What is your take-away message from this paper?**

- A key takeaway from this paper is the application of deep recurrent neural networks (RNNs) to an NLP-like task, specifically predicting the prices of license plates based on the raw characters on the plates. This can be considered an NLP problem because it involves processing sequences of characters, similar to how NLP models process text data.-
- By treating the license plate prediction problem as an NLP task, the paper demonstrates the power of RNNs in extracting meaningful patterns from raw sequences without relying on handcrafted features. This is akin to how NLP models, such as those for text classification or language modeling, can learn from raw text data.
- The paper proposes an interesting solution: by extracting feature vectors from the final recurrent layer of the RNN and feeding them into a k-nearest neighbors (k-NN) model, a search engine for similar plates can be created making the model more practical and acceptable in real-world settings.

**What is the motivation for this work (both people problem and technical problem), and its distillation into a research question?**

**1. People Problem:**

- a. **Market Demand for License Plate Pricing:** There is a growing interest in understanding and predicting the prices of license plates, which can be viewed as a collectible item, investment, or unique asset. People often want to know the market value of specific plates, and this requires a robust, data-driven approach to price prediction.

**2. Technical Problem:**

- a. **Challenge of Feature Engineering:** Traditional models like regression and n-gram require extensive feature engineering and domain knowledge to capture meaningful patterns in license plate data. These handcrafted features are difficult to design for a problem as unique as license plate pricing, which requires capturing subtle relationships in the characters (both numeric and alphabetic).

**3. Limitations of Traditional Models:**

- a. Other models often fall short because they need to rely on these handcrafted features, which can be error-prone and less adaptable to new data patterns. The challenge is to develop a more automated approach that can learn directly from the raw input (the license plate strings) without the need for such hand-crafted features.

**4. Research Question:** The research question is naturally derived from these motivations:

- a. How can a deep learning model, specifically a Recurrent Neural Network (RNN), be applied to predict license plate prices from raw plate data?
- b. How can the model's predictions be made interpretable so that users can understand and trust the rationale behind the price predictions?

This leads to the development of a system where:

- The RNN can automatically learn from the raw characters on the license plates.
- The model provides a means of explaining its predictions by generating feature vectors that can be used in a k-NN search engine for similar plates, effectively offering historical context or rationale for the predictions.

**Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

- **Human trust and interpretation:** For a model to be adopted in practical scenarios (e.g., license plate price prediction), people need to understand how the model arrived at its conclusions. However, traditional deep learning models do not naturally offer explanations that are easy for non-expert users to interpret, which is a critical gap.
- **Traditional Regression and n-gram Models:** These models were among the earliest attempts at solving this problem by using handcrafted features to predict license plate prices. They rely heavily on domain expertise to create meaningful features from the raw data (the characters on license plates).
- **Feature Engineering:**
  - In earlier solutions, feature engineering was the primary method for preparing data for machine learning models. This involves selecting or creating features that represent important characteristics of the input data (e.g., the numeric or alphabetical parts of a license plate). However, feature engineering can be time-consuming, prone to errors, and dependent on domain knowledge that might not always be available or up-to-date.
  - As license plates become more diverse and personalized, it becomes increasingly difficult to design features that capture all the nuances of the data. As a result, these models often fail to generalize or struggle with unseen patterns in the data.
- **Need for a Solution that Provides Interpretability:** While deep learning models like RNNs are powerful for learning complex patterns in data, the challenge lies in explaining how they arrive at their predictions. In contexts like license plate pricing, users need to see the rationale behind the model's predictions to trust it.

**What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

The proposed solution in this study focuses on improving the interpretability and accuracy of license plate price predictions using a deep recurrent neural network (RNN) and a k-nearest neighbor (k-NN) model for explainability.

- **Deep Recurrent Neural Network (RNN):**

- The RNN is used to predict the price of license plates based on the raw characters (numbers and letters) on the plates. Unlike traditional models that rely on handcrafted features, the RNN learns to make predictions directly from the raw data without manual feature engineering.
- The RNN is trained to learn the complex patterns and relationships in the license plate data, capturing the sequential nature of the characters (e.g., how the number of digits or specific letters affect the value of the plate).
- The hypothesis here is that deep learning models like RNNs can effectively learn from large datasets with minimal feature engineering, making them more adaptable and efficient than traditional models.

- **Feature Vector Extraction:**

Once the RNN is trained, a feature vector is extracted for each license plate by summing the outputs of the last recurrent layer over time. This feature vector, which represents the "understanding" of the license plate by the model, is used as input to a k-NN model.

- The feature vector is of the same size as the number of neurons in the last layer of the RNN, and it provides a condensed representation of the key characteristics of the plate that contribute to its predicted price.

- **k-Nearest Neighbor (k-NN) Model:**

- The k-NN model is used to provide explainability for the RNN's predictions. By feeding the feature vectors for all plates (generated by the RNN) into the k-NN model, it can search for similar plates based on their feature vectors and provide historical examples as explanations for the RNN's predictions.
- When a new license plate price prediction is made, the k-NN model returns the most similar plates from the training set, offering rationale for the prediction.
- This method helps bridge the gap between the complex predictions of the RNN and the need for human-understandable explanations.

- **Why it will work:**

- RNNs are well-suited for sequence data like license plate characters, which may have inherent relationships between the letters and digits (e.g., numeric digits may correlate with lower prices, or specific letters may be more desirable for higher-value plates).
- The feature vector extraction and k-NN combination is a novel approach to make predictions interpretable. The k-NN model provides a way for users to understand the rationale behind the prediction by showing similar plates, thus

enhancing trust and usability. For instance, if a plate is predicted to have a high value, the k-NN can show historical examples of similar high-value plates, which justifies the prediction.

- **Improvements over existing systems:**

- **No Need for Handcrafted Features:** Traditional models required manual feature engineering, which is time-consuming and often ineffective for more complex or diverse datasets. The deep RNN bypasses this need by learning directly from the raw characters on the plates. This makes the solution more flexible and scalable.
- **Increased Predictive Accuracy:** By using an RNN, the model is able to learn from the raw data and capture intricate patterns that might not be apparent through traditional models. This is expected to lead to higher accuracy, as demonstrated by the significantly better performance of the RNN compared to other models like regression and n-grams.
- **Explainability and Trust:** One of the biggest issues with deep learning models is their lack of transparency. The use of the k-NN model provides a simple way to explain why a certain price is predicted. This improves user trust by giving human users something tangible to look at (similar plates) as the rationale for a prediction.

**What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

- **Improved Prediction Accuracy:**

- The deep RNN model outperforms traditional models like regression and n-grams in terms of accuracy. The model's ability to learn directly from the raw character sequences of license plates, without relying on handcrafted features, allows it to capture complex patterns in the data that other models cannot. The author suggests that this leads to significantly higher accuracy for price predictions.

- **Explainability of Predictions:**

- The k-NN model provides explanations for the predictions made by the RNN. This is crucial for gaining user trust, as the k-NN approach allows users to see



historical examples that resemble the current query, showing why the model made a specific prediction.

- **Generalizability:**

- The approach is expected to generalize well to similar types of problems, especially in cases where sequential data or categorical data (like license plates) need to be analyzed for pricing or valuation tasks.

- **Evidence:**

- Table 1 in the paper shows the accuracy of the model using RNN and how it achieves high accuracy reaching 80%.
- Table 2 in the paper illustrates how the system works with three examples: low-value, middle-value, and high-value plates. The predictions made by the RNN model are paired with historical examples retrieved by the k-NN model, showing how the rationale varies based on plate type and value. These examples serve as concrete evidence that the system can handle real-world use cases.

**What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

- The proposed approach is a solid and innovative approach to a somewhat niche but real-world problem. By combining a deep learning model (RNN) with a classical machine learning model (k-NN) for explanation, the solution addresses two key concerns:
  - **Prediction accuracy:** The RNN's ability to handle complex, sequential data is well-suited for the license plate problem.
  - **Interpretability:** Using the k-NN model to provide rational examples of why a particular prediction was made helps bridge the gap between complex model behavior and user trust.

This hybrid approach, combining the strengths of both deep learning and traditional machine learning models, is a powerful idea for applications requiring not only accuracy but also explanation, particularly in fields where human users need to understand the rationale behind automated predictions.

- **Practical Implications:** Access to a large dataset of historical license plate prices to train and improve the model.

**What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

- **Software/Implementation:** The paper doesn't explicitly mention the software used for implementation, but it likely relies on deep learning frameworks like TensorFlow or PyTorch for training the RNN, and a basic k-NN model for the search engine. Both of these are well-established tools in machine learning and are capable of handling the tasks described.
- **Experimental Results:**
  - The author presents experimental results comparing the RNN model's performance to traditional models (regression and n-grams). The results show a significant improvement in prediction accuracy, confirming the effectiveness of the RNN approach for license plate pricing.
  - Historical examples retrieval: For each predicted plate price, the system retrieves top examples from the training set using the k-NN model, providing transparency to the prediction.
  - The paper demonstrates that the system works for plates with varying characteristics (e.g., numeric or alphabetical components) and can handle complex cases where prices are harder to predict.

**What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

The model's reliance on raw character features could potentially lead to biases in pricing predictions, especially if certain characters are perceived as more valuable based on cultural or social biases. Future work could explore fairness in predictions, ensuring that the model does not inadvertently favor certain types of plates over others based on external factors. This could involve bias mitigation techniques to ensure equitable pricing for all types of license plates.

**What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

- How can we improve the transparency and trustworthiness of the RNN-based predictions?
- While the use of k-NN to generate rationale for predictions is an interesting idea, does it truly provide interpretable insights for all types of license plates, especially rare or personalized ones?
- How should the model handle personalized plates, and could the current system predict these prices effectively?
- Given the subjective nature of personalized plates, could the model be adapted to account for cultural and personal significance in the predictions?
- What methods could be used to capture the value of vanity plates that often don't follow general numeric/alphabetic rules?
- Could the system be used for exploitation or unfair pricing?
- Is there a risk that such a system could be used to artificially inflate the value of certain plates, especially for personalized or vanity plates?
- What are the ethical implications of using machine learning for pricing in domains like license plates, where the value might be tied to personal or social significance?
- How feasible is it to deploy this system at scale in real-world environments?
- What would be the computational and storage requirements for running such a model on a large-scale platform? Could it be deployed efficiently in practice, or would it require significant resources?
- How might the system be integrated with existing auction platforms, or could it operate independently in a government or commercial setting?
- Who owns the data used to train these models, and how does privacy factor into this?
- Given that license plates are tied to specific individuals or vehicles, how does the system address privacy concerns related to using such data for predictions?
- Should users have the option to control how their data is used in the prediction system, especially if personalized plates are involved?

## A.18 Summary of [18]

**Title:** Ethereum-based implementation of English, Dutch, and First-price sealed-bid auctions

*Engineering Research Paper  
Question–Answer Form*

**Authors:** Pop, C., Prata, M., Antal, M., Cioara

**Published in:** 05 Sep 2020

### **What is your take-away message from this paper?**

The paper demonstrates that blockchain-based implementations, specifically using Ethereum smart contracts, can enhance the security, transparency, and efficiency of online auctions (English, Dutch, and First-price sealed-bid auctions) by addressing the limitations of traditional centralized systems.

**What is the motivation for this work (both people problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

#### **People Problem:**

- Centralized online auction systems lack transparency, are susceptible to fraud (e.g., false bids, item misrepresentation), and require trust in third-party payment systems.

#### **Technical Problem:**

- Centralized platforms suffer from vulnerabilities like bid repudiation and reliance on intermediaries for fair execution.
- Blockchain technology promises a decentralized solution but faces challenges like ensuring bid privacy and transaction costs.

#### **Research Question:**

How can Ethereum-based smart contracts and blockchain technology be leveraged to improve the transparency, security, and efficiency of online auctions while minimizing transaction costs?

#### **Why a Trivial Solution Is Inadequate:**

- Centralized solutions require trust in platforms, which may manipulate auctions.
- Privacy concerns make traditional blockchain implementations unsuitable, especially for sealed-bid auctions.

#### **Previous Solutions and Their Inadequacies:**

- Previous blockchain auction systems struggled with confidentiality (sealed-bid privacy) and computational inefficiency.

**What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

#### **Key Ideas:**

- Use Ethereum smart contracts for auction management, eliminating third-party intermediaries.
- Implement a Merkle proof algorithm to verify the integrity of sealed bids.
- Integrate IPFS for decentralized storage to reduce on-chain costs.

#### **Why It's Believed to Work:**

- Ethereum's immutability ensures transaction validity and prevents repudiation.
- Smart contracts enforce rules, ensuring fairness and transparency.

#### **Improvements:**

- Incorporates cryptographic methods for sealed-bid verification.
- Optimizes gas usage through a reusable auction engine.

#### **Achieving the Solution:**

- Developed distinct workflows for each auction type.
- Used Merkle Proof for sealed-bid verification and IPFS for off-chain data storage.

**What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

#### **Evidence and Experiments:**

- Simulated auctions on Ethereum to evaluate gas costs and performance.
- Comparative analysis of transaction costs for each auction type:
  - Dutch auctions are cost-effective and fast.
  - English auctions favor bidders by enabling incremental bids.

- First-price sealed-bid auctions ensure bid privacy but incur higher costs.
- Validation using metrics like gas usage and execution times.

**What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

**Strengths:**

- Addresses key weaknesses of traditional auctions.
- Demonstrates practical applicability through real-world examples.
- Highlights the trade-off between security, transparency, and cost.

**Flaws:**

- High transaction fees and response times remain challenges.
- Sealed-bid auctions require further work to achieve complete privacy.

**Interesting Ideas:**

- Combining blockchain with IPFS for efficient data storage.
- Using Ethereum smart contracts to enforce strict auction rules.

**Practical Implications:**

- Suitable for high-value or fraud-sensitive auctions.
- Adoption depends on reducing transaction costs and improving user accessibility.

**What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

**Authors' Contributions:**

- Defined workflows for three popular auction types.
- Integrated Merkle Proof for sealed-bid auctions.
- Demonstrated cost-effectiveness through simulations.

**Additional Insights:**

- IPFS integration offers a scalable off-chain storage solution.
- Comparative cost analysis provides guidance for auction choice.

**What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

**Authors' Suggestions:**

- Explore advanced protocols for enhanced bid privacy.
- Optimize computations to reduce transaction fees and response times.

**Additional Ideas:**

- Investigate integrating layer-2 solutions for lower transaction costs.
- Extend the system to support multi-item and combinatorial auctions.

**What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

- How can transaction fees be reduced without compromising security?
- Can this system be generalized for non-auction applications like supply chain contracts?
- What are the trade-offs between gas optimization and scalability for large-scale adoption?

## A.19 Summary of [19]

**Title:** FastFabric: Scaling Hyperledger Fabric to 20,000 transactions per second

*Engineering Research Paper  
Question–Answer Form*

**Authors:** Christian Gorenflo, Stephen Lee, Lukasz Golab, Srinivasan Keshav

**Published in:** 21 Jan 2020

### What is your take-away message from this paper?

- The key take-away message from the paper is that Hyperledger Fabric, a prominent permissioned blockchain platform, faces significant performance bottlenecks that limit its transaction throughput.
- The authors propose "FastFabric," a re-architected version of Hyperledger Fabric, which achieves a substantial improvement in throughput (up to 20,000 transactions per second) by addressing non-consensus-related inefficiencies, such as transaction ordering and validation. This work demonstrates the potential for architectural optimizations to enhance blockchain scalability and make it more suitable for high-demand applications, without requiring changes to existing interfaces or user workflows.

### What is the motivation for this work (both people problem and technical problem), and its distillation into a research question?

- **People Problem:** Blockchain technologies like Hyperledger Fabric are increasingly being adopted across industries for applications requiring secure, transparent, and decentralized systems. However, their limited transaction throughput creates a barrier to adoption in high-demand use cases, such as financial services, supply chain management, and large-scale enterprise applications.
- **Technical Problem:** Hyperledger Fabric suffers from architectural bottlenecks that restrict its scalability, particularly in transaction ordering and validation processes. These inefficiencies prevent it from achieving the high throughput required for large-scale operations, making it less competitive with traditional distributed database systems.



- **Research Question:** How can the architecture of Hyperledger Fabric be modified to overcome its performance limitations and achieve a significantly higher transaction throughput (e.g., 20,000 transactions per second) without altering its core APIs or compromising its security and functionality?

## **Why doesn't the people's problem have a trivial solution? What are the previous solutions and why are they inadequate?**

**The people's problem cannot be solved trivially because:**

- **Complexity of the Architecture:** Hyperledger Fabric's modular design involves multiple interdependent components, such as ordering, endorsement, and validation processes. Optimizing one component without disrupting others is a challenging task.
- **Competing Requirements:** Enhancing throughput must not compromise other critical properties, such as security, decentralization, and fault tolerance. Balancing these requirements requires sophisticated solutions.
- **Performance Trade-offs:** Increasing throughput often involves trade-offs with other metrics, such as latency or resource utilization, which complicates straightforward fixes.

## **Previous Solutions and Their Inadequacies:**

- **Sharding and Partitioning:** Techniques like sharding divide the blockchain into smaller partitions to process transactions in parallel.
  - Limitation: While effective for public blockchains, these techniques are complex to implement in permissioned systems like Hyperledger Fabric due to dependencies between partitions.
- **Caching and Resource Scaling:** Increasing hardware resources or implementing caching mechanisms to handle higher loads.
  - Limitation: This approach does not address the architectural inefficiencies and quickly becomes cost-prohibitive for large-scale systems.
- **Simplified Blockchain Designs:** Some researchers propose simplifying blockchain designs by removing components, such as smart contract layers or endorsement policies.
  - Limitation: These solutions reduce the functionality and flexibility of the blockchain, making it less suitable for enterprise use cases.

The previous solutions fail to address the broader architectural limitations of Hyperledger Fabric, particularly the bottlenecks in transaction ordering and validation. These areas remain the primary hurdles to achieving high throughput, and addressing them requires novel architectural changes, as proposed by "FastFabric."

**What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

- **Hypothesis:** Hyperledger Fabric's transaction throughput can be significantly improved by rearchitecting the ordering and validation processes, which are the primary bottlenecks, without changing the system's core APIs or its modular design.
- **Idea:** The authors introduce FastFabric, a modified version of Hyperledger Fabric. The solution focuses on optimizing the system's transaction processing pipeline, particularly by:
  - Reducing computational and I/O overhead during transaction ordering.
  - Enhancing the efficiency of validation processes to reduce delays in approving transactions.
- **Design:**
  - **Optimized Transaction Ordering:**
    - i. Streamlining the ordering process to minimize redundant computations and disk I/O.
    - ii. Introducing more efficient batching techniques for handling transaction requests.
  - **Improved Validation Mechanisms:**
    - i. Designing a leaner validation protocol that reduces the computational load without compromising the integrity of the transactions.
    - ii. Employing parallel processing to validate multiple transactions concurrently.
  - **Backward Compatibility:**
    - i. The solution is designed to be plug-and-play, meaning it requires no changes to existing APIs or external interfaces, ensuring easy integration into current Hyperledger Fabric deployments.
- **Why Is It Believed to Work?**
  - Because it focuses on Critical Bottlenecks By targeting the ordering and validation processes, the proposed solution directly addresses the core limitations of the system.
- **How It Represents an Improvement**

- Throughput: FastFabric increases throughput from the typical 3,000 transactions per second in Hyperledger Fabric to 20,000 transactions per second.

**What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

- **Author's Evaluation of the Solution:** The authors rigorously evaluate the effectiveness of FastFabric through a series of experiments and comparisons with the original Hyperledger Fabric. They conclude that their proposed solution achieves a significant improvement in throughput and scalability, confirming its feasibility for enterprise-scale applications.
- **Evidence and Artifacts**
  - **Proof-of-Concept System:** The authors implemented a prototype of FastFabric to demonstrate the feasibility of their architectural changes.
  - **Experimental Setup:** They conducted experiments using realistic workloads and configurations to evaluate the performance of FastFabric against the baseline Hyperledger Fabric.

**What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

- **Analysis of the Identified Problem:** The authors have effectively identified critical bottlenecks in Hyperledger Fabric, specifically in the ordering and validation processes. These components are indeed major barriers to achieving high throughput in permissioned blockchains, making the problem well-defined and relevant. Their focus on these bottlenecks aligns with the need for scalable blockchain solutions in enterprise applications.
- **Perceived Flaws:**

- Limited Scope of Optimization:  
The authors focus on ordering and validation but may not address other potential bottlenecks, such as smart contract execution or network communication overhead.
- Resource Trade-offs:  
The improvements might come at the cost of increased resource utilization in some scenarios (e.g., higher memory or CPU usage for parallel validation).  
This trade-off is not deeply explored in the paper.
- **Will This Work:** Yes, the results strongly suggest that FastFabric is a viable solution for improving Hyperledger Fabric's throughput. However, further validation in diverse real-world environments would solidify its effectiveness.
- **Who Would Want It?:** Enterprises and organizations using Hyperledger Fabric for applications requiring high transaction throughput, such as financial services, supply chain management, and IoT networks, would greatly benefit from this solution.
- **What Will It Take to Give It to Them?:**
  - Adoption would require robust documentation, open-source availability, and active support for integrating FastFabric into existing systems.
  - Training for developers and administrators to understand and leverage the new optimizations.
- **When Might It Become a Reality?:** If the prototype is made open source and actively supported, it could be adopted within 1–2 years, assuming further testing and validation are successful.

**What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

- Authors' Contributions:
  - Optimized Hyperledger Fabric Architecture (FastFabric):
    - i. The authors introduce a rearchitected version of Hyperledger Fabric that enhances throughput and scalability.
    - ii. Focus is placed on optimizing transaction ordering and validation, key bottlenecks in the original system.
  - Experimental Results:

- i. The paper provides empirical evidence demonstrating a sixfold increase in throughput (up to 20,000 TPS) and reduced latency compared to the original system.
  - ii. Experiments also highlight improved scalability under high transaction loads.
- Parallel Validation Technique:
  - i. Proposes and implements a parallel validation mechanism to process transactions efficiently, reducing bottlenecks in transaction validation.
- Proof-of-Concept Implementation:
  - i. A prototype of FastFabric is developed and evaluated, serving as a practical demonstration of the proposed improvements.
- Opinion on contributions: FastFabric represents a substantial advancement in permissioned blockchain systems, with contributions that blend theoretical innovation with practical implementation and evaluation.

**What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

- **Authors' Suggestions (Implied):**
  - Enhancing Scalability:
    - i. The paper emphasizes scalability improvements but leaves room for further optimization, particularly for networks with hundreds or thousands of nodes.
  - Real-World Deployment:
    - i. While FastFabric demonstrates impressive results in experimental settings, the authors hint at the need for broader testing in diverse real-world applications.

**What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you find difficult to understand? List as many as you can.**

- **Real-World Deployment:**
  - What industries or applications are best suited for FastFabric?

- Have there been any pilot deployments or collaborations with enterprises to test FastFabric?
- **Compatibility:**
  - Are there limitations or challenges when integrating FastFabric with existing Hyperledger Fabric-based applications.
  - Does maintaining backward compatibility introduce inefficiencies?
- **Resource Usage:**
  - How much additional computational or memory overhead is required for FastFabric compared to standard Hyperledger Fabric?
- **Questions for Open Discussion**
  - How does FastFabric's improvement in throughput and latency affect the perception and adoption of permissioned blockchains compared to public blockchains?
  - Could FastFabric's design influence other blockchain platforms, including Ethereum or Corda?
  - Can FastFabric's techniques be generalized for blockchains handling smart contracts with significantly higher computational requirements?
- **Points of Difficulty**
  - Validation Process Details: The exact mechanisms used to maintain consistency and detect conflicts during parallel validation are complex and warrant deeper analysis.
  - Experimental Limitations: The paper does not fully explore the limits of scalability, leaving some ambiguity about performance in extreme or edge-case scenarios.

## **Appendix B**

### **Application of SPI Model**

## B.1 Implementation Plan

	ACTIONS	ACTIVITIES	
1	Proposal Preparation	<p>1. Prepare a presentation outlining <b>project ideas</b> with the expected scope of work for each, and emphasizing feasibility and relevance to <b>project goals</b>.</p>	
2	Researching Related Works and Applications	<p>2. Build a strong scientific background by <b>reading</b> papers, <b>summarizing</b> them, and <b>explaining</b> them to each other.</p> <p>3. Search about reliable and credible <b>open-sources</b> that we can <b>use</b> in our project.</p> <p>4. Re-iterate the <b>comparisons</b> between related work while trying to add <b>more extensions</b> to the current work and <b>finalizing the project scope</b>.</p> <p>5. Research the possibility of using <b>ML</b> in BitAuction.</p> <p>6. Identify and document functional and non-functional <b>requirements</b> for the BitAuction system.</p>	
3	Learning	<p>7. Learn about <b>Hyperledger frameworks</b> (Fabric, Composer, Raft), <b>Smart contracts</b> (Chaincode), and <b>Off-chain</b> storage (Database, IPFS).</p> <p>8. Learn about <b>Docker</b>, <b>Kubernetes</b>, cache tools like <b>Redis</b>, and <b>Apache Flink/ Spark</b> for data streams processing</p>	
4	Implementation	<p>9. Design the <b>architecture</b> of the system:</p> <p>9.1. Plan <b>integration points</b> for blockchain and off-chain components.</p> <p>9.2. Define the <b>technology stack</b> for frontend, backend, and blockchain.</p> <p>10. Develop the <b>Blockchain Smart Contracts</b> (chaincode on Hyperledger):</p> <p>10.1. Implement <b>core auction logic</b> for <b>Open-Cry</b> as smart contract.</p> <p>10.2. Define <b>auction-specific rules</b>, including bidding timeframes, reserve prices, and dispute resolution.</p> <p>10.3. Design an efficient <b>transaction ordering</b></p>	



	<p>system.</p> <p><b>10.4. Test</b> smart contracts for <b>security</b> and <b>correctness</b>.</p> <p><b>11. Build the Backend services:</b></p> <p><b>11.1.</b> Develop <b>APIs</b> for auction operations.</p> <p><b>11.2.</b> Implement user <b>authentication, authorization, and role management</b>.</p> <p><b>11.3.</b> Integrate blockchain with <b>off-chain</b> components.</p> <p><b>11.4.</b> Use <b>off-chain</b> data storage like <b>IPFS</b>.</p> <p><b>11.5.</b> Improve <b>scalability</b> by using <b>multichain</b> solutions.</p> <p><b>12. Develop User friendly interface</b> (Web or Mobile Application)</p> <p><b>12.1.</b> Support dynamic <b>live</b> auction interface ensuring <b>low latency</b> and <b>real-time updates</b>.</p> <p><b>12.2.</b> Design a <b>clean, intuitive, and easy-to-navigate</b> user interface.</p> <p><b>13. Conduct testing and evaluation</b> of the system</p> <p><b>13.1.</b> Conduct an evaluation for the <b>blockchain</b> and the <b>application</b>.</p>
--	---

*Table B.1 Implementation Plan*

## **Appendix C**

### **Sealed-bid optimization**

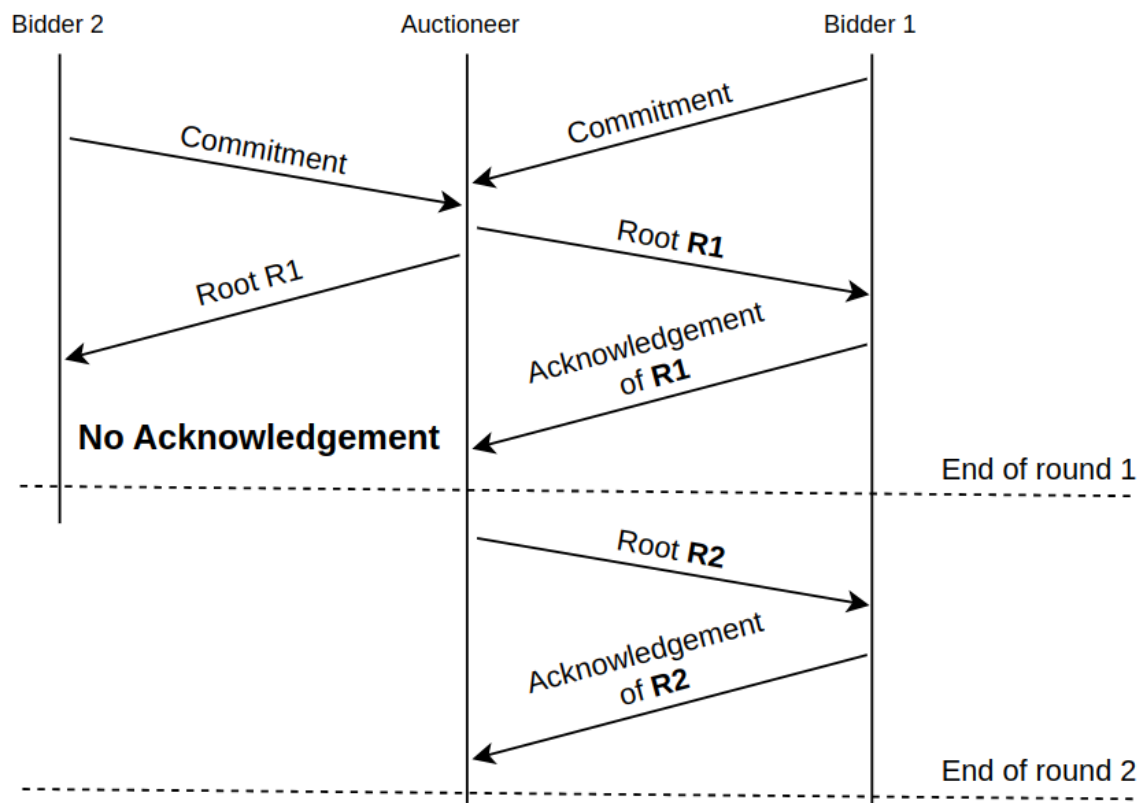
# Explanation of Sealed-bid Auction Optimization

Implement a sealed bid auction with optimizing the bidding stage by allowing bidders to submit their bids to the auctioneer signed by the bidders.

## C.1 Offchain Protocol

The offchain protocol between the auctioneer and the bidder goes as follows:

1. Each bidder that want to participate in the communication sends his bid signed by his private key to the auctioneer
2. Auctioneer announces the start of a new round of communication
3. Auctioneer collects the bids of all the bidders that participated in this round and creates a hash of an array of all these bids. Lets call it Root (R)
4. Auctioneer sends an certification to each bidder that participated in this round
  - a. contains the bidder's address
  - b. contains the Root (R)
  - c. signs it by the auctioneer's private key
5. Each bidder that participates in the round sends the auctioneer back an acknowledgement of the auctioneer's certification.
  - a. contains the bidder's address
  - b. contains the Root (R)
  - c. signs it by the bidder's private key
6. The auctioneer checks to see if all the bidders in this round sent their acknowledgement
  - a. All participating bidders sent their acknowledgement: round finished and auctioneer can put his root onchain
  - b. Not all participating bidders sent their acknowledgement: Go to step 2



## C.2 Auction onchain protocol

### 1. Initialization

Auctioneer sets up the smart contract

### 2. Bidders Register

all bidders that want to join the auction pays a deposit first before participation

### 3. Auctioneer writes the root

communicates with bidders (offchain) and write the root they agreed on (onchain)

### 4. Bidders not included in root submit bids

If bidders communicated with the auctioneer, they can check if he included them by checking the root in the certificate he sent them  $R' =$  the root he wrote onchain  $R$

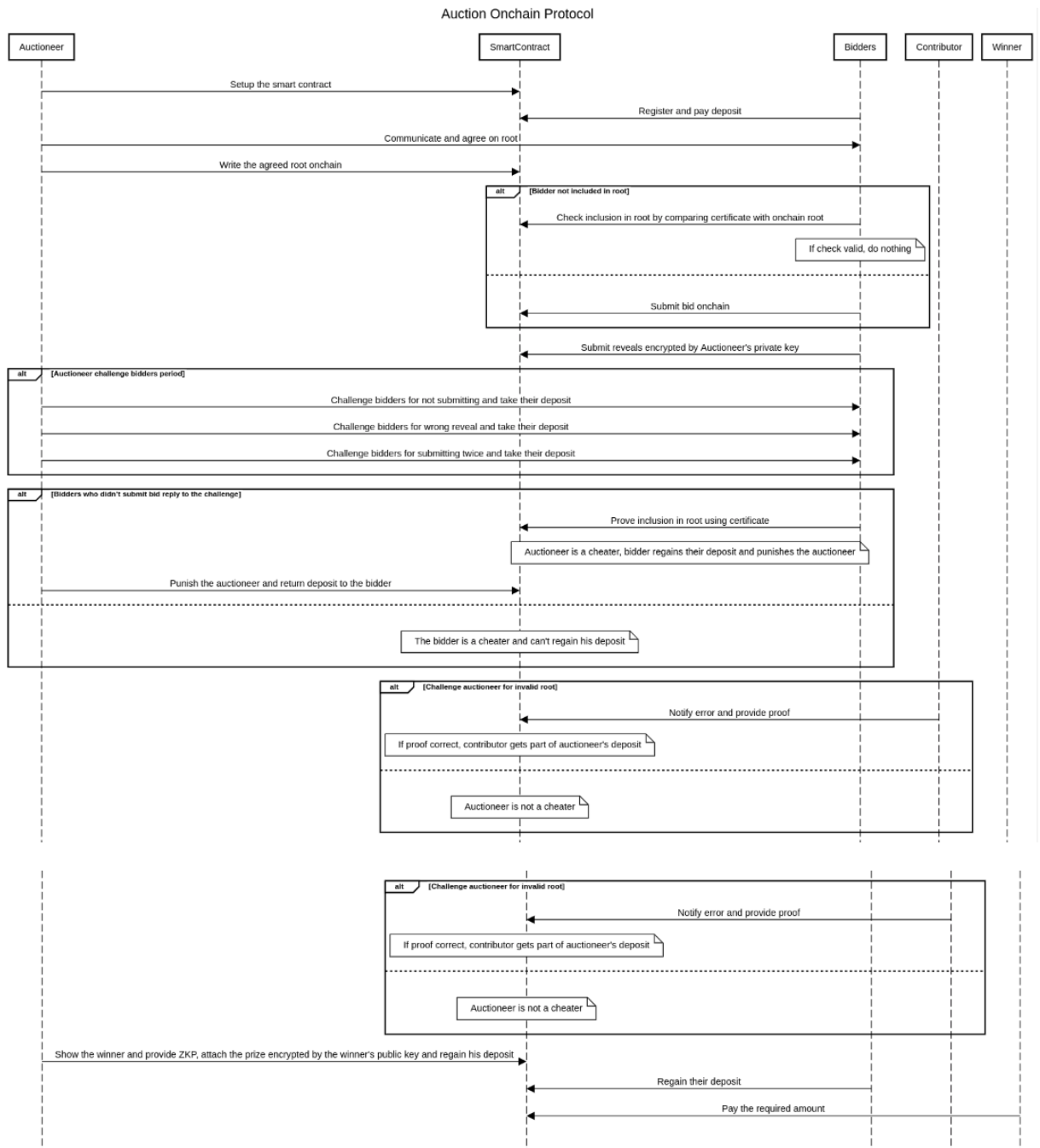
### 5. Bidders submit their reveals encrypted by the private key of the auctioneer

### 6. Auctioneer begins challenging people that he claim they are malicious

#### a. Bidders didn't submit their bid

wait for their response


- b. Bidders that revealed a value that doesn't map to their commitment (only bidders that exist in the root the auctioneer uploaded)
      - i. Auctioneer proves bidders exist in his root (using their acknowledgement)
      - ii. Auctioneer shows their commitments that are encrypted by their private keys
      - iii. Auctioneer shows their reveals don't open their commitments
      - iv. Auctioneer take their deposit
    - c. Bidders that included themselves in his root and outside his root
      - i. Auctioneer proves bidders exist in his root (using their acknowledgement)
      - ii. Auctioneer shows they submitted their bids outside
      - iii. Auctioneer take their deposit
- 7. Bidders that the auctioneer challenged them for not submitting
  - a. either proves onchain that the auctioneer included them in his root (using his certificate) -> that way they prove the auctioneer is a cheater and gets part of his deposit
  - b. or do nothing -> Auctioneer proved they cheated and take all their deposit
- 8. People checks the Root R correctness offchain, but if somebody sees error, notifies the contract onchain -> if proof is correct, this contributor gets part of the auctioneer's deposit and auctioneer is considered cheater
  - a. Proof goes as follows:
    - i. collect all the reveals  $x_i$  of bidders who submitted their bids outside the root
    - ii. For each reveal  $x_i$ , get its corresponding commitment  $X_i'$  by hashing it
    - iii. Make a new root  $R'$  by hashing all the commitments  $X_i'$
    - iv. Compare  $R'$  to the root  $R$  the auctioneer wrote onchain before
- 9. Auctioneer computes the winner and provides a Zero knowledge proof onchain
- 10. Verification of the auctioneer's zero knowledge proof
- 11. Pay the winner and Finish the auction and the auctioneer gets a small portion of the bidders' deposit that he included in his root.



## **Appendix D**

### **UI samples of BitAuction**

Live Auction Platform



### Auction Login


Organization  
Org1

User ID \*  
user1

Role  
Seller

LOGIN

Figure D.1: Login

 Seller Dashboard [CREATE AUCTION](#) [MY AUCTIONS](#) [OPEN AUCTIONS](#)

### Create Auction

Auction ID \*  
starry\_night\_auction

Item \*  
The Starry Night

Time Limit (ISO, optional)

Description  
"The Starry Night" is a renowned oil-on-canvas painting I

Picture URL  
<https://sanctuarymentalhealth.org/wp-content/uploads/20>

CREATE AUCTION

Figure D.2: Create an account



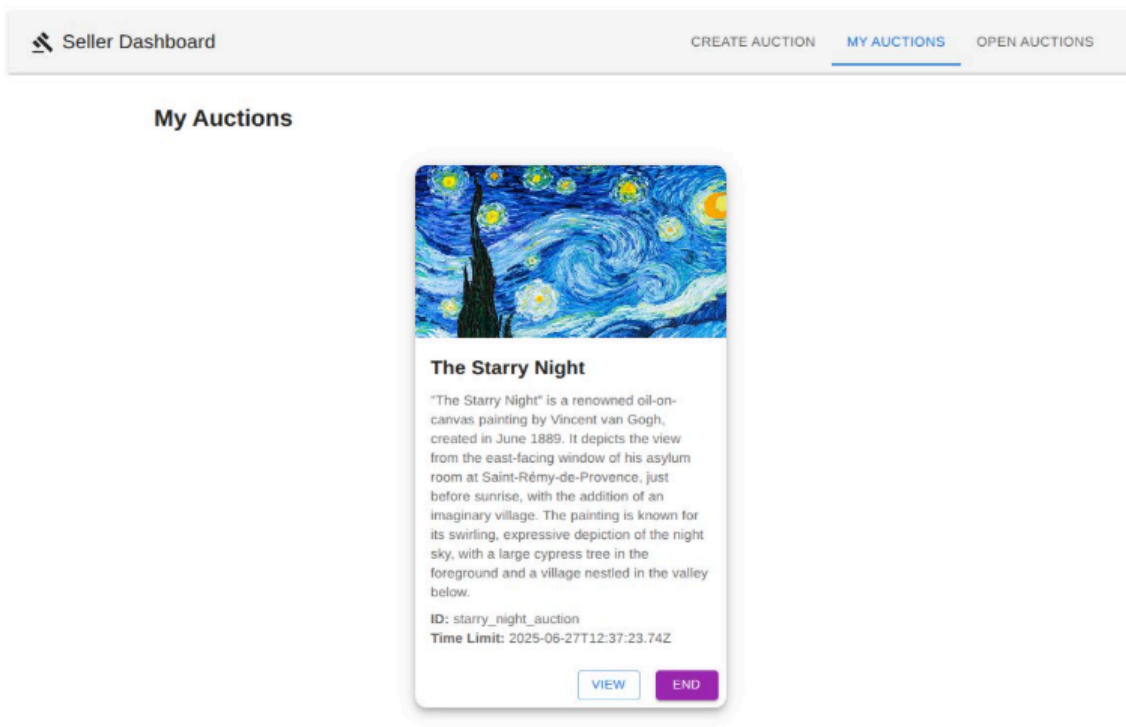


Figure D.3: Show seller auctions

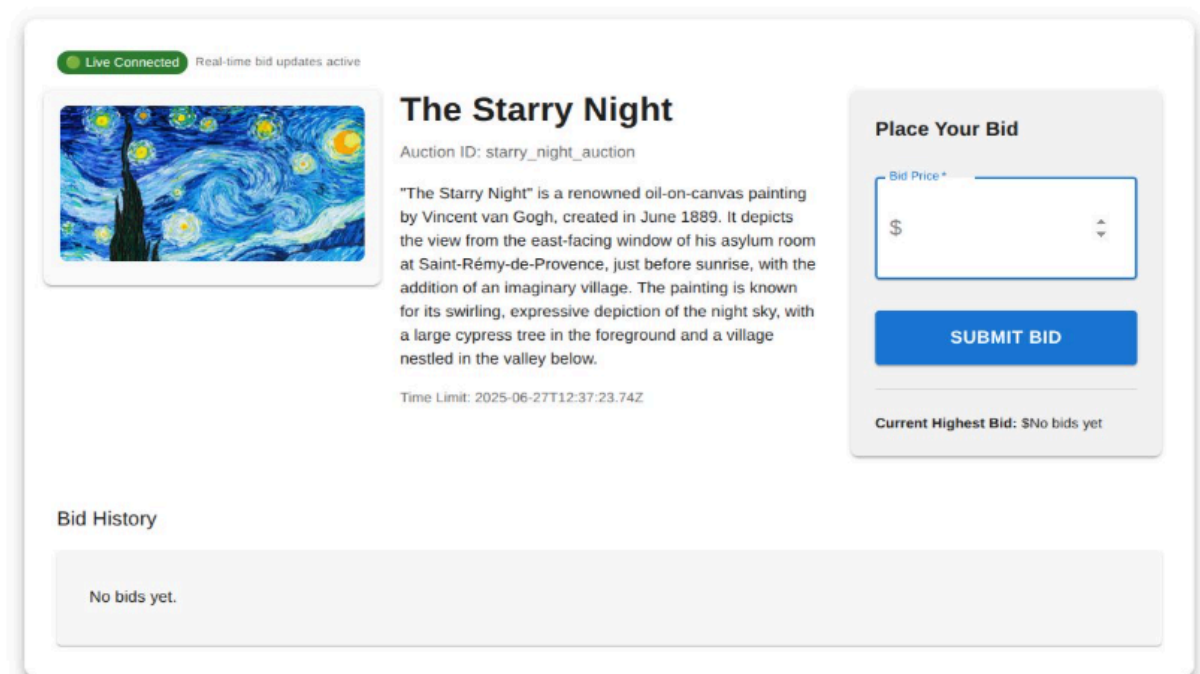


Figure D.4: Show auction details



