IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 8 April 2024, accepted 19 April 2024, date of publication 1 May 2024, date of current version 13 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3395791

RESEARCH ARTICLE

Digitalized and Decentralized Open-Cry Auctioning: Key Properties, Solution Design, and Implementation

ERIC CHIQUITO, ULF BODIN, OLOV SCHELÉN, (Member, IEEE), AND AHMED AFIF MONRAT, (Member, IEEE)
Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 97187 Luleå, Sweden

Corresponding author: Eric Chiquito (eric.chiquito@ltu.se)

This work was supported in part by the DigiPrime Project under Grant 873111; in part by the European Commission in the Scope of the H2020 Program; and in part by the RemaNet Project, as part of the HORIZON Program under Grant 101138627.

ABSTRACT Open-cry electronic auctions have revolutionized the landscape of high-value transactions for buying and selling goods. Online platforms such as eBay and Tradera have popularized these auctions due to their global accessibility and convenience. However, these centralized auctioning platforms rely on trust in a central entity to manage and control the processing of bids, e.g., the submission time and validity. The use of blockchain technologies for constructing decentralized systems has gained popularity for their versatility and useful properties toward decentralization. However, blockchain-based open-cry auctions, are sensitive to the order of transactions and deadlines which, in the absence of a governing party, need to be provided in the system design. In this paper, we identify the key properties for the development of decentralized open-cry auctioning systems, including verifiability, transaction immutability, ordering, and time synchronization. Three prominent blockchain platforms, namely, Ethereum, Hyperledger Fabric, and R3 Corda were analyzed in terms of their capabilities to ensure these properties for gap identification. We propose a solution design that addresses these key properties and presents a proof-of-concept (PoC) implementation of such design. Our PoC uses Hyperledger Fabric and mitigates the identified gaps related to the time synchronization of this system by utilizing an external component. During the chaincode execution, the creation and submission of bids initiate requests to the time service API. This API service retrieves trusted timestamps from NTP services to obtain accurate bid times. We then analyzed the system design and implementation in the context of the identified key properties. Lastly, we conducted a performance evaluation of the time service and the PoC system implementation in time-sensitive scenarios and assessed its overall performance.

INDEX TERMS Auctions, blockchain, decentralized systems, time synchronization.

I. INTRODUCTION

Electronic auctions are a popular way of buying and selling goods or services over the internet [1]. They provide users with a platform on which to compete on iterative or sealed bidding processes to secure the best offer, usually over a specified time frame.

The transition from traditional auctions, which take place in physical locations or using physical communication mechanisms is motivated by the growing adoption of digital technologies and Internet-based trading. Electronic auctions offer users a convenient way to engage in competitive bidding processes from anywhere, at any time, eliminating the constraints of physical presence, and enhancing their accessibility, scalability and transparency. This approach has made electronic auctions the preferred negotiation mechanism in industries such as consumer goods, automotive, real estate, and even government procurement [2].

The transition from traditional auctioning to electronic systems comes however, with its own set of considerations,

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Sookhak.

such as online security, trust and the need for a reliable auctioning platform to conduct high-value transactions [3]. The auctioning platforms used to conduct electronic auctions and their associated transactions (or the parties that control them) may not always be trustworthy and can maliciously influence transactions to benefit or harm a particular party [4]. Additionally, if not properly managed these platforms are subject to system malfunction which may translate into important data such as winning bids and personal information, being lost [5].

The integration of electronic auctions with decentralized systems has been proposed as an efficient way of addressing the aforementioned considerations [6]. Blockchain technologies are often preferred over other distributed implementations for their properties toward decentralization, despite their trade-offs. Its decentralization ensures transaction transparency, tamper-proof records, and resistance to manipulation in a globally accessible system [7]. Smart contracts, present in blockchain systems, facilitate the automatic execution of auction rules and transactions such as bid creation, validation and winner selection.

Auctions can be divided into two main types [8]: Noniterative or sealed and iterative or open-cry. In noniterative auctions, participants submit their bids privately, often in sealed envelopes, and the highest bidder is awarded the item or service without knowledge of other bids. On the other hand, iterative auctions focus on open competition. Here, participants make successive bids, with each bid being visible to all participants, and the auction continues until a condition is met.

In the current state of the art (SotA) of blockchain-based auction systems, there is a predominant focus on noniterative bidding systems [9], i.e., double auctions [10], [11] and single-sided sealed bid auctions [12], [13], [14]. However, in regard to open-cry (iterative) auction systems, the SotA is relatively limited in terms of research papers [15], [16], [17]. In a previous survey work, we analyzed decentralized auctions as a whole and concluded blockchain systems to be a promising implementation choice [18]. For this reason, we selected blockchain as the core of this paper.

Some of the main issues in open-cry decentralized auctions relate to the validity of the entries and transactions and determine the correct order of the transactions and the time these were created [9]. The issue of accurate time synchronization is a common issue present in several decentralized systems [19]. The most common way to allow nodes to synchronize on time is to give up some degree of decentralization by relying on a time reference provided by a centralized system. This time reference can be either an external source or a singular node which has the authority to provide its time as a reference to other nodes.

Blockchain systems frequently encounter limitations in concurrent transaction processing [20]. As a result, they introduce various mechanisms for serializing transactions instead of processing them in parallel [21]. However, in the context of an open-cry auction implementation, this serialization process can potentially compromise the overall perception of correctness. Therefore, it is essential to identify blockchain approaches that effectively address the key properties of decentralized auction systems. In cases where existing solutions present limitations, especially in the context of transaction ordering and time synchronization, it is crucial to support the implementation of the needed features to fulfill these properties.

The scope of this paper consists of the identification of key properties of open-cry auctioning systems and the analysis of SotA blockchain solutions in regard to these properties for gap identification. It also includes the practical implementation of a proof-of-concept (PoC) based on the proposed system design using Hyperledger Fabric with a focus on transaction ordering and time synchronization. The validation of this PoC entails a qualitative analysis of its alignment with the identified properties and a quantitative analysis of performance aspects related to transaction ordering and time synchronization.

The research contributions of this paper are as follows:

- The identification of fundamental quality attributes and key properties for decentralized open-cry auctioning systems (Sections II and III).
- The analysis of various blockchain frameworks based on the identified key properties. With this analysis, we identified gaps in present blockchain platforms on the context of open-cry auction systems (Section IV).
- The proposal of a design concept for developing open-cry decentralized auction systems aimed to assess the identified gaps (Section V-A). A proof-of-concept implementation for an open-cry auctioning system using the Hyperledger Fabric blockchain. We also propose necessary enhancements to address the identified properties (Section V-B).
- The analysis of the system design and implementation complemented by a performance evaluation. Furthermore, we provide reflections on blockchain systems regarding the implementation of open-cry auction systems (Section VI).

The PoC implementation using Hyperledger Fabric demonstrates a practical application of our proposed design. The open-access implementation presented in this paper addresses the key properties of decentralized open-cry auctions. This is achieved through the integration of functionalities within Hyperledger Fabric, complemented by external components for time synchronization. We also identify limitations in the SotA of blockchain technologies related to concurrency. The results of the implementation analysis, as discussed in Section VI, affirm that the system design presented herein serves as a robust foundational blueprint for open-cry auctioning systems adaptable to diverse blockchain technologies.

It is important to emphasize that our solution design is versatile and not confined to specific use cases. The choice of blockchain implementation should align with the specific requirements and objectives of the intended use case.

The paper is structured as follows: Sections II to VI are structured as presented in the contributions. Section VII presents references relevant to the topic that are not in the scope of this paper. Later, in Section VIII, we present the discussion related to the identified gaps, the proof-of-concept implementation and the additional references. Finally, in Section IX, we present the conclusions of the present paper.

II. PRELIMINARIES ON ELECTRONIC AUCTIONING

As mentioned in Section I, traditional auctions are commonly conducted in offline settings controlled by auctioneers that conduct the auctioning procedure [2]. With the digitalization of these procedures, auctions transcend space and time, improving transaction availability.

Centralized auctioning platforms such as eBay [22] and Tradera [23] rely on trusting a central entity to manage and control not only the validity of transactions but also the order in which they arrive at the negotiation system and when they do so. However, in the context of decentralized settings, it becomes necessary to explore alternative approaches that can offer these essential features while still harnessing the advantages of decentralized systems for trading.

A. QUALITY ATTRIBUTES FOR ELECTRONIC AUCTIONS

As mentioned in the introduction section, the truthful execution of an electronic auction has several quality attributes. These attributes serve as design properties that should be considered in the design and development of decentralized auctioning systems to ensure trust. Further information on several of these attributes can be found in previous work [18]. The main quality attributes we focus on are as follows:

- Correctness: This attribute refers to the correct execution of the auction rules; i.e., the allocation of resources is given to the user who values such resources the most.
- Nonrepudiation: Once a bid is submitted to a given auction, it cannot be repudiated.
- Transparency: In the absence of a centralized entity, every user shall be able to verify the outcome of a transaction and receive the same results.
- Security: Decentralized auctioning systems should have safety measures against malicious entities from altering the auction process.
- Scalability: The system shall be able to support the creation and processing of multiple concurrent auctions in the same system.

B. ELECTRONIC OPEN-CRY AUCTIONING

Open-cry auctions, which incorporate real-time price discovery and public bidding, have benefits that make them more suitable for high-value transactions [24]. These auctions promote a transparent and active bidding environment by allowing participants to compete openly and react to each other's bids. This openness encourages fair competition and empowers participants to make well-informed decisions in accordance with the current state of the market.

Furthermore, open-cry auctioning systems rely on the dynamic calculation of winning bids and their arrival. In centralized systems, the responsibility for calculating, ordering, and ultimately determining the winning bid lies with the centralized institution [9]. When designing decentralized auctioning systems, it becomes crucial to develop robust mechanisms for transaction ordering.

Additionally, in certain scenarios where submission deadlines are strict, achieving time synchronization between interacting parties becomes necessary, e.g., for auctions ending on a hard deadline. Time synchronization ensures that all participants have a consistent understanding of deadlines and can reach a consensus on time-sensitive events within the auction process [16].

III. KEY PROPERTIES FOR DECENTRALIZED AUCTIONING

Based on the quality attributes and the overview of open-cry auction systems presented in Section II, we now define the key properties that need to be supported when decentralized auctioning systems.

A. VERIFIABILITY

In a decentralized auction system, one of the primary requirements is to ensure the authenticity and validity of transactions [25]. Verifiability conveys more than just validating bid submissions; it also involves confirming that the auction rules are followed and that the selection of the transaction winner aligns with these predefined rules. Verifiability is fundamental for establishing trust and integrity in the bidding process. It allows participants to independently verify the validity of transactions and their compliance with predefined auction rules.

B. TRANSACTION IMMUTABILITY

In the context of decentralized auction systems, it is essential to ensure the immutability of transactions once they are verified and validated, whether for a bid or an auction. This property relates to not only the nonrepudiation of bids but also the relation of transactions to those that come either before or after. This immutability serves to enhance the level of trust in the auction process and promote a perception of fairness among participants [9]. Furthermore, in the specific case of open auction formats, it is crucial to ensure equal access to information. This means that no user should have an advantage over others by possessing privileged information. Maintaining information transparency and equal accessibility is vital for creating a level playing field in the auction environment.

C. TRANSACTION ORDERING

In centralized auction systems, the auctioneer typically determines the order in which transactions are processed. However, in decentralized systems, where there is no central auctioneer, the challenge lies in establishing a consensus mechanism among peers to determine the order of transactions. This consensus mechanism is crucial for appending transactions to the blockchain in a synchronized manner [26].

In the context of iterative bidding processes, the ordering of transactions becomes even more critical for the correctness of the auctioning procedure. The winner of the auction is determined by the highest bid; in cases where bids of the same value are submitted at similar times, the order of arrival becomes a deciding factor. The concept of multiple transactions being appended to the blockchain is a prevalent issue in this type of system. The most prevalent way to manage these transactions is by serializing such transactions and determining the order in which they will be appended to the chain.

D. TIME SYNCHRONIZATION

Similar to how transaction ordering is commonly determined by the central auctioneer, the time in which transactions occur is also determined by this entity. The auction deadline is handled differently in various auction procedures. For example, platforms such as Tradera have fixed bid submission deadlines, where there is a predetermined date and time until which bids can be submitted. However, some auction systems employ a dynamically extending deadline approach. In these approaches, if a bid is submitted close to the deadline, then the deadline is extended by a small delta [27].

The use of a dynamically extending deadline reduces the reliance on precise time synchronization, as it extends the deadline instead of requiring precise calculation of submission times. However, in use cases such as Tradera, accurate time synchronization becomes necessary, and achieving consensus in a decentralized environment becomes a crucial requirement. When dealing with concurrent transactions, it is a common practice to employ serialization as a means of managing them [21]. However, this approach may result in the timestamps of bid entries in the blockchain not accurately reflecting the actual initiation times of the transactions, which can adversely impact the perception of fairness of the auction system.

E. PERFORMANCE

Achieving good performance in an auction system is fundamental. A responsive auction system allows potentially large numbers of users to submit concurrent bids and receive updates with little delay. Thus scalability is essential, especially for popular auctions or auctions with time-sensitive deadlines.

Decentralized systems performance is not a singular term. It is a combination of various parameters, and these parameters are interdependent, e.g., block size influences the storage of the system. Block size can affect transaction throughput and latency, which in return can be linked to the choice of consensus model [28]. A larger block size can store more transactions, thus directly raising the throughput, but it also causes an increase in block propagation time.

In the context of this study, the performance of a decentralized open-cry auctioning system depends on the reliable provisioning of transaction ordering and time synchronization, and the selection of consensus models and block propagation settings. For this reason, we believe it is important for additions related to the ordering of transactions and time synchronization to consider additional delays on transactions.

IV. BLOCKCHAIN SYSTEMS

In this section, we provide a brief explanation of blockchain technology and its categorization according to levels of anonymity. Subsequently, we examine popular blockchain frameworks and how they address the identified key properties for electronic open-cry auctions. Finally, we offer insights into the current state of the art (SotA) regarding auction systems implemented using these frameworks.

The analysis of the diverse blockchain frameworks based on their ability to address the identified key properties serves the purpose of identifying the most suitable framework for the PoC implementation provided in this paper. With this analysis, we aim to find gaps within the analyzed frameworks that need to be addressed in the system design.

A. PRIMER ON BLOCKCHAINS

A blockchain is a distributed ledger that maintains a record of ownership for digital items without the need for a centralized entity. The term blockchain refers to the sequential list of blocks within a publicly accessible ledger, which contains all concluded transactions. The blockchain expands as new blocks are consistently added [29]. The concept of a blockchain is characterized by systems that feature (1) inclusivity for users depending on the design level of anonymity, (2) a distributed public ledger for verifying transactions, and (3) a robust and secure consensus protocol to establish trust among participants [30].

Blockchain systems can be divided into multiple types depending on the anonymity of their design.

- Permissionless blockchains: This type of blockchain provides complete transparency in transactions and lacks centralized authority. Here, any user can join the network and view all the transactions. The identities of the participants of the network remain pseudonymous instead of disclosing their real information. They mainly use cryptocurrencies to perform transactions. The main example of this type of blockchain is Bitcoin [31].
- Permissioned blockchain: In permissioned blockchains, one or more core aspects of the network, such as user access and data encryption, are partially controlled by a central entity or consortium. Because of this restriction, users must be identified to participate, and their identities are often known by other peers. This type of blockchain is implemented by Hyperledger Fabric [32] and R3 Corda [33].

By comparing the two blockchain types, we can see that a fundamental difference lies in the level of transaction anonymity. While permissionless blockchains offer transparent access to all transaction information, the identities of the parties involved are often anonymous. In contrast, permissioned blockchains sacrifice decentralization by relying on a centralized entity or consortium, yet they provide authorized and verified identities of participants to strengthen trust during transactions.

B. BLOCKCHAIN TRADEOFFS

Previously, we mentioned that the use of decentralized technologies can leverage some of the pitfalls we find in centralized e-auctioning systems. Blockchain and smart contract technologies have great potential to improve traditional centralized auction models in many fields, as they can create a decentralized, transparent and trustworthy trading environment. For different application scenarios, different researchers have used different auction models and blockchain technologies to handle auctions. Most of these uses are related to energy trading, wireless communication, service allocation and demand-supply matching [16]. When performing transactions that carry high economic value, it is not often desired to keep the identity of interacting parties completely anonymous, as doing so may reduce the overall perception of trust [34]. A balance must be reached so that parties can know that their peers are authorized to perform such transactions, but consumption habits can remain private.

These systems, in conjunction with smart contracts, pave the way for immutable and auditable decentralized auction systems [35]. A smart contract is a blockchain-based program (code) consisting of functions that are triggered by events and have predefined responses to these events. The main purpose of this code is to facilitate, execute and enforce the terms of an agreement [36]. Some blockchain approaches, such as R3 Corda constitute a legally binding contract between the parties [33], in which each party must fulfill its responsibilities [37]. A smart contract and the logic it contains cannot be altered for consistency and immutability purposes.

Despite all the features blockchain systems provide for decentralized auctioning systems, they come with several tradeoffs that need to be taken into account in a design that implements such systems.

A diagram illustrating the blockchain trilemma is shown in Figure 1. This diagram shows three properties of blockchain systems that cannot be implemented simultaneously [38]. As previously discussed in this section, in the development of high-value auction systems, there is a desire to strike a balance between maintaining user identity anonymity and ensuring security through identity management.

Both user registry and access control compromise the decentralization aspect of blockchain systems but offer increased security concerning identity management. On the other hand, opting for open-cry auctioning systems on permissionless blockchains tends to prioritize decentralization without third-party control over identities. However, such an approach can lead to scalability challenges due to the overhead associated with consensus mechanisms [39]. Alternative blockchain approaches such as Hyperledger Fabric do not face challenges related to consensus costs. However, they do have limitations in regard to scalability, as they employ a global consensus model, i.e., all transactions are appended to a global block. While this may not pose a problem in the context of a single auction, it is important to contemplate scenarios involving multiple simultaneous auctions contributing entries to the same total order.

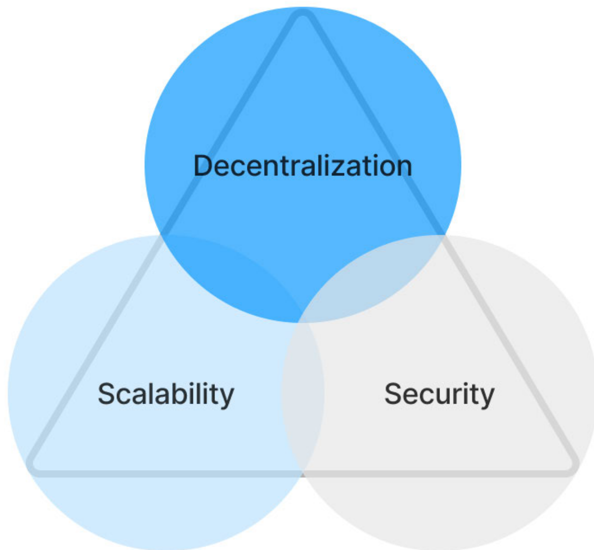


FIGURE 1. **Blockchain trillema diagram. This diagram shows the three properties of blockchain systems that cannot be achieved simultaneously.**

C. POPULAR BLOCKCHAIN FRAMEWORKS

- Ethereum: Ethereum is an implementation of blockchain technology that focuses on enabling the execution of smart contracts. As discussed earlier, smart contracts are self-executing agreements with predefined conditions that cannot be altered once created. Ethereum operates on a permissionless network, providing an open platform for individuals and businesses to engage in economic transactions without the need for authorization or intermediaries [40]. They use proof of stake (POS) as a consensus mechanism. PoS requires members of the network to make contributions to the network from their own holdings of the blockchain's native cryptocurrency as their stake. The right to append a block to the chain depends on their stake in the network.
  Ethereum uses its native cryptocurrency Ether (ETH) as the medium of exchange for value within the network. Furthermore, Ethereum utilizes Solidity as its programming language for writing smart contracts. Ethereum has gained significant popularity and has become the foundation for numerous blockchain-based projects and applications. It also has a thriving ecosystem with a large community of developers and users.

TABLE 1. Desired properties on blockchain approaches.

	Ethereum	Hyperledger Fabric	R3 Corda
Verifiability	Outcome of smart contract verified by all peers	Peers validate and endorse transactions	Outcome verified by involved peers
Transaction immutability	New transactions appended as blocks in the chain	New transactions appended as blocks in the chain	Records transactions as immutable state objects
Transaction ordering	Order is determined by miner upon puzzle solving	Order is determined by ordering service	Order determined by the notary pool
Time synchronization	Timestamps of blocks created by miners	Relies on external time resources and synchronized clocks among peers	Timestamp is created based on time windows
Performance	Avg. latency: 8 sec Avg. Tx/sec: 100	Avg. latency: 2.5 sec Avg. Tx/sec: 200	Avg. latency: 2.5 sec Avg. Tx/sec: 150

- Hyperledger Fabric: In contrast to Ethereum, Hyperledger Fabric is a permissioned blockchain framework specifically designed for enterprise use cases. It is aimed at organizations that seek secure transactions in a trusted network. The use of permissioned blockchains ensures that data are only shared among trusted entities, making it suitable for confidentiality-focused use cases [32].
  The smart contracts implemented in Hyperledger Fabric are called "chaincode". This chaincode allows organizations to define the business logic and rules that govern transactions within the network.
- R3 Corda: R3 Corda, similar to Hyperledger Fabric, is a permissioned blockchain platform that prioritizes privacy and confidentiality. It operates on a "need-to-know" principle, ensuring that transaction data are shared only among the involved parties [33].
  R3 Corda was specifically designed with a focus on financial transactions, which often require the establishment of legal agreements. In addition to smart contracts, R3 Corda incorporates Ricardian contracts, which provide the necessary legal prose for these agreements. This combination allows R3 Corda to effectively address the unique requirements of financial transactions within a secure and trusted environment.
  Several works have been presented that address the implementation of English open-cry auctioning systems using Ethereum blockchain [17], [41], [42]. As previously mentioned in this section, Ethereum operates as a permissionless blockchain. Such blockchains facilitate unrestricted participation and a notable level of decentralization. However, in regard to electronic auctions involving transactions of significant economic value, there is a need to establish a more controlled and restricted environment [9].
  Permissioned blockchain-based English auction systems have also been showcased through the utilization of Hyperledger Fabric [43]. As mentioned in the introduction section, the current state of the art (SotA) is relatively restricted in terms of these methodologies [16]. In the case of R3 Corda, such approaches are primarily relegated to repositories on platforms such as GitHub [44].

D. VERIFIABILITY AND TRANSACTION IMMUTABILITY

In Table 1, we analyze how the different blockchain approaches presented earlier address the properties mentioned in Section III. In this table, we can see that in regard to the verifiability and immutability of transactions, all of the presented blockchain approaches share similarities. These properties are supported in similar ways regarding the features that are inherent in blockchain. This is one of the reasons why blockchain approaches are often desired for the design of decentralized applications.

E. TRANSACTION ORDERING

When considering transaction ordering and time synchronization, it is worth noting that the evaluated blockchain approaches employ distinct methods to support these features as seen in Table 1. Ethereum as a permissionless blockchain implements the concept of Proof of Stake to determine the order and time of the transactions. It manages concurrent transactions by serializing them instead of attempting to run them concurrently [45]. Contrary to Ethereum, R3 Corda relies on the notary pool service to determine the order of transactions based on the input states, i.e., the hash of the ledger status when the transaction is created.

According to its design, Hyperledger Fabric includes a dedicated service that utilizes consensus mechanisms, e.g., Raft, to determine the order of transactions. This mechanism ensures a reliable and consistent transaction ordering process within the network. Hyperledger Fabric does not support concurrency [20]. In this blockchain system, concurrent transactions that have read-write conflicts on the state of the world result in error. In the case of a mismatch between read and write sets of concurrent transactions, all transactions but one will fail. This happens because transactions in Hyperledger Fabric are atomic; i.e., transactions either succeed entirely or fail entirely and are not included in the block.

F. TIME SYNCHRONIZATION

Time synchronization has been recognized as a significant concern in decentralized systems [46]. Ensuring accurate time synchonization is important for optimizing

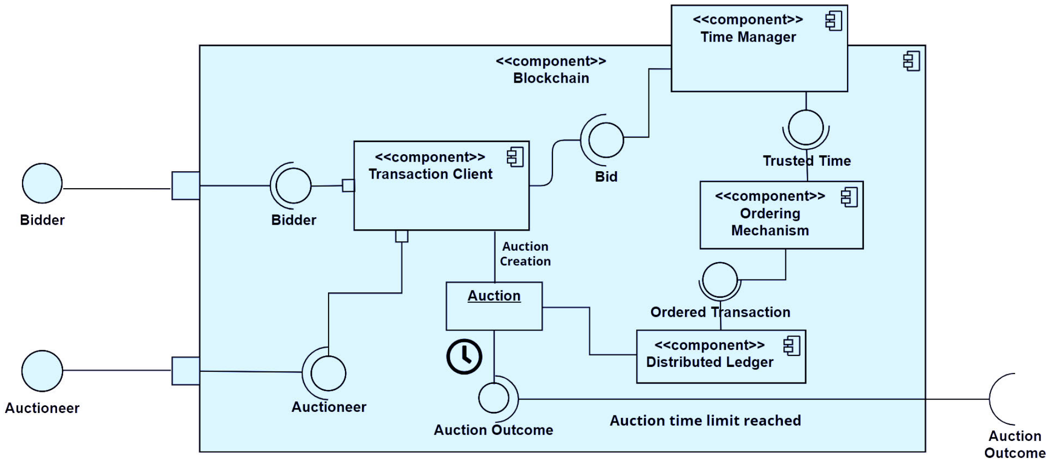


FIGURE 2. **Component diagram of the proposed solution design.**

power-efficient duty cycling, coordinating scheduled operations, and particularly in scenarios such as auctions featured in this study, where adherence to strict deadlines is crucial [47]. In [48], a method to improve the security of Bitcoin timestamps is proposed. This system uses external authorities to assert block creation time. In a permissionless blockchain that is intended to be fully decentralized, this poses a compromise in regard to political decentralization. In Table 1, we can see how different blockchain approaches address time synchronization.

Permissionless blockchain technologies such as Ethereum have a difficult time synchronizing previous block creation times between nodes, as blocks in Ethereum are produced periodically and are ultimately controlled by peers [49]. R3 Corda implements the concept of time windows; this window establishes a time in which a given transaction can be validated by the notary pool [50].

Hyperledger Fabric, by design, relies on the synchronization of clocks among the participating peers. The timestamp for transactions is set by the submitting peer and it is assumed that peers have synchronized clocks. There is a gap in establishing the notion of a trusted time in Hyperledger. Most applications rely on the ordering service to append transactions in order of arrival. However, by lacking a trusted notion of time, time-sensitive applications need to rely on non-malicious nodes within the blockchain.

Several approaches have been presented to address time synchronization in Hyperledger Fabric. Timestamp [16] introduces a consistent notion of time across nodes, allowing timestamps to be assigned at transaction validation. In [46], a global clock model is proposed. In this approach, "clock nodes" obtain trusted timestamps from different sources and use them to achieve consensus when a new timestamp is requested.

G. PERFORMANCE

A performance evaluation on permissioned blockchain platforms was previously performed [51]. The performance analysis findings across various evaluation metrics, including throughput and latency, demonstrate that Hyperledger Fabric outperforms other permissioned platforms. The latency of Ethereum is hindered by the requirement of achieving consensus between all peers when a transaction is created.

In auction systems where low latency is crucial for competing peers, Hyperledger Fabric can be seen as the optimal implementation choice. Nonetheless, it's worth noting that the latency of blockchain systems, including Hyperledger Fabric, can still be relatively high in highly competitive environments.

V. SOLUTION DESIGN AND IMPLEMENTATION

In this section, we introduce the solution design for a decentralized auction system aimed at fulfilling the key properties outlined earlier in this paper. This design serves as a roadmap for implementing a PoC system. Subsequently, we detail a PoC implementation based on this design, using Hyperledger Fabric for its decentralized features. Furthermore, we complement this blockchain solution with additional components to address the gaps identified in Section IV.

A. PROPOSED SOLUTION

The solution that we propose in this paper is intended to cover the quality attributes for the design of decentralized auctioning systems. We address them considering the implementation of the key properties presented in Section III.

A component diagram for the proposed solution design is depicted in Figure 1. The elements present in this figure can be summarized as follows:

- A robust blockchain system that is capable of recording and linking transactions, ensuring a reliable and transparent audit trail [9]. Transaction ordering is crucial to determine the accurate outcome of the auction.
- Smart contract logic that is specifically tailored to manage the auction process. These smart contracts play a vital role in automating auction rules and ensuring fair and reliable execution.

- Time synchronization capabilities are an important part of our implementation. To achieve this, we rely on an external time resource that facilitates a trusted and synchronized notion of time among peers. This is important in scenarios where synchronizing with real-world time is important, e.g., bids submitted close to the auction deadline. This component can be implemented as part of the blockchain component or external to it [48].

Based on the blockchain preliminaries presented in Section 1, we find that permissioned blockchain is an adequate approach when we consider high-value transactions in a trusted environment. The use of permissionless blockchain would allow for more costly transactions to take place when this is not required. Additionally, ensuring that the users interacting are identified can have a positive impact on the auction outcome. Ensuring that users are identified does not imply that the identity has to be disclosed to interacting users so that the bid-bidder relationship remains concealed [34].

The choice of using a permissioned blockchain for this solution concept is, in part, made for the features that these technologies provide by design. Referring to the desired properties identified in Section III, the concept solution does not provide additional enhancements to the blockchain technology. The system concept focuses on the establishment of smart contract logic and additional external sources to manage time synchronization features.

B. PROOF OF CONCEPT IMPLEMENTATION

For this proof-of-concept, we use Hyperledger fabric as the implementation choice for a robust blockchain system which is capable of recording transactions following the proposed solution design. This blockchain system provides the smart contract logic required for the automation of the auction rules. In this PoC we use several of the features provided by Hyperledger Fabric to address several of the desired properties mentioned in Section III. However, we find some elements that cannot be properly addressed with these existing features, one of which is the concept of time synchronization.

Table 1 illustrates that various blockchain approaches for auctioning systems show fairly similar approaches when addressing the initial properties, i.e., verifiability, transaction immutability and performance. In our opinion, Hyperledger Fabric provides better features for transaction ordering than the other evaluated approaches by the introduction of the ordering service and the lowest latency. As we mentioned earlier in this paper, if the auction system has a strict deadline, then any bid that arrives after that time is considered invalid. With the features provided by Hyperledger Fabric, bids that arrive simultaneously with similar timestamps shall be ordered according to the ordering service.

We add a link to the git folder for further reference for implementation and functionality.[1]

[1]E. Chiquito, HLF auction. URL (accessed 2023-07-08): https://github.com/EricChiquitoG/HLF-Open-Cry-Auction-System.

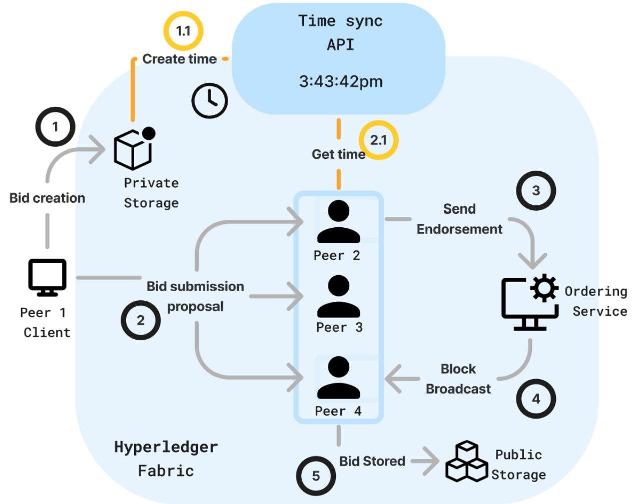


FIGURE 3. Representation of components present in the proof-of-concept system architecture.

In Figure 3, we present the components of the proposed solution. We use the core components of Hyperledger Fabric to cover some of the desired properties for decentralized auctioning systems. We embed an additional component external to the Hyperledger Fabric architecture to achieve time synchronization.

As seen in the aforementioned image, the overall core functionality of Hyperledger Fabric has not been modified, as we believe that the core components present in the system serve well for the intended purpose of the tool. The system modifications take place when the bid is created by the submitting peer in Step 1 and during the endorsement execution of the bid submission. All peers that endorse a *submit bid* transaction submit requests to such API.

1) AUCTION CHAINCODE

The proposed solution contains several executable smart contracts:

- Create auction: This chaincode is to be executed by the peer owner of the auction. This chaincode creates an action with a set time limit.
- Create bid: This chaincode is to be executed by peers who are not the owners of the auction. The creation of the bids sets the offered price, creates an ID for such bid and retains data privacy; i.e., this data is not public or visible by any other peer. In the execution of this chaincode, a POST request is performed on the external API to establish a set of trusted timestamps. This transaction only has to be endorsed by a single party, unlike a submit bid transaction; for this reason, this transaction has no R/W conflicts if executed concurrently and can be used to establish the bid timestamp.
- Submit bid: This chaincode refers to the ID of a previously created bid and has to be executed by that peer. In addition, it must be endorsed by the other interacting peers. This chaincode initiates a GET request

to an external API to retrieve a list of trusted timestamps. Subsequently, it generates an integer hash value based on the transaction ID (txID). This hash value is employed to shuffle the timestamps in a deterministic manner, ultimately selecting a timestamp for utilization.

Algorithm 1 Create Auction Smart Contract

```
function CREATEAUCTION(auID, item, timelimit)
    Read invoke values
    Obtain userID
    if timelimit < time.now() then
        Create auction struct
        auction.item ← item
        auction.auID ← auID
        auction.seller ← userID
        auction.timelimit ← timelimit
        auction.winningbid ← 0    ▷ Current implementation initializes price on 0
        Save auction in public state
    else
        Error: Auction could not be created
```

Algorithm 2 Create Bid Smart Contract

```
function CREATEBID(auID, price)
    Read invoke values
    Obtain userID
    Read auction data
    if auction.seller != userID then
        Create bid struct
        bid.bidder ← userID
        bid.price ← price
        POST request to external API to determine bid creation time
        Save bid into private state
        Return bidID
    else
        Error: Bids cannot be created by auction owner
```

Algorithm 3 Submit Bid Smart Contract

```
function SUBMITBID(auID, bidID)
    Read invoke values
    Obtain userID
    Read auction and bid data
    if bid.bidder == userID then
        GET request to external API to obtain List_TTS
        Create a hashValue of the bidID to get a number from 1 to 10
        shuffleSeed ← hashValue
        Order and shuffle List_TTS according to shuffleSeed
        timeStamp ← first element in shuffled list
        if timeStamp < auction.timelimit then
            bid.timestamp ← timeStamp
            Save bid into public state
            if bid.price > auction.winningbid then
                auction.winningbid ← bid.price
        else
            Error: Bid invalid, auction time limit reached
    else
        Error: Bids can only be submitted by the owner
```

These chain codes provide the basic functionalities needed in English auction systems. While the proposed solution addresses only English auction systems, the system can be further extended to support other types of auctioning approaches, such as those in Dutch.

2) EXTERNAL COMPONENT

An external Flask API, deployed on a docker container, plays a crucial role in the system. Its primary function is to register NTP (network time protocol) timestamps of several NTP servers when the chaincode for bid creation is executed. These timestamps serve as trusted records of when the bids are initialized.

Later, during the endorsement of submitted bids, peers initiate GET requests to acquire the trusted timestamps recorded in the previous step. These trusted timestamps are essential for validating and ensuring the integrity of the submit bid process. The detailed functionality of these requests is further shown in the subsequent algorithms.

Algorithm 4 External API POST Request

```
function NEWBID(bidID)
    Receive POST request
    Connect to MongoDB database
    for <server in trusted NTP servers> do
        Connect to NTP server to obtain trusted timestamp TTS
        List_TTS ← [server.TTS]
    Create bidTimes object to be stored
    bidTimes.address ← peer.address
    bidTimes.bid ← bidID
    bidTimes.time ← List_TTS
```

Algorithm 5 External API GET Request

```
function GETTIME(bidID)
    Receive GET request
    Connect to MongoDB database
    Get bidTimes object from Database
    Return(bidTimes.List_TTS)
```

The objective of incorporating this external component is to introduce the notion of trusted time into Hyperledger Fabric. This is achieved through minimal integration into the chaincode execution, avoiding an extensive overhaul of the existing fabric modules. The transaction execution in chaincode contains multiple steps, providing an opportunity for peers to obtain and secure trusted timestamps.

The requests sent to the external API, as mentioned in previous sections, are structured to minimize additional delays on fabric components. Any potential delays to the system may occur if the centralized component becomes unavailable.

C. IMPLEMENTATION ASPECTS

The Hyperledger Fabric framework provides some of the most basic functionalities of a blockchain, as presented in Section IV. By using a permissioned blockchain, we can offer a certain level of security; i.e., interacting users can trust that the interacting peers are authorized to interact in the network. The network contains a record of all transactions appended into the chain in the form of the blockfile. These appended transactions can be verified by the members of the network who are enrolled to participate in a particular transaction.

As part of our design, we opt for an external element to handle time synchronization. The applicability of assigning trusted timestamps in Hyperledger Fabric and other permissioned blockchain systems is constrained by their concurrency limitations [52]. These limitations derive from the methods employed to manage concurrency, which often involve the use of locks or transaction serialization. When multiple bids are submitted concurrently, they are processed sequentially. Consequently, the timestamp associated with each bid may not accurately reflect the actual time at which it was submitted by the user.

By using the ordering service, Hyperledger Fabric removes the responsibility for ordering transactions from the interacting peers. Instead, it employs an ordering service that utilizes consensus mechanisms to establish an accurate sequence of transactions. This system does not organize the transactions appended into the system by time only but rather by the order in which transactions arrive to the ordering peers.

The PoC implementation was created from Hyperledger Fabric samples,[2]

The chaincode manages the logic related to how transactions are executed, which provides basic functionalities needed by English auctioning systems. The only addition to this auction logic involves integrating the external API request functionality for time synchronization tasks. The requests submitted to such external API for the sake of time synchronization are designed with the purpose of the chaincode in mind, which is the quick execution of tasks by the endorsing peers. Minimizing the computational complexity and blocking any additional task, as described in Algorithms 5 and 6.

VI. SYSTEM ANALYSIS

A. EXPERIMENTAL SETUP

In this section we present 5 experiments. experiments 1 and 2 are aimed to evaluate the performance of the external API service created for this paper were created using a set of four peers in each auction. Each peer represent a single company (Org1, Org2, Org3 and Org4) and have the following roles: comp1 → Auctioneer and (Org2, Org3, Org4) → Bidder. The external API component for time synchronization was deployed as a Docker container inside the same network as HLF. Experiment 3 is designed to evaluate the latency of the time synchronization system with concurrent requests independently. Finally, an additional set of experiments, experiments 4 and 5, were performed with the purpose of testing the performance of the blockchain system in the two main transactions: bid creation and bid submission. The aforementioned experiments consist of 400 total sequential bids created and submitted by 4, 8 and 12 peers.

The orderer settings were kept as default.

- OrdererType: Solo
- BatchTimeout: 2s
- BatchSize: 10 Mb

The specs of the PC used for the experimental analysis are the following.

- Processor: Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz

[2]Hyperledger Fabric samples. URL (accessed 2023-07-08): https://github.com/hyperledger/fabric-samples.

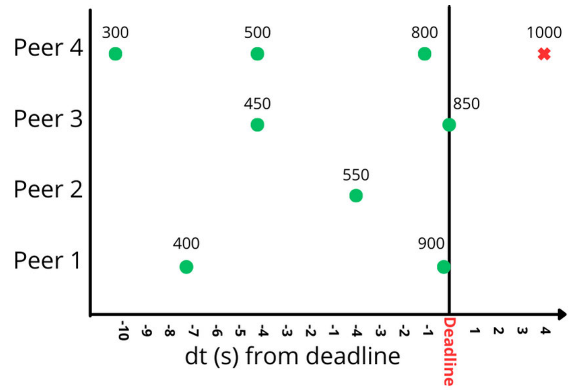


FIGURE 4. Experiment 1: This experiment shows bids being created using the proposed model.

- RAM: 24GB
- System type: 64-bit
- Operating System: WSL with Ubuntu 20.04
- Memory WSL: 4GB
- Processors WSL: 3

The settings of the orderer may affect the overall transaction throughput and latency of this implementation. While this may have an impact on the concurrency of transactions, this is not the main concern of Hyperledger fabric as we will cover in the next Subsection.

B. EXPERIMENTAL VALIDATION OF TIME SYNCHRONIZATION SOLUTION

These experiments aim to recreate edge cases where ordering and time synchronization may be needed. Due to the limitations of Hyperledger Fabric regarding the concurrency of R/W operations, the experiments that require accessing public data are executed sequentially. In the first experiment, bids are created close to the deadline and executed concurrently. As seen in Figure 4, the bid timestamps are created concurrently when the bid is created, resulting in the same timestamp. The validity of the transactions can also be seen as bids before the deadline are marked as not valid while those submitted afterwards are counted as valid. Bids that have the same value as others in the same timestamp are counted as valid, as the order is not determined at this step. It is important to note that bids that are invalid based on deadlines do not fail; instead, they are marked as nonvalid and appended to the chain. The purpose of this approach is to create a ledger that can be checked by users and reviewed in case of disputes.

The endorsing process, which is performed by endorsing peers, involves capturing the read sets of the bids. In the context of the auction system we designed, the read set of a bid transaction includes key-value pairs related to both the auction itself and the bid being submitted. The parallel execution of submitted bid transactions is not feasible when two transactions are simultaneously submitted for the same auction. In such cases, Hyperledger Fabric detects inconsistencies between the read and write sets, resulting in

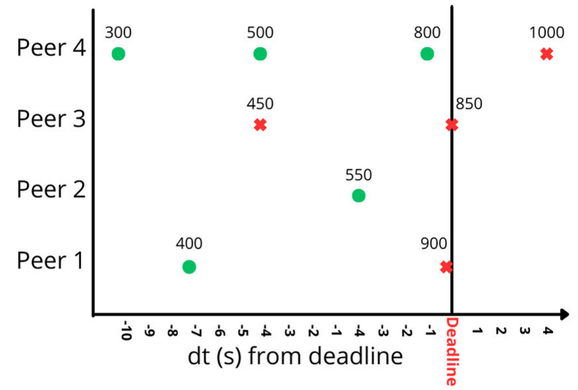


FIGURE 5. Bid submission behavior in the case of concurrent transactions.

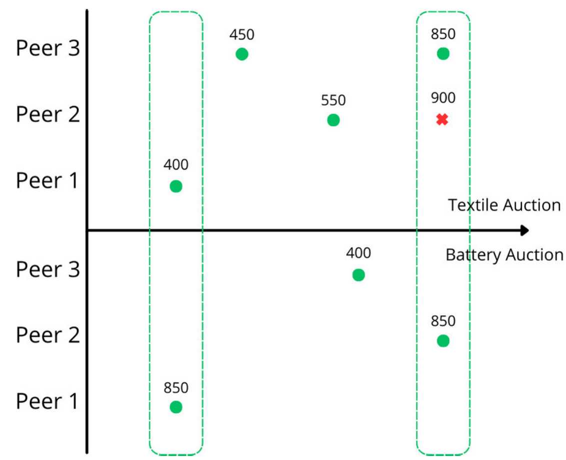


FIGURE 6. Results of experiment 2 with multiple bids being submitted in different auctions.

the failure of all concurrent transactions except one, as seen in Figure 5. Transactions in Hyperledger Fabric are atomic; i.e., they either succeed entirely or fail entirely. If a transaction fails during the endorsement phase, it will not be included in the block.

However, as the read set of the bid relates to the auction to which it belongs, the execution of a "submit bid" transaction can occur concurrently for two different auctions, as seen in the second experiment. This experiment attempts to evaluate the capabilities of the system to process serial transactions occurring in multiple auctions at the same time, i.e., bids that are executed serially inside an auction but at the same time as in other auctions. The initial expectation is that read-write conflicts in concurrent transactions will primarily occur when transactions reference the same auction. The results displayed in Figure 6 confirm this assumption. As mentioned in Section IV.B, all transactions created in Hyperledger Fabric are appended to a global block, even if transactions such as bids belong to different auctions. This poses a limitation in scalability if multiple auctions occur in the same blockchain system.

The outcomes of Experiment 3, depicted in Figure 7, focus on assessing the latency of the time synchronization

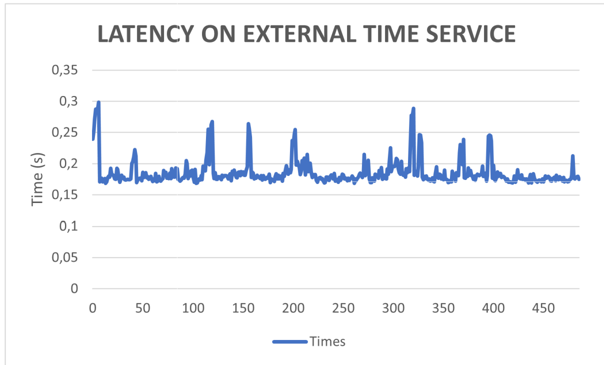


FIGURE 7. Sample of the latency in seconds of external time service.

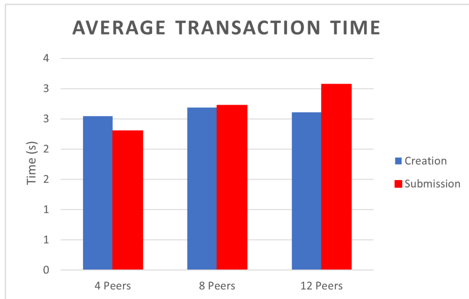


FIGURE 8. Average transaction time for bid submission and bid creation with different number of peers.

service independently from the blockchain system. In this experiment, we executed 500 simultaneous requests to evaluate the performance of the time synchronization system. The findings reveal an average latency of 0.32 seconds. However, it's worth noting that the average latency was affected by outliers that exceeded 1 second. Out of the 500 concurrently sent requests, 15 exhibited a latency exceeding 1 second.

C. EXPERIMENTAL EVALUATION OF THE BLOCKCHAIN SYSTEM

These experiments evaluate the performance of the blockchain system for the two main transactions of this PoC: bid creation (Algorithm 2) and bid submission (Algorithm 3). For these experiments we use a set of 4, 8 and 12 bidders and one auctioneer.

In Figure 8 we present the average transaction time for both operations depending on the number of peers. Here we identified that the average transaction time for bid submission increases as we increase the amount of peers involved in an auction. However as we will analyze in the following experiments, the time for a bid submission transaction to be finished also depends on the amount of bids on the block.

As mentioned in Section V, bid creation transaction is not public, andtherefore is not endorsed by multiple parties nor is appended to the global block. For this reason, we expect the transaction time to remain constant regardless of the amount of bidders and bids.

The results of Experiment 4 shown in Figure 9 support the initial assumptions. Regardless of the amount of bidders

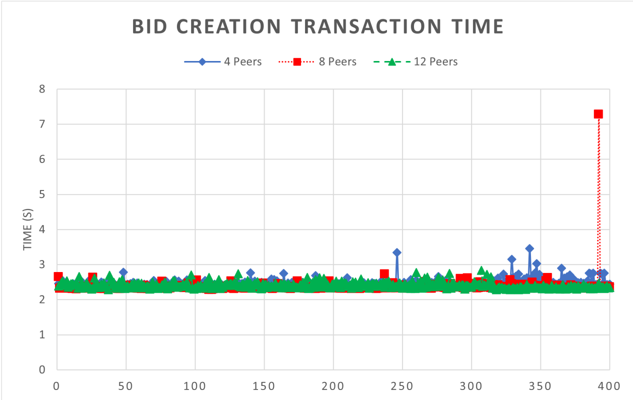


FIGURE 9. Transaction time for the bid creation transaction.

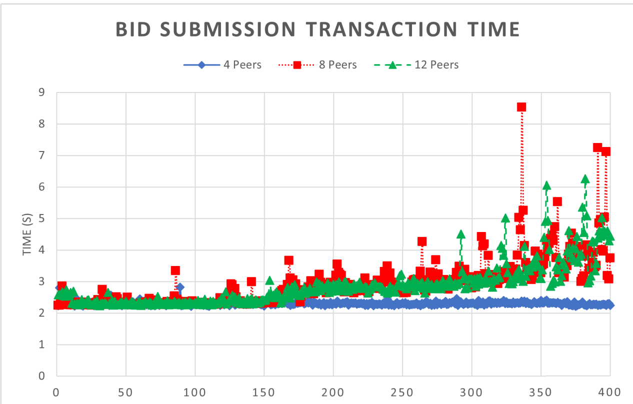


FIGURE 10. Transaction time for the bid submission transaction.

and bids created the transaction time does not increase significantly.

Experiment 5 shown in Figure 10 evaluates the bid submission transaction time. This transaction involves the endorsing of transactions by multiple peers and the appending of such transactions into the global block. In this experiment we expect the transaction time to increase as the number of bidders and amount of bids for each auction increase due to the consensus time required and the block size involved.

The results of such experiment confirm that while the time to submit a bid remains similar at the beginning of the auction, as we increase the amount of bids and peers endorsing transactions, the time to append a new bid also increases. This is due to the addition of valid transactions to the increasing world state.

In both experiments 4 and 5, outliers impacting transaction time are observed, attributed to delays in the external API service as discussed earlier in this section. These delays, coupled with service unavailability, pose vulnerabilities that could impact the system's reliability.

D. SUMMARY

In this paper, we analyzed the performance of the external time service in isolation and the transaction time of the PoC system using HLF. We consider the delays presented by the external service to be non-significant to its performance despite the outliers presented in Figures 9 and 10. The evaluation performed with several transactions submitted by multiple peers also showed that the transaction latency is not constant, i.e., it does not remain constant either by amount of peers or number of transactions. This variability steems from Hyperledger Fabric's requirement for all peers to read the entire block of transactions, a process that becomes increasingly time-consuming as the number of users and transactions grows.

We use the external service to create a trusted notion of time and to prevent the delay presented by the transaction submission to influence the bid submission time. This additional service ensures that each bid accurately reflects its submission time, even if the transaction is confirmed in the block after the auction concludes.

VII. OTHER RELATED WORK

A. NONBLOCKCHAIN DECENTRALIZED APPROACHES

Nonblockchain auctioning systems have been demonstrated using distributed agent architectures [53] and distributed hash tables [54], [55]. Among these, those with a focus on open-cry auctioning are restricted [53]. While these approaches attend to verifiability and transaction immutability, they do not address properties such as transaction ordering and time synchronization.

An alternative method for decentralized time synchronization, unrelated to blockchain, is demonstrated in [56]. This system employs a gossip-based approach to synchronize time across peers. In this scheme, interacting peers disseminate information to their neighbors using a gossip protocol and conform to their neighbor's time until a consensus is reached. Iterations are reduced as the time differences approach a minimum.

B. CENTRALIZED APPROACHES

Centralized auctioning platforms typically address the properties of verifiability, transaction immutability, transaction ordering, and time synchronization through centralized control and management. Transactions are commonly submitted to centralized or physically distributed databases. These auctioning systems are broadly available and popular, as seen in platforms such as Tradera [23] and eBay [57]. Other services such as [58] and [59] allow companies or users to host their auctions in B2B contexts. These services provide users with an interaction platform and data allocation options.

VIII. DISCUSSION AND FUTURE WORK

As discussed in Section I and further reviewed in the previous section, there are several approaches that can be taken with regard to the design of open-cry auctioning approaches. The most commonly known is the use of centralized platforms to conduct these transactions. However, these platforms also come with certain drawbacks, such as a single point of control, potential for bias, and a lack of transparency.

While non-blockchain systems have addressed some of the key properties outlined in this paper, we believe that blockchain offers a suitable solution thanks to its inherent design characteristics. The implementation presented in this paper emphasizes the limitations of blockchain systems in managing open-cry auction systems, especially in regard to concurrent submission of bids. The proposed design and PoC implementation presented in Section V demonstrates a means to overcome these limitations.

Hyperledger Fabric was selected for this implementation due to its versatile and modular architecture, which enables the seamless integration of various functionalities, e.g., the execution of smart contracts to trigger on-chain and off-chain events. However, we acknowledge that other permissioned blockchain frameworks, such as R3 Corda, offer alternative implementation routes and should not be disregarded. We consider blockchain approaches as suitable for auction implementations for the features they provide. A permissioned blockchain was chosen in this work for the anonymity and security features it provides compared to its permissionless counterpart.

The chaincode responsible for managing the logic of this PoC has been designed for efficient execution by peers, aimed at preventing delays. The delays possible in this system may be caused by the unavailability of the external service and it is not expected to cause significant performance issues when it comes to time synchronization. The transaction times obtained in these experiments, including the external time service are in line with the update transaction latency obtained in other Hyperledger Fabric approaches [60]. In Section VI we identified that Hyperledger fabric may present a vulnerability regarding scalability with an increasing number of peers, resulting in higher transaction times. The performance of this PoC system can also be influenced by the hardware of the machine used to run the experiments.

The ledger data structure of Hyperledger Fabric contributes to maintaining a record of transactions. This record contributes to covering the identified properties of verifiability and transaction immutability. Each block appended to the ledger contains the signature of the submitting peer and relates to the previous block. This provides a tamper-proof history of all committed transactions that must be consistent across all nodes, making it difficult to alter or delete transactions.

The ordering service component serves its purpose to order serially submitted transactions. This component prevents concurrent transactions from being executed, altering the sequential structure of the chain. It is debated whether this ordering service can be considered fair. The order in which two transactions are incorporated into the chain might be influenced by factors such as the physical proximity and network performance of peers to the ordering service. This, however, can be leveraged with the use of a cluster set of distributed nodes instead of a single ordering node.

The proof-of-concept implementation incorporates API calls linked to a centralized database for collecting NTP timestamps. While this approach ensures secure time synchronization, it may deviate from the decentralized nature of blockchain architecture. This concept design serves as a blueprint for implementation in different blockchain systems. The choice of blockchain depends on the use case in which the auction system is to be implemented.

Concurrently submitting transactions in Hyperledger Fabric poses significant complexities. We can see why Hyperledger Fabric is used mostly for the creation of sealed auction systems given its limitation of decentralized iterative interaction systems. As previously discussed, the typical workarounds for addressing concurrency challenges involve implementing locks, often utilizing Redis for transaction serialization. However, these measures are essentially patches rather than fundamental solutions to inherent concurrency issues.

The transaction serialization allows transactions to be eventually processed by all nodes. This allows for conflict resolution in terms of which transaction was received first by the ordering nodes which will determine the correct order of transactions. However, when it comes to auction deadline, the serialization of transactions poses delays oi transaction processing from 1 to 3 seconds in the current state of implementation. This would prevent transactions from being considered valid even when submitted on time in case of several scheduled transactions.

We consider the PoC approach proposed in this paper to be a workaround that aims to establish a correct time even if transactions are serially executed and appended to the chain. This is accomplished by establishing a trusted timestamp for the bid during creation, allowing concurrent execution and subsequent addition to the chain upon bid submission. This approach can be extended to take into account the propagation delay produced by the requests to NTP servers using algorithms such as Cristian's algorithm [46].

Alternative methodologies such as Timefabric [16] and the global clock model [46] introduce additional modules into the Hyperledger Fabric architecture. Similar to the approach presented in this paper, the modifications performed to Hyperledger Fabric add little overhead to the transaction processing time. Considering the modular structure of Hyperledger Fabric, we deem this approach to be the most suitable path toward the development of an open-cry auctioning system using Hyperledger Fabric.

IX. CONCLUSION

This paper has focused on the analysis of key properties for decentralized electronic auctions. Initially, we identified the essential quality attributes that should be provided by design. Subsequently, we identified the necessary components to be implemented in a proof-of-concept to fulfill these properties. Furthermore, we analyzed the most popular blockchain systems in the scope of the identified properties for gap identification.

We determined that blockchain systems are suited for implementing open-cry auction systems. The different systems address the identified properties in different ways, however, they need to be complemented by additional features to fulfill such properties. A qualitative and quantitative evaluation was conducted on the entire Blockchain system, as well as on the external time service in isolation.

The scalability issues observed in the experimental Section are inherent to the Hyperledger architecture itself and are not a result of the modifications introduced in this paper. We consider the delays presented by the external service to be non-significant to its performance.

In summary, the current SotA in decentralized open-cry auctioning is fairly limited compared to sealed approaches, primarily due to challenges in ensuring transaction ordering and accurate time synchronization. This paper highlights the gaps in popular blockchain systems and suggests a PoC implementation to tackle the issue of time synchronization. The primary contributions of this paper lie in identifying gaps in current decentralized auction approaches and proposing a PoC solution for time synchronization.

ACKNOWLEDGMENT

The authors thank Andre Rooas, Peggy Khialie, Sebastian Lindgren, and Emil Lock for their contributions to the initial development of this proof of concept.

REFERENCES

[1] W. Chen and F. Lei, "A simple efficient electronic auction scheme," in *Proc. 8th Int. Conf. Parallel Distrib. Comput., Appl. Technol. (PDCAT)*, Dec. 2007, pp. 173–174.

[2] R. Cassidy, *Auctions and Auctioneering*. Berkeley, CA, USA: Univ. California Press, 1967.

[3] S. Thakur, B. P. Hayes, and J. G. Breslin, "Distributed double auction for peer to peer energy trade using blockchains," in *Proc. 5th Int. Symp. Environ.-Friendly Energies Appl. (EFEA)*, Sep. 2018, pp. 1–8.

[4] J. Martins, M. Parente, M. Amorim-Lopes, L. Amaral, G. Figueira, P. Rocha, and P. Amorim, "Fostering customer bargaining and E-procurement through a decentralised marketplace on the blockchain," *IEEE Trans. Eng. Manag.*, vol. 69, no. 3, pp. 810–824, Jun. 2022.

[5] I. A. Omar, H. R. Hasan, R. Jayaraman, K. Salah, and M. Omar, "Implementing decentralized auctions using blockchain smart contracts," *Technol. Forecasting Social Change*, vol. 168, Jul. 2021, Art. no. 120786. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0040162521002183

[6] Y.-H. Chen, S.-H. Chen, and I.-C. Lin, "Blockchain based smart contract for bidding system," in *Proc. IEEE Int. Conf. Appl. Syst. Invention (ICASI)*, Apr. 2018, pp. 208–211.

[7] V. Hassija, V. Saxena, and V. Chamola, "Scheduling drone charging for multi-drone network based on consensus time-stamp and game theory," *Comput. Commun.*, vol. 149, pp. 51–61, Jan. 2020, doi: 10.1016/j.comcom.2019.09.021.

[8] T. H. N. H. von der Fehr, M. M. E. Maasland, and M. H. Rothkopf, *Paul Klemperer: Auctions: Theory and Practice*. Princeton, NJ, USA: Princeton Univ. Press, 2004.

[9] Z. Shi, C. de Laat, P. Grosso, and Z. Zhao, "Integration of blockchain and auction models: A survey, some applications, and challenges," 2021, *arXiv:2110.12534*.

[10] T. D. T. Nguyen and M. T. Thai, "A blockchain-based iterative double auction protocol using multiparty state channels," *ACM Trans. Internet Technol.*, vol. 21, no. 2, pp. 1–22, Mar. 2021.

[11] H. T. Doan, J. Cho, and D. Kim, "Peer-to-peer energy trading in smart grid through blockchain: A double auction-based game theoretic approach," *IEEE Access*, vol. 9, pp. 49206–49218, 2021.

[12] W. Liu, H. Wu, T. Meng, R. Wang, Y. Wang, and C.-Z. Xu, "AucSwap: A vickrey auction modeled decentralized cross-blockchain asset transfer protocol," *J. Syst. Archit.*, vol. 117, Aug. 2021, Art. no. 102102. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762121000813

[13] J. J. Deshpande, M. Gowda, M. Dixit, M. S. Khubbar, B. S. Jayasri, and S. Lokesh, "Permissioned blockchain based public procurement system," *J. Phys., Conf.*, vol. 1706, Dec. 2020, Art. no. 012157, doi: 10.1088/1742-6596/1706/1/012157.

[14] E.-O. Blass and F. Kerschbaum, "Strain: A secure auction for blockchains," in *Computer Security*, J. Lopez, J. Zhou, and M. Soriano, Eds. Cham, Switzerland: Springer, 2018, pp. 87–110.

[15] P. Lafourcade, M. Nopere, J. Picot, D. Pizzuti, and E. Roudeix, "Security analysis of auctionity: A blockchain based e-auction," in *Foundations and Practice of Security*, A. Benzekri, M. Barbeau, G. Gong, R. Laborde, and J. Garcia-Alfaro, Eds. Cham, Switzerland: Springer, 2020, pp. 290–307.

[16] A. Mitra, C. Gorenflo, L. Golab, and S. Keshav, "TimeFabric: Trusted time for permissioned blockchains," in *Proc. Int. Symp. Found. Appl. Blockchain*, 2021, pp. 1–15. [Online]. Available: https://api.semanticscholar.org/CorpusID:235365262

[17] A. Tezel, P. Febrero, E. Papadonikolaki, and I. Yitmen, "Insights into blockchain implementation in construction: Models for supply chain management," *J. Manag. Eng.*, vol. 37, no. 4, Jul. 2021, Art. no. 04021038.

[18] E. Chiquito, U. Bodin, and O. Schelén, "Survey on decentralized auctioning systems," *IEEE Access*, vol. 11, pp. 51672–51688, 2023.

[19] A. Sampath and C. Tripti, "Synchronization in distributed systems," in *Advances in Computing and Information Technology*, N. Meghanathan, D. Nagamalai, and N. Chaki, Eds. Berlin, Germany: Springer, 2012, pp. 417–424.

[20] L. Xu, W. Chen, Z. Li, J. Xu, A. Liu, and L. Zhao, "Solutions for concurrency conflict problem on hyperledger fabric," *World Wide Web*, vol. 24, no. 1, pp. 463–482, Nov. 2020.

[21] T. Dickerson, P. Gazzillo, M. Herlihy, and E. Koskinen, "Adding concurrency to smart contracts," in *Proc. ACM Symp. Princ. Distrib. Comput.* New York, NY, USA: Association for Computing Machinery, Jul. 2017, p. 303, doi: 10.1145/3087801.3087835.

[22] A. Brilz, *The Online Auction Market EBay. Why is Online Auction a Perfect Competition Market?* Grin Verlag, 2020.

[23] G. Johansson, "Consumer online resale at tradera: A qualitative study of valuation and pricing in the online auction marketplace," Dept. Sociol., Uppsala Univ., p. 53, 2018.

[24] J. Ahlberg, "Multi-unit common value auctions: Theory and experiments," Örebro Univ., School Bus., p. 22, 2012.

[25] J. Dreier, H. Jonker, and P. Lafourcade, "Defining verifiability in e-auction protocols," in *Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secur.* New York, NY, USA: Association for Computing Machinery, 2013, pp. 547–552, doi: 10.1145/2484313.2484387.

[26] S. Mu, Y. Cui, Y. Zhang, W. Lloyd, and J. Li, "Extracting more concurrency from distributed transactions," in *Proc. USENIX OSDI*, 2014, pp. 479–494.

[27] A. E. Roth and A. Ockenfels, "Last-minute bidding and the rules for ending second-price auctions: Evidence from eBay and Amazon auctions on the internet," *Amer. Econ. Rev.*, vol. 92, no. 4, pp. 1093–1103, Aug. 2002. [Online]. Available: https://www.aeaweb.org/articles?id=10.1257/00028280260344632

[28] M. Gracy and B. Rebecca Jeyavadhanam, "A systematic review of blockchain-based system: Transaction throughput latency and challenges," in *Proc. Int. Conf. Comput. Intell. Comput. Appl. (ICCICA)*, Nov. 2021, pp. 1–6.

[29] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data*, Jun. 2017, pp. 557–564, doi: 10.1109/bigdatacongress.2017.85.

[30] T. Aste, P. Tasca, and T. Di Matteo, "Blockchain technologies: The foreseeable impact on society and industry," *Computer*, vol. 50, no. 9, pp. 18–28, 2017.

[31] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," vol. 4, no. 2, p. 15, 2008. [Online]. Available: https://bitcoin. org/bitcoin.pdf

[32] (2015). *Hyperledger Project*. [Online]. Available: https://www.hyperledger.org/

[33] M. Hearn and R. G. Brown, "Corda: A distributed ledger," R3 CEV, New York, NY, USA, Corda Tech. White Paper, 2016.

[34] J. Trevathan, "Security, anonymity and trust in electronic auctions," *XRDS, Crossroads, ACM Mag. Students*, vol. 11, no. 3, pp. 1–2, 2005.

[35] D. Wang, J. Zhao, and C. Mu, "Research on blockchain-based E-bidding system," *Appl. Sci.*, vol. 11, no. 9, p. 4011, Apr. 2021, doi: 10.3390/app11094011.

[36] M. Alharby and A. van Moorsel, "Blockchain-based smart contracts: A systematic mapping study," 2017, *arXiv:1710.06372*.

[37] D. Macrinici, C. Cartofeanu, and S. Gao, "Smart contract applications within blockchain technology: A systematic mapping study," *Telematics Informat.*, vol. 35, no. 8, pp. 2337–2354, Dec. 2018, doi: 10.1016/j.tele.2018.10.004.

[38] J. Kim, "Blockchain technology and its applications: Case studies," *J. Syst. Manag. Sci.*, vol. 10, no. 1, pp. 83–93, 2020.

[39] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, "Everything you wanted to know about the blockchain: Its promise, components, processes, and problems," *IEEE Consum. Electron. Mag.*, vol. 7, no. 4, pp. 6–14, Jul. 2018.

[40] S. Tikhomirov, "Ethereum: State of knowledge and research perspectives," in *Foundations and Practice of Security*, A. Imine, J. M. Fernandez, J.-Y. Marion, L. Logrippo, and J. Garcia-Alfaro, Eds. Cham, Switzerland: Springer, 2018, pp. 206–221.

[41] C. Pop, M. Prata, M. Antal, T. Cioara, I. Anghel, and I. Salomie, "An Ethereum-based implementation of English, Dutch and first-price sealed-bid auctions," in *Proc. IEEE 16th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2020, pp. 491–497.

[42] S. Seven, G. Yao, A. Soran, A. Onen, and S. M. Muyeen, "Peer-to-peer energy trading in virtual power plant based on blockchain smart contracts," *IEEE Access*, vol. 8, pp. 175713–175726, 2020.

[43] P. Manimaran and R. Dhanalakshmi, "Blockchain-based smart contract for E-bidding system," in *Proc. 2nd Int. Conf. Intell. Commun. Comput. Techn. (ICCT)*, Sep. 2019, pp. 55–59.

[44] A. Meher. (2020). *Auction-Cordapp. GitHub Repository*. [Online]. Available: https://github.com/ashutoshmeher-r3/auction-cordapp

[45] P. S. Anjana, S. Kumari, S. Peri, S. Rathor, and A. Somani, "An efficient framework for optimistic concurrent execution of smart contracts," in *Proc. 27th Euromicro Int. Conf. Parallel, Distrib. Network-Based Process. (PDP)*, Feb. 2019, pp. 83–92.

[46] C. Zan and H.-C. Xu, "A global clock model for the consortium blockchains," in *Blockchain and Trustworthy Systems*, Z. Zheng, H.-N. Dai, M. Tang, and X. Chen, Eds. Singapore: Springer, 2020, pp. 71–80.

[47] N. M. Freris, S. R. Graham, and P. R. Kumar, "Fundamental limits on synchronizing clocks over networks," *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1352–1364, Jun. 2011.

[48] P. Szalachowski, "(Short paper) towards more reliable Bitcoin timestamps," in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, Jun. 2018, pp. 101–104.

[49] A. Biryukov, D. Khovratovich, and S. Tikhomirov, "Findel: Secure derivative contracts for Ethereum," in *Proc. Financial Cryptography Workshops*, 2017, pp. 453–467. [Online]. Available: https://api.semanticscholar.org/CorpusID:27986785

[50] A. C. Jiménez, "Development of a distributed application over R3 corda," Ph.D. dissertation, UPC, Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, Dept. d'Enginyeria Telemàtica, Jun. 2021. [Online]. Available: http://hdl.handle.net/2117/357130

[51] A. A. Monrat, O. Schelén, and K. Andersson, "Performance evaluation of permissioned blockchain platforms," in *Proc. IEEE Asia–Pacific Conf. Comput. Sci. Data Eng. (CSDE)*, Dec. 2020, pp. 1–8.

[52] Z. Lai, "Deterministic concurrency control for private blockchains," Chin. Univ. Hong Kong, Tech. Rep., 2023.

[53] M. Fontoura, M. Ionescu, and N. Minsky, "Decentralized peer-to-peer auctions," *Electron. Commerce Res.*, vol. 5, no. 1, pp. 7–24, Jan. 2005, doi: 10.1023/B:ELEC.0000045971.43390.c0.

[54] R. Mutnuru. (Sep. 2014). *Weighted Rendezvous Hashing*. [Online]. Available: https://patents.google.com/patent/US9571570B1/en

[55] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 149–160, Oct. 2001, doi: 10.1145/964723.383071.

[56] K. Iwanicki, M. van Steen, and S. Voulgaris, "Gossip-based clock synchronization for large decentralized systems," in *Self-Managed Networks, Systems, and Services*, A. Keller and J.-P. Martin-Flatin, Eds. Berlin, Germany: Springer, 2006, pp. 28–42.

[57] T. S. Chandrashekar, Y. Narahari, C. H. Rosa, D. M. Kulkarni, J. D. Tew, and P. Dayama, "Auction-based mechanisms for electronic procurement," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 3, pp. 297–321, Jul. 2007, doi: 10.1109/TASE.2006.885126.

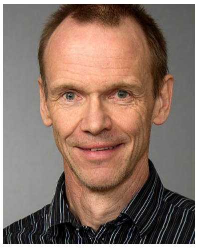
[58] Auctria. (2023). *Fundraising & Auction Features of Auctria*. [Online]. Available: https://www.auctria.com/features/

[59] Clarity. (2023). *Auction Ecommerce*. [Online]. Available: https://www.clarity-ventures.com/auction-ecommerce-platform-overview

[60] L. Foschini, A. Gavagna, G. Martuscelli, and R. Montanari, "Hyperledger fabric blockchain: Chaincode performance analysis," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

ERIC CHIQUITO received the B.S. degree in nanotechnology engineering from the Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO), Mexico, in 2019. He is currently pursuing the PhD degree in cyber-physical systems with Luleå University of Technology, Sweden. Prior to this, he was a Data Scientist with Grupo Posadas, a largest hotel group in Mexico. His research interests include decentralized systems and contractual automation for auctioning systems, data sharing, and the circular economy.

ULF BODIN received the Ph.D. degree in computer networking from Luleå University of Technology. He is a Professor with Luleå University of Technology, where he is conducting research on the Industrial IoT, distributed system of systems, computer communications, distributed ledgers, and applied machine learning. He has more than 15 years of experience in academia and the software industry, including standardization in ETSI and other organizations.



OLOV SCHELÉN (Member, IEEE) received the Ph.D. degree in computer networking from Luleå University of Technology. He is a Professor with Luleå University of Technology and the CEO with Xarepo AB. He has more than 25 years of experience in industry and academia. His research interests include mobile and distributed systems, the Industrial IoT and CPS, software orchestration, computer networking, artificial intelligence, and blockchain.

AHMED AFIF MONRAT (Member, IEEE) received the M.Sc. degree in computer science and technology from Erasmus Mundus Master Scholarship Program [Pervasive Computing and COMmunications for Sustainable Development (PERCCOM)], in 2018, and the Ph.D. degree in tradeoff analysis and sharding of blockchain technology from Luleå University of Technology (LTU), Skelleftea, Sweden, in 2023. He is a Postdoctoral Researcher with LTU. His research area is focused on performance and scalability of blockchain technology. His research interests include expert systems, artificial intelligence, machine learning, blockchain, digital twin, and cloud computing.