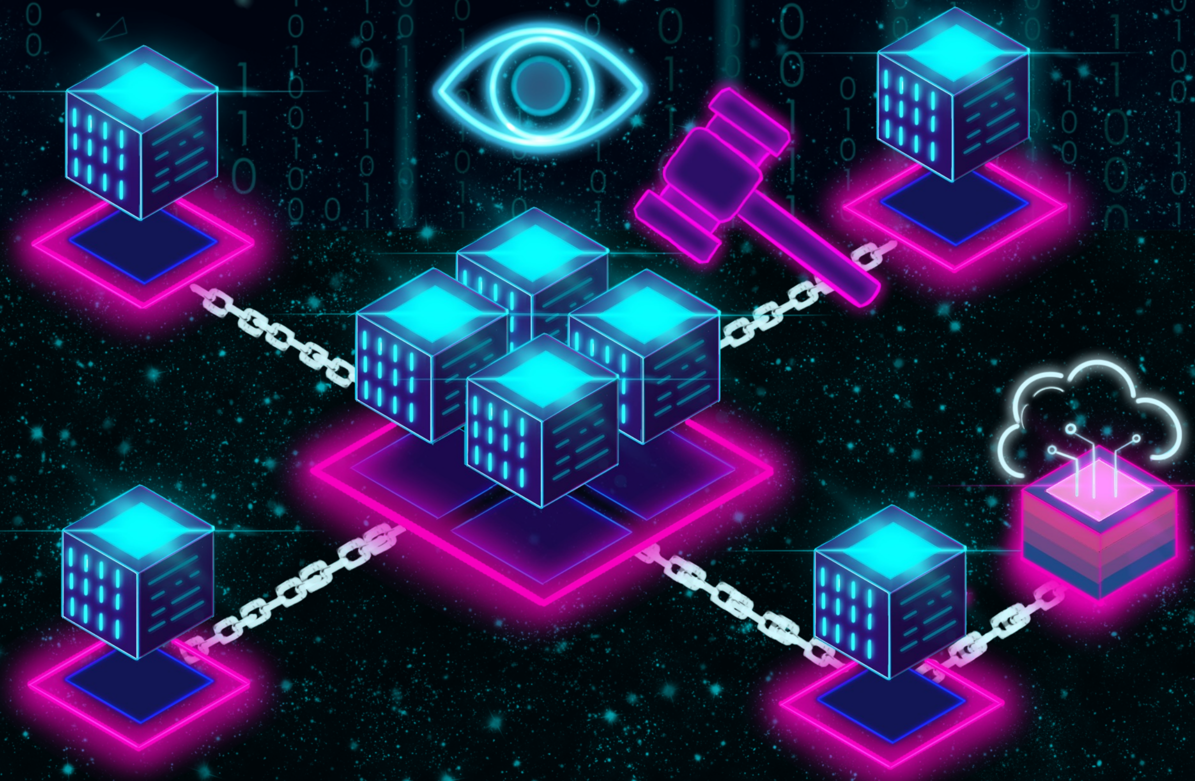
Enhancing Service-Level Agreements Using Decentralized Auctions and Witnesses

Zeshun Shi

Enhancing Service-Level Agreements Using Decentralized Auctions and Witnesses

Zeshun Shi



Enhancing Service-Level Agreements Using Decentralized Auctions and Witnesses

Zeshun Shi

The author was financially supported by Chinese Scholarship Council.

This research was also funded by the European Union's Horizon 2020 research and innovation program under grant agreement No. 825134 (ARTICONF), No. 862409 (BlueCloud), and No. 824068 (ENVRI-FAIR).

The author would like to thank ExoGENI for providing the testbed to conduct the experiments.

Cover design by Qianru Zuo.
Printed and bound by Ipskamp Printing.

ISBN: 978–94–6421–916–6

Enhancing Service-Level Agreements Using Decentralized Auctions and Witnesses

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op donderdag 3 november 2022, te 10.00 uur

door Zeshun Shi
geboren te GANSU

Promotiecommissie

Promotores:	prof. dr. ir. C.T.A.M. de Laat	Universiteit van Amsterdam
	dr. P. Grosso	Universiteit van Amsterdam
Copromotores:	dr. Z. Zhao	Universiteit van Amsterdam
Overige leden:	prof. dr. R.A. Prodan	University of Klagenfurt
	prof. dr. A.D. Pimentel	Universiteit van Amsterdam
	prof. dr. ing. L.H.M. Gommans	Universiteit van Amsterdam
	prof. dr. S. Klous	Universiteit van Amsterdam
	dr. A.M. Oprescu	Universiteit van Amsterdam
	dr. Z.A. Mann	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Contents

Chapter 1

Introduction

> The blockchain symbolizes a shift in power from the centers to the edges of the networks.
>
> William Mougayar

The cloud computing paradigm provides flexible services based on pay-as-you-go business models [102]. In the current cloud marketplace, several well-known service providers maintain the cloud marketplace, and the share of these top providers is continuously growing. According to a report, as of October 2020, AWS, Azure, Google, and Alibaba control 63% of the entire cloud marketplace, whereas all other providers only share 37%.[1] Since product migration is complex, consumers become locked in a particular provider's ecosystem. In addition, providers on the market are diverse in terms of service quality, price, and reputation, often making it difficult and time-consuming to select suitable providers that meet specific application requirements.

An auction is one of the effective and fair solutions to this problem. It is a sale activity in which potential buyers make competitive bids for assets or services [117]. In the field of cloud computing, the auction-based pricing strategy can effectively reflect potential trends in cloud resource demand and supply. Thus it is an effective way to allocate resources and satisfy both buyers and sellers [221]. Auction-based cloud pricing strategies have been developed rapidly in the past few years. For instance, large cloud service providers (e.g., AWS, Azure, and Google) have supported spot instance pricing for users to bid for unused capacity in a cloud data center. Some users can even save up to 90% of the cost compared with the traditional on-demand instance pricing.[2] However, it is still challenging to apply auctions in the current centralized cloud transaction model, which is mainly because:

[1] https://www.canalys.com/newsroom/worldwide-cloud-market-q320

[2] https://aws.amazon.com/ec2/spot/

- **There is a lack of a trustworthy platform** for users to auction cloud services from multiple providers. Most existing cloud auction solutions have the vendor lock-in issue; the provider also acts as an auctioneer, which may lead to bias and untrustworthiness.
- **There is a lack of an automated and cost-effective mechanism** to enforce the service lifecycle from auction agreement generation to service delivery; traditional auction houses or auctioneers are cumbersome and expensive.
- **There is a lack of a fair mechanism to detect the auction agreement violation** (e.g., the cloud service is not delivered as agreed) without bias. The provider has more power in the current model to verify service violations and decide whether to compensate the customer.

There are typically two cloud transaction models: centralized and decentralized [194]. In a centralized cloud environment, all service trading and trust-related issues rely on trusted third parties (TTPs), e.g., some well-known cloud service providers with good reputations and track records. However, those providers are not always trustworthy in practice and can be biased or conspire with any party. On the other hand, in a decentralized trading environment, all sellers or buyers perform transaction management and operations, avoiding the concentration of power and making the transactions more trustworthy. In this case, all trust assurance comes from a decentralized platform (e.g., blockchain), which needs to be appropriately designed, implemented, deployed, and monitored.

Traditionally, a Service-Level Agreement (SLA) is a business concept that defines the contractual financial agreements between the roles engaging in the business activity. In the context of a cloud marketplace, it is an agreement between the cloud customer and provider regarding the cloud service quality [160]. For instance, the IaaS (Infrastructure-as-a-Service) provider, Amazon Elastic Compute Cloud (Amazon EC2), claims that the availability of its data center is no less than 99%. If this is not achieved, it will pay back 30% credits to its customers as compensation. In practice, however, this agreement is hard to enforce fairly and transparently; it is usually performed manually and dominated by giant providers in the traditional SLA management process.

In recent years, blockchain has attracted tremendous attention as an enabling technology for building decentralized systems. In general, blockchain is a decentralized ledger system that combines existing technologies such as distributed data storage, peer-to-peer (P2P) networking, consensus mechanisms, and cryptographic algorithms. The ledger is maintained by all nodes participating in the system and is therefore decentralized, tamper-proof, transparent, and secure [16]. Blockchain was originally introduced as the underlying technology for Bitcoin [150]. Now, with smart contract technology bringing powerful programmability, it is widely believed that blockchain can be applied to build decentralized systems in various application scenarios, e.g., healthcare, finance, energy trading, wireless communication, service allocation, electronic voting, and supply chain management [114, 27, 130, 108].

Blockchain technology can be used to support decentralized applications (DApps), bringing new hints of possible solutions to address the challenges in service auction and SLA management [207, 210]. Due to its immutable and verifiable properties, blockchain has proven to be a promising tool for auction usage without requiring a TTP [24, 110]. It inspires the emergence of a new decentralized cloud marketplace that encourages greater inclusivity and participation from different service parties. In 2018, for the first time in the world, million-dollar artworks from Andy Warhol had been tokenized and auctioned on the blockchain.[3] This mechanism of bidding on item ownership with cryptocurrencies and smart contracts has shown its great potential. We can foresee that such a decentralized model will provide more choices and opportunities for both cloud providers and consumers.

The smart contract makes it possible to manage and automate the SLA process on the blockchain in a fair and tamper-proof way [174]. However, reaching a consensus on events that occur outside the blockchain is another possible challenge. Cloud customers or providers can still violate the agreed SLA despite using the blockchain to complete cloud transactions. For example, the provider may not provide the QoS (Quality of Service) they promised, and the customer may refuse to pay for the claimed cloud resources. In the blockchain community, the bridge between on-chain and off-chain events is called "oracle" [148]. One of the solutions to build this bridge is to retrieve data from Oraclize[4], a third-party company that performs as a trusted data source for the blockchain. However, this solution suffers from a single point of failure and needs extra commission fees. In this case, a decentralized witness mechanism is promising to judge SLA violations that occur off-chain.

This thesis aims to enhance the traditional cloud marketplace and SLA management lifecycle by introducing a novel Auction and Witness Enhanced trustworthy SLA for Open, decentralized service MarkEtplaces (AWESOME) framework. Specifically, a new role called auction witness is involved in the entire cloud service trading process. In our model, cloud service providers/customers can conduct decentralized P2P auctions. Decentralized blockchain users can join the SLA judgment and work as witnesses through an incentive mechanism that motivates them to make truthful judgments to win profits. An illustration of the proposed solution is shown in Figure 1.1.

[3] https://finance.yahoo.com/news/andy-warhol-multi-million-dollar-162928721.html

[4] http://www.oraclize.it/

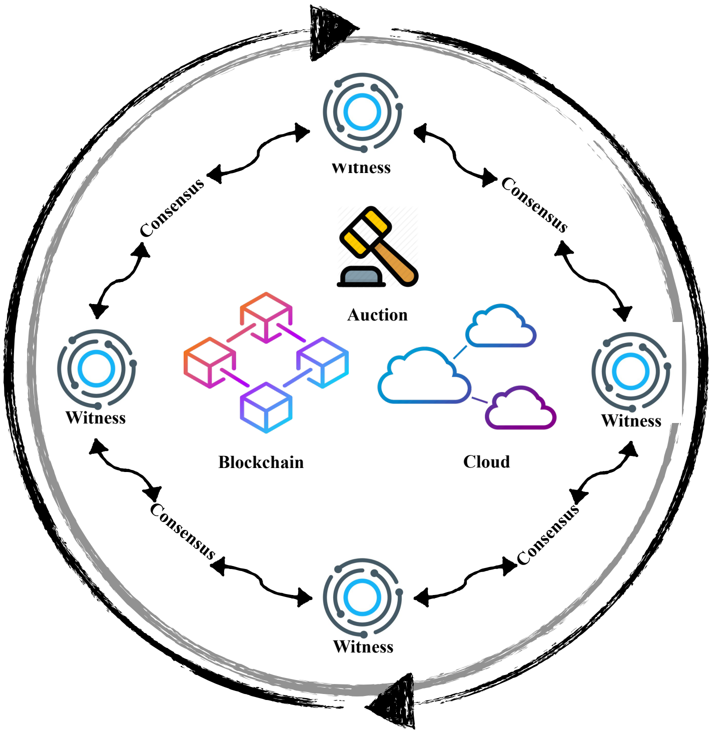


Figure 1.1: An illustration of the proposed decentralized cloud marketplace.

1.1 Research Questions

We thus identify our key research question as:

RQ: How to enhance the efficiency and trustworthiness of the cloud SLAs using decentralized auctions and witnesses?

To answer this main research question, we further define the following sub-questions:

RQ1: What are the state-of-the-art technologies and open challenges for building a decentralized service auction framework?

The investigation of theoretical knowledge is crucial to the framework proposed in this thesis. However, to the best of our knowledge, there is no general survey on the current landscape of blockchain-based auction models. Research gaps still remain on how blockchain technology can be leveraged to optimize auction models, especially in a cloud service marketplace.

RQ2: How to automate the decentralized service auction and quality monitoring process in an SLA model?

Traditional service auctions and SLA management processes are typically performed manually and led by giant providers, which is cumbersome and unreliable. The current cloud marketplace requires a trustworthy mechanism to automate the process of service auction and SLA enforcement.

RQ3: How to improve the efficiency of service auctions for managing federated clouds?

The effectiveness of the auction is crucial to the prosperity of the model. An ideal auction model should incentivize bidders to join the auction and produce the optimal buyer/seller combination. Despite the variety of auction models on the market, designing auctions to select cost-effective providers to construct federated cloud services remains a challenge.

RQ4: How to enhance the trustworthiness of federated SLAs in a decentralized service environment?

The trustworthiness of federated SLAs monitoring determines the reliability of the model. The model will not be able to attract witnesses to join and produce the desired output without the help of a suitable incentive model. Therefore, it remains a challenge to design such an effective incentive mechanism for decentralized witnesses to ensure the consistency and trustworthiness of federated SLAs monitoring.

RQ5: How to operate blockchain services to meet the scalability requirements of the AWESOME framework?

Traditional permissionless blockchains suffer from limited scalability, which significantly limits the wider adoption of the AWESOME framework. On the other hand, permissioned blockchains usually offer better scalability and performance. However, with a wide range of permissioned blockchain platforms in the market, it remains a challenge to choose the appropriate blockchain platform to support a scalable AWESOME framework.

1.2 Key Contributions

This paper contributes literature review, models, algorithms, and prototypes for SLA management in a blockchain-based decentralized cloud marketplace. Specifically, the main contributions of this thesis are the following:

A State-Of-The-Art Literature Review On Blockchain-Based Auction Models

We provide a comprehensive research landscape on the blockchain-based decentralized auction models. Unlike the existing survey efforts that focus on only one specific application field (e.g., energy trading [200, 158, 83]), our research covers the main auction application fields currently covered in existing literature, with several taxonomies generated. The main contributions of this part can be summarized as follows:

- Provide a conceptual schema to analyze research and innovation opportunities by reviewing existing blockchain technologies and auction models.
- Provide a taxonomy to classify applications and solutions by investigating existing research on blockchain-based auction models.
- Guide the design of applications that require blockchain for auction models by identifying open research challenges from the reviewed models.

AWESOME: An Auction and Witness Enhanced SLA Management Framework for Decentralized Cloud Marketplaces

We aim to enhance the cloud marketplace and SLA management lifecycle by introducing a novel AWESOME framework. Specifically, a new role called auction witness is involved in the entire cloud service trading process. In our framework, decentralized cloud service providers/customers can perform P2P auction transactions. Decentralized blockchain users can join the SLA judgment and work as witnesses through an incentive mechanism that motivates them to make truthful judgments to win profits. In brief, the main contributions of this part can be summarized as follows:

- A novel auction and witness enhanced SLA framework called AWESOME for decentralized cloud marketplaces. The model can support interactions between service providers, customers, and witnesses to complete trustworthy transactions and SLA enforcement.
- A prototype DApp based on the AWESOME framework is fully developed on the Ethereum blockchain.[5] It contains customizable graphical user interfaces (GUIs) and advanced smart contract protocols to support the SLA business process.
- Extensive experiments are designed to evaluate the execution latency and cost of the proposed model and DApp. The experimental results demonstrate that our model is economical and feasible to implement.

Towards an Incentivized AWESOME Framework: A Bayesian Game Approach for Federated Cloud Services

[5] Code repository: `https://github.com/ZeshunShi/AWESOME`

We propose an incentivized AWESOME framework using Bayesian game theory and blockchain for federated cloud services. Specifically, we first model the partition of federated cloud services as a graph partition problem. Then, Bayesian games are leveraged to model incomplete information sharing among different participants, and to enhance the effectiveness and trustworthiness of the AWESOME framework. Finally, a new algorithm is proposed to deal with privacy challenges. Our enhanced AWESOME framework considers both the effective bidding and the trustworthy enforcement of the SLAs. In brief, the main contributions of this part are summarized as follows:

- An off-chain federated cloud partition model is proposed to help cloud customers determine the number of cloud providers needed and prepare for the auction.
- Two unique Bayesian Nash Equilibriums (BNEs) are derived to select cost-effective providers, and to monitor federated SLAs in a consistent and trustworthy way.
- A timed message submission (TMS) algorithm is designed to protect the privacy during the message submission phase.
- We validate the equilibrium results of two BNEs and implements the proposed model and algorithm on the Ethereum blockchain.[6] The analytical and experimental results demonstrate the feasibility, trustworthiness, and cost-effectiveness of our model.

Towards a Scalable AWESOME Framework: A Permissioned Blockchain Approach and Empirical Study

We conduct extensive performance studies of permissioned blockchains to provide insights for a more scalable AWESOME solution. Firstly, an empirical study on five permissioned blockchain platforms is performed. The study demonstrates that the AWESOME approach is feasible and provides insights into which blockchain to choose when constructing such an AWESOME ecosystem. Then, a case study of operating permissioned blockchain in a dynamic cloud environment is presented. In summary, the main contributions of this part are as follows:

- A comparative analysis of five different blockchain platforms to demonstrate their performance in terms of scalability, stability, and resource consumption.
- An empirical study of Hyperledger Sawtooth to demonstrate the performance of operating permissioned blockchains in a dynamic cloud environment.
- Provides insightful suggestions on the selection of permissioned blockchains when building such an AWESOME ecosystem.

[6] Code repository: `https://github.com/ZeshunShi/SC4CloudAuction`

1.3 Thesis Overview

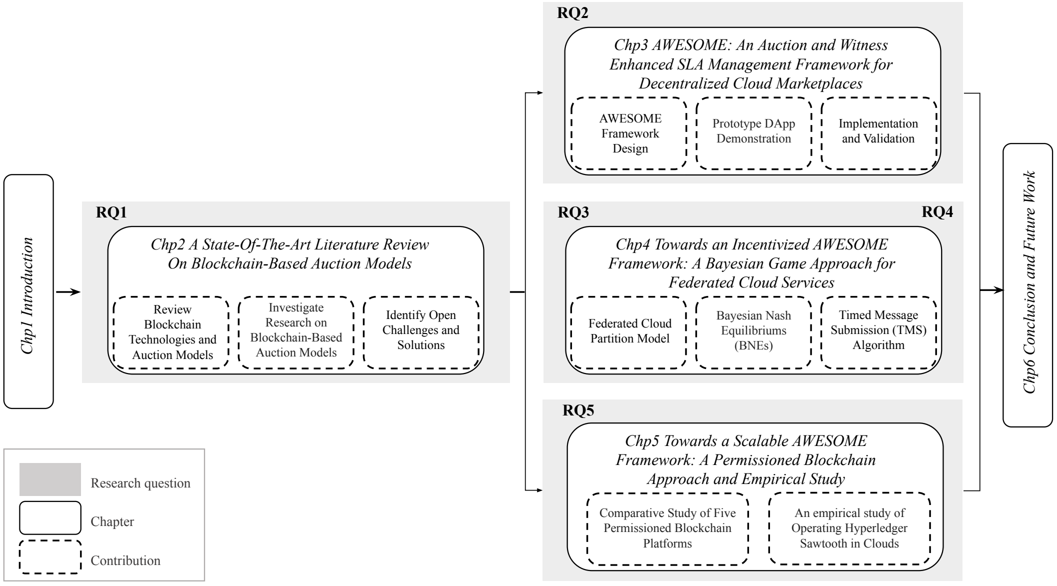

Chp1 Introduction
RQ1
Chp2 A State-Of-The-Art Literature Review On Blockchain-Based Auction Models
Review Blockchain Technologies and Auction Models
Investigate Research on Blockchain-Based Auction Models
Identify Open Challenges and Solutions
RQ2
Chp3 AWESOME: An Auction and Witness Enhanced SLA Management Framework for Decentralized Cloud Marketplaces
AWESOME Framework Design
Prototype DApp Demonstration
Implementation and Validation
RQ3
RQ4
Chp4 Towards an Incentivized AWESOME Framework: A Bayesian Game Approach for Federated Cloud Services
Federated Cloud Partition Model
Bayesian Nash Equilibriums (BNEs)
Timed Message Submission (TMS) Algorithm
RQ5
Chp5 Towards a Scalable AWESOME Framework: A Permissioned Blockchain Approach and Empirical Study
Comparative Study of Five Permissioned Blockchain Platforms
An empirical study of Operating Hyperledger Sawtooth in Clouds
Chp6 Conclusion and Future Work
Research question
Chapter
Contribution

Figure 1.2: The overview of the thesis (including chapters, research questions, and contributions).

Figure 1.2 shows the overview of this thesis, including the relationship between chapters, research questions, and key contributions. The thesis consists of six chapters in total. Chapter 1 introduces the background of the study, the research questions, and the structure of the thesis. Chapter 2 begins with an extensive literature review of the technologies, applications, and open challenges for blockchain-based decentralized auctions and marketplaces, which is aimed explicitly at **RQ1**. This chapter lays the theoretical foundation for the blockchain-based cloud marketplace proposed in this thesis. The following three chapters form an detailed overview of the entire AWESOME framework, including the design and implementation of the AWESOME framework in Chapter 3, a Bayesian game-enhanced AWESOME framework in Chapter 4, and a scalable AWESOME apporach based on permissioned blockchains in Chapter 5. In response to **RQ2**, Chapter 3 focuses on the design of the AWESOME framework and the implementation of the DApp. A prototype system based on the Ethereum permissionless blockchain is also fully implemented and validated. For **RQ3** and **RQ4**, Chapter 4 proposes an incentivized AWESOME framework to improve the effectiveness of service auctions and the trustworthiness of SLA management. Specifically, the partition of federated cloud services is first modeled as a graph partition problem to help customers choose the appropriate number of service providers. Then, two BNEs are derived respectively to motivate bidders and witnesses, and to achieve the desired system goal. Finally, an algorithm is proposed to deal with the privacy challenges on the blockchain. To answer **RQ5**, Chapter 5 proposes a permissioned blockchain approach to enhance the scalability of the AWESOME framework. In order to select the appropriate blockchain infrastructure, this chapter provides a comparative analysis of five popular permissioned blockchain platforms in the community. In addition, an empirical study of operating a permissioned blockchain in clouds is presented to provide insights on deploying the AWESOME framework in a dynamic cloud environment. Finally, Chapter 6 concludes the full thesis and provides an overview of future work.

1.4 Sources of the Chapters

A complete list of 16 publications is presented at the end of the thesis on Page 161. Here, we provide a quick overview of the material on which each chapter is based and the contributions of the authors.

- Chapter 2 is based on the following paper:

  - **Zeshun Shi**, Cees de Laat, Paola Grosso, and Zhiming Zhao. "Integration of Blockchain and Auction Models: A Survey, Some Applications, and Challenges". *IEEE Communications Surveys & Tutorials.* (To appear)

  ZS conceived the original idea and wrote the manuscript. CdL, PG, and ZZ supervised the whole project.

- Chapter 3 is based on the following paper:

  - **Zeshun Shi**, Veno Ivankovic, Siamak Farshidi, Jayachander Surbiryala, Huan Zhou, and Zhiming Zhao. "AWESOME: An Auction and Witness Enhanced SLA Model for Decentralized Cloud Marketplaces". *Journal of Cloud Computing* (2022): 11(1), pp.1-25.
  - **Zeshun Shi**, Siamak Farshidi, Huan Zhou, and Zhiming Zhao. "An Auction and Witness Enhanced Trustworthy SLA Model for Decentralized Cloud Marketplaces". *In ACM International Conference on Information Technology for Social Good (GoodIT)*, pp. 109-114. ACM, 2021.

  In the first publication, ZS designed the model and performed the experiments. ZS wrote the manuscript with the help of VI, JS, and SF. HZ and ZZ supervised the whole project. In the second publication, ZS conceived the original idea and wrote the manuscript with the help of SF. HZ and ZZ supervised the whole project.

- Chapter 4 is based on the following paper:

  - **Zeshun Shi**, Huan Zhou, Cees de Laat, and Zhiming Zhao. "A Bayesian Game-Enhanced Auction Model for Federated Cloud Services Using Blockchain". *Future Generation Computer Systems* (2022): 136, pp.49-66.

ZS designed the framework and carried out the experiments. ZS wrote the manuscript with the help of HZ. CdL and ZZ supervised the whole project.

- Chapter 5 is based on the following paper:
  - **Zeshun Shi**, Huan Zhou, Yang Hu, Jayachander Surbiryala, Cees de Laat, and Zhiming Zhao. "Operating permissioned blockchain in clouds: A performance study of hyperledger sawtooth". *In 2019 18th IEEE International Symposium on Parallel and Distributed Computing (ISPDC)*, pp. 50-57. IEEE, 2019.
  - **Zeshun Shi**, Huan Zhou, Jayachander Surbiryala, Yang Hu, Cees de Laat, and Zhiming Zhao. "An automated customization and performance profiling framework for permissioned blockchains in a virtualized environment". *In 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), workshop on resource brokering with blockchain (RBChain)*, pp. 404-410. IEEE, 2019.
  - Huan Zhou, **Zeshun Shi**, Ouyang Xue, and Zhiming Zhao. "Building a blockchain-based decentralized ecosystem for cloud and edge computing: an ALLSTAR approach and empirical study". *Peer-to-Peer Networking and Applications* (2021): 14(6), pp.3578-3594. (as co-first author)

In the first publication, ZS designed the study and performed the experiments. ZS wrote the manuscript with the help of HZ, YH, and JS. CdL and ZZ supervised the whole project. In the second publication, ZS conceived the original idea and wrote the manuscript with the help of HZ, YH, and JS. CdL and ZZ supervised the whole project. In the third publication, HZ and OX developed the theoretical framework. ZS carried out the experiment and wrote half of the manuscript. ZZ supervised the whole project.

Chapter 2

A State-Of-The-Art Literature Review On Blockchain-Based Auction Models

In recent years, blockchain has gained widespread attention as an emerging technology for decentralization, transparency, and immutability in advancing online activities over public networks. As an essential market process, auctions have been well studied and applied in many business fields due to their efficiency and contributions to fair trade. The practices of using blockchain to enable decentralized auctions trigger a great potential for research and innovation. This is because the decentralized nature of blockchain can provide a trustworthy, secure, and cost-effective mechanism to manage the auction process. This opportunity has attracted enormous research and innovation activities in both academia and industry; however, there is a lack of an in-depth review of existing solutions and achievements. In this chapter, we conduct a comprehensive state-of-the-art survey of this research topic. We review existing solutions for blockchain-based auction models and generate application-oriented taxonomies. Additionally, we highlight several open research challenges and future directions.

This chapter is based on:

- **Zeshun Shi**, Cees de Laat, Paola Grosso, and Zhiming Zhao. "Integration of Blockchain and Auction Models: A Survey, Some Applications, and Challenges". *IEEE Communications Surveys & Tutorials*. (To appear)

2.1 Introduction

Over the past decade, we have witnessed the success of blockchain as a novel technology for building decentralized systems. In general, blockchain is a decentralized ledger technology that incorporates cryptography, peer-to-peer (P2P) networks, and consensus mechanisms. The records in a ledger are maintained by all nodes participating in the system and are decentralized, tamper-proof, transparent, and secure [16]. In 2008, Satoshi Nakamoto first introduced blockchain as the foundation technology for a cryptocurrency named Bitcoin [150]. Besides distributed ledgers, the concept of smart contracts brings programmability to the blockchain, and allows developers to build applications, e.g., transportation and logistics, agriculture and food, energy and utilities, healthcare, and life sciences [39] on blockchains in a decentralized paradigm. According to MarketsandMarkets [139], the worldwide blockchain market is predicted to expand to $39.7 billion and cover specific applications across more than 15 industries.

An auction is a process of buying and selling goods or services. This process involves offering items for bidding, waiting for bids to be accepted, and then selling goods to the highest bidder under the supervision of an auctioneer [123]. Typically, auctions tend to be centrally organized and offline. Due to their fairness properties, auctions are widely used in trading activities for artworks, cars, radio spectra, online advertisements [146]. In the field of economics, auction theory has become one of the most successful and active branches [116]. Hundreds of auction models have been designed to serve different auction scenarios. A case in point is the spectrum auction that the Federal Communications Commission (FCC) has been conducting since 1994 [147]. Since then, spectrum auctions have contributed more than $200 billion of revenue to the U.S. government. The two designers of the FCC auction were awarded the Nobel Prize in 2020 for their improvements to auction theory and the invention of new auction formats [154].

Potential research and innovation opportunities across both blockchain and auction models have emerged recently [152]. Traditional centralized auctions usually require a third-party auctioneer or auction house to manage the entire auction process, which is expensive due to high commission fees. They also suffer from a single point of failure, and auctioneers can potentially be malicious in some cases [206]. In this context, blockchain has emerged as a decentralized platform to support trustworthy online auction applications. In 2018, for the first time in the world, multi-million dollar artworks by Andy Warhol were tokenized and auctioned successfully using the Ethereum blockchain [205, 52]. It is also reported that major auction houses (e.g., Sotheby's and Christie's) are actively working on applying blockchain in secure and trusted auction use cases [151]. Thus, we can foresee that this mechanism of bidding for ownership of items with blockchain could become the future trend.

The opportunities of applying blockchain in auctions have attracted many research and innovation activities; however, there is a lack of surveys to systemically review those different technical developments and achievements, and to identify the important open challenges. The remainder of this chapter is organized as follows. Firstly, the preliminary knowledge of auction models and blockchain technologies is presented in Section 2.2. Then, Section 2.3 introduces the motivations and considerations for the integration. Section 2.4 highlights and summarizes the current research challenges and solutions. Finally, the chapter is concluded in

Section 2.5. Appendix A provides a detailed review of blockchain-based auction applications, including a summary of auction models and blockchain technologies used in studies of different application domains.

2.2 Background Knowledge

In this section, we begin with a brief overview of different auction models and blockchain technologies. We then proceed to discuss the opportunities and considerations behind their combinations.

2.2.1 Auction Models

An auction is a sale activity in which potential buyers make competitive bids for objects or services [117]. There are usually several fundamental elements in an auction: 1) a seller who owns and wants to sell the objects; 2) one or several bidders who want to buy the objects via the auction; 3) the auction objects traded between the seller and the buyer(s); and 4) an auctioneer who works as an intermediary agent to host and control the auction process.

Auction models can be categorized along different dimensions, e.g., the bidding process, the number of items, the roles of buyers/sellers, and the bidding participants [76]. In the rest of this section, we review auction models that are frequently used in the blockchain-related literature. A comparison of those auction models is also shown in Table 2.1.

Open-Outcry Auction vs. Sealed-Bid Auction

From the perspective of the bidding process, an auction model can be either open-outcry or sealed-bid. In an open-outcry auction, a bidder's bidding activities are transparent and visible to all bidders. Whereas in a sealed-bid auction, 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 [49]:

- *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 only one bidder is left and no higher bids are obtained within the specified time span. The whole process of requesting bids is open and transparent. It can be very competitive, with pressure rising as bidders' offers increase. Since the auctioneer would try to get the best price for the seller, an English auction is expected to benefit the seller. An English auction can be profitable for sellers, but they often pose problems for bidders. In addition, it requires iterative communications and adjustments, which can sometimes be a bit difficult and costly.

- *Dutch Auction* (also called open-outcry descending-price auction or clock auction). In a Dutch auction, the auctioneer starts by announcing a high asking bid and then keeps lowering this bid until a buyer is willing to accept it. This auction is often used to sell goods that must be sold quickly (e.g., fresh products). For example, such auctions are very common in the Dutch flower sales market. In some cases, Dutch auctions may result in inappropriate bidding, which may be caused by a lack of sufficient information among bidders.
- *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 his/her 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, but they can collect information about their competitors' bids before submitting their own. Since bidders could not see the bids of other participants, they could not adjust their bids accordingly. In addition, bidders are vulnerable to the winner's curse.
- *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. In Vickrey auctions, truthful bidding is the dominant strategy [116]. One concern with this type of auction is that it has been well studied in theory but not very popular in practice.

Single-Item Auction vs. Multi-Item Auction

From the perspective of the number of items, an auction model can be single-item or multi-item. The above-mentioned four auction models are the main types of auctions where a single item is sold [49]. However, in some situations, selling multiple items at the same time is a more efficient way. Multi-item auctions can be further subdivided into two cases: an auction is said to be homogeneous if all items offered in the auction are identical; otherwise, it is considered heterogeneous.

- *Combinatorial Auction* (also called multi-lot auction) [43]. This is a popular auction in which heterogeneous items are sold at the same time. Bidders can place bids on combinations (or “packages") of items. It is suitable to auction scenarios where bidders have non-additive valuations for bundled items. Despite allowing more expression for bidders, combinatorial auctions present computational and mechanism design challenges compared to traditional auctions. For example, the winner determination problem is often a computationally intensive NP-hard problem.
- *Multi-Unit Auction* [123]. This is an auction in which several homogeneous items are sold. Based on the different payments for each unit, it can be

further divided into two types, i.e., pay-as-bid auction (or discriminatory price auction) and uniform price auction (or clearing price auction) [204]. In the former, bidders pay their bids for each unit they won. Whereas in the latter, all winning bidders pay the same price regardless of their actual bid. It should be noted that the incentive of the multi-unit auction may cause bidders to bid less than their true value, resulting in inefficient allocations.

Forward Auction vs. Reverse Auction

An auction model can be either forward or reverse in terms of the roles of buyers/sellers. A forward auction is also called a seller-determined auction, in which one seller sells products to multiple potential buyers (bidders). The auction models discussed so far are all forward ones. In a reverse auction, however, the roles of buyers and sellers are swapped: sellers need to bid and compete for the opportunity to sell their products.

- *Reverse Auction* (also called buyer-determined auction or procurement auction) [106]. In a reverse auction, one buyer needs to trade with multiple potential sellers. The buyer first makes a request for the required goods or services. Then sellers place bids for the goods or services they are willing to deliver. A reverse auction is highly suitable for procurement activities proposed by governments, companies, and organizations since it motivates sellers' competition. One of the main disadvantages of a reverse auction is that it does not require bidders to provide information about the specific costs involved in the contract. This can lead a buyer to choose a seller who appears to bid the lowest price but offers inferior products or poor customer service.

Single-Sided Auction vs. Double Auction

In terms of the participants in the bidding process, an auction model can be single-sided or double-sided. The single-sided approach has been widely implemented in traditional auctions (e.g., forward and reverse auctions). However, in some cases, they cannot accommodate additional sellers/bidders in a large-scale situation. The double auction is an extension of the conventional auction, which adopts the many-to-many strategy to generate multiple winning bidders in each round [128].

- *Double Auction* (also called double-sided auction) [63]. In this auction, multiple sellers and buyers submit their bids/offers, respectively. The market institution (auctioneer) then chooses a price that clears the market. Many different market clearing mechanisms already exist, including average mechanism, VCG (Vickrey-Clarke-Groves) mechanism, trade reduction mechanism, and McAfee's mechanism [14]. In reality, a double auction is suitable for marketplaces with multiple sellers and buyers, e.g., stock exchanges. The double auction mechanism is challenging to handle the auction of heterogeneous items with multiple attributes due to the substantial execution time and cost required.

Others

Some other emerging auction models found in the literature are listed as follows:

- *All-Pay Auction* [17]. Every bidder must pay regardless of whether he/she wins or not. The auction is awarded to the highest bidder as in a conventional auction. It is popular among governments and central banks. However, overbidding is a common behavior in the auction process and can result in winner's curse.
- *Multi-Attribute Auction* [21]. The bids could have multiple attributes (e.g., service time and quality) other than price. In this case, a scoring mechanism is needed to calculate the total bidding value. It is suitable when the auction needs to consider multiple attributes (e.g., service allocation). One challenge in multi-attribute auctions is designing a reasonable scoring mechanism to determine which bid is the best. Unfortunately, this cannot be addressed by simply comparing different attributes.
- *Sponsored Search Auction* (also called keyword auction) [105]. It is specially designed for search advertising scenarios. In this auction, n advertisers (bidders) compete for the assignment of k advertisement slots/positions. Each bidder submits a bid, then the highest bidder gets the first slot (with his/her bid), the second-highest bidder gets the second slot, and so forth. Based on the winner's different payment strategies, it can be further divided into generalized first-price (GFP) auction, generalized second-price (GSP) auction, and VCG auction. There are some trade-offs among them: GFP auctions are easy to use but less stable; GSP auctions incorporate the advantages of Vickrey auctions but do not support truthful bids; VCG is a truthful auction and is relatively stable, but users may find it difficult to understand and use in reality.

2.2.2 Blockchain Technologies

Introduced by Satoshi Nakamoto in 2008, blockchain was initially used as the underlying technology for Bitcoin. It records transactions among distributed participants as identical copies through a decentralized ledger, which is represented as a chain of blocks. Based on the consensus among distributed participants, new blocks are generated and attached to the chain using a cryptographic algorithm. In this process, a blockchain builds trust among its distributed users by virtue of the immutability and security of the ledger.

Table 2.1: Summary of Representative Auction Types

Auction Type	Alternative Name	Auction Mechanism	Properties/Suitable Scenarios
English auction	Open-outcry ascending-price auction	• The price starts low and increases as buyers bid. • The auction continues until no higher bids are received.	• Support a dynamic price discovery process and maximize sellers' profits.
Dutch auction	Clock auction; Open-outcry descending-price auction	• The auctioneer starts the auction with a high asking price. • The price is gradually reduced until one bidder accepts it.	• Suitable for perishable auction items or auctions that need to be completed quickly.
FPSB auction	Blind auction	• All bidders simultaneously submit a sealed bid. • The highest bidder wins and pays his or her bid.	• Prior to making their own offers, bidders can collect details about their competitors' bids.
Vickrey auction	Second-price sealed-bid auction	• All bidders simultaneously submit a sealed bid. • The highest bidder still wins but only pays the second-highest bid.	• Well studied in theory due to the truthful bidding property, but uncommon in practice.
Double auction	Double-sided auction	• Multiple sellers and buyers submit their bids/offers. • The auctioneer chooses a price that clears the market.	• Real word marketplaces with multiple sellers and buyers, e.g., stock exchanges.
Combinatorial auction	Multi-lot auction	• Several heterogeneous items are sold. • Bidders can place bids on combinations of items.	• Suitable when bidders have non-additive valuations on bundles of items, e.g, spectrum allocation.
Uniform price auction	Clearing price auction	• Multiple homogeneous items are sold. • Winners pay the same price regardless of their actual bid.	• Bidders tend to shade their bids when they demand multiple units.
Pay-as-bid auction	Discriminatory price auction	• Multiple homogeneous items are sold. • Winners pay their bids based on the items they won.	• A common way to allocate assets and commodities. • Bidders face no uncertainty about the price they will receive if they win.
All-pay auction	-	• Every bidder must pay regardless of whether they win. • The auction is still awarded to the highest bidder.	• Very popular among governments and central banks. • Overbidding is a common behavior.
Multi-attribute auction	-	• The bids may have multiple attributes. • A scoring mechanism calculates the attributes' value.	• Suitable when multiple attributes (e.g., service time, quality) need to be considered in an auction.
Reverse auction	Buyer-determined auction; Procurement auction	• The buyer makes a request for the required goods. • Sellers place bids for the goods they are willing to buy.	• Suitable for procurement by governments and companies, as it causes sellers' competition.
GFP auction	-	• n bidders compete for k slots/positions. • The highest bidder gets the first slot (with his bid), the second-highest gets the second, and so on.	• The auction structure is naturally unstable. • The first mechanism introduced in sponsored search auctions.
GSP auction	-	• n bidders compete for k slots/positions. • The highest bidder gets the first slot and pays the second highest bid, and so on.	• An extension of Vickrey auction for multiple units. • The most commonly used mechanism for sponsored search auctions.
VCG auction	-	• Bidders submit bids that report their true value. • Each bidder pays for the losses he or she causes to others. • Items are assigned in a socially optimal way.	• An extension of Vickrey auction for multiple units. • More complex to interpret and implement than the GSP auction in sponsored search auctions.

Blockchain Architecture

Blockchain researchers and practitioners often model blockchain systems using a layered architecture, and abstract typical blockchain technologies and functional components as six bottom-up layers: data, network, consensus, incentive, contract, and application layer [222]. The three layers at the bottom are usually considered a blockchain's basic elements, while the upper three layers are the extended elements.

- *Data Layer* defines the schema, data structure, and storage of all the data information on the blockchain. As the name suggests, a blockchain uses the "chained blocks" data structure as its backbone. Each block consists of several transactions, with useful information (e.g., version, hash, nonce, timestamp, and Merkle root) contained in the block header. The blocks are chained to each other via cryptographic algorithms (e.g., asymmetric encryption, digital signature, and hashing algorithm), making the data layer constitute a tamper-proof database for the blockchain. In this regard, Bitcoin uses double iterative SHA-256 as the hash function, while Ethereum uses KECCAK-256. ECDSA (Elliptic Curve Digital Signature Algorithm) is the transaction signature algorithm used by both Bitcoin and Ethereum.
- *Network Layer* models protocols for connecting blockchain nodes and validating data transferred across them. Blockchain nodes are typically connected using a P2P paradigm, where the network is maintained by all peer nodes together, and no single agent can control the whole system. Based on the type of underlying P2P network (e.g., whether it is structured or unstructured), different blockchain platforms may use different communication protocols. Bitcoin, for example, uses a gossip-based protocol to select peers and exchange states. When new transactions are generated on a node, they are first propagated to the neighboring nodes for validation. If the data structure and syntax are valid, they are saved for further processing; otherwise, they are simply rejected. Ethereum, on the other hand, relies on the Kademlia distributed hash table (DHT) protocol to manage communication in its P2P network. This is different from the unstructured P2P network used by Bitcoin [201].
- *Consensus Layer* is the foundation and core of a blockchain system. It defines protocols and algorithms for decentralized nodes to reach a consensus on the update of the blockchain. The most common and successful consensus algorithm is Proof of Work (PoW). Other 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, have also been widely discussed recently [202]. These algorithms will be discussed in detail in the next section.

- *Incentive Layer* provides incentive mechanisms for a blockchain to motivate participants to validate the data and maintain the whole system. Incentive mechanisms are typically based on block rewards and transaction fees. For example, the issuance mechanism of the Bitcoin blockchain guarantees that successful miners are rewarded with 6.25 Bitcoins when a new valid block is mined. At the same time, the transaction fees associated with each transaction can be allocated to the corresponding miners. This layer is essential in permissionless blockchains. Whereas in a permissioned blockchain, the incentive mechanism is often optional since the participants are selected organizations [119].
- *Contract Layer* defines decentralized programming paradigms in a blockchain, which was initially promoted by the Ethereum smart contract technology. A smart contract is a tamper-proof and self-executing program running on the blockchain, which enables a much broader range of application innovations in addition to cryptocurrencies. The concept of smart contracts has also extended to other blockchain platforms, e.g., chaincodes [96] and transaction processors [101] are smart contracts offered by Hyperledger Fabric and Sawtooth, respectively.
- *Application Layer* defines application programming interfaces (APIs) and programming models for developing specific applications. Blockchain was once well known for its cryptocurrency application (e.g., Bitcoin). Now with the popularity of smart contract technology, blockchain-based applications, namely decentralized applications (DApps), are showing huge market potential in many industrial sectors [139].

Consensus Algorithms

Consensus algorithms lie at the heart of blockchain technology. Considering the decentralized nodes involved in the blockchain network and the potential instability of communication, the design of consensus algorithms is full of challenges. Since the invention of the blockchain, new consensus mechanisms have been created continuously. Some of them are the improvements on PoW, while others are the traditional distributed fault-tolerant algorithms. This section summarizes commonly used consensus algorithms in popular blockchain platforms.

- *Proof of Work (PoW).* This is the most famous and successful blockchain consensus algorithm. In PoW, miners need to earn bookkeeping rights by demonstrating the amount of work they contribute. The process of proving workload is to solve a puzzle (also known as mining), and the miner who solves the puzzle faster has priority for bookkeeping [202]. The advantage of PoW is the high level of decentralization it can provide. PoW is also considered to be the most secure blockchain consensus mechanism to date. The disadvantage is that it can cause energy waste because mining requires a lot of computational resources. In addition, it limits the performance of the blockchain network. Bitcoin and Ethereum use PoW as the underlying consensus algorithm.

- *Proof of Stake (PoS).* In the PoS consensus, whoever has more stakes (i.e., tokens) gets the right to produce blocks. A fundamental assumption of POS is that the stake owners prefer to maintain the consistency and security of the blockchain system [54]. It has the prominent advantage of being more efficient than PoW. However, the security needs to be further validated due to the low level of decentralization. Examples of industry-leading PoS blockchains include Cardano and Avalanche. Ethereum, originally designed as a PoW blockchain, is also being upgraded to a PoS version called Ethereum 2.0.
- *Delegated Proof of Stake (DPoS).* This is a voting-based consensus algorithm; token holders vote for a certain number of representatives (based on the tokens held in their hands) to be responsible for producing new blocks and maintaining the network. As a variant of PoS, DPoS optimizes the traditional PoS using a voting mechanism. However, it can suffer from low enthusiasm for voting and concentration of power. Blockchain projects that use DPoS include EOS and Lisk.
- *Proof of Authority (PoA).* This consensus algorithm aims to unify the state of the blockchain by electing authoritative validators with good reputations [42]. There are many similarities between PoA and PoS, for example, they both do not require mining and therefore have good performance. The disadvantage of PoA is the low level of decentralization it caused. This consensus algorithm typically serves test networks and private blockchains. For example, Ethereum Kovan testnet and the private Ethereum version on Azure Blockchain Workbench are both based on the PoA protocol.
- *Practical Byzantine Fault Tolerance (PBFT).* The idea of Byzantine Fault Tolerance (BFT) was first proposed in the 1980s and there are many implementations of the algorithm. Among them, PBFT is the most famous one, which provides (n-1)/3 fault tolerance while guaranteeing system liveness and safety [42]. It has the advantage of dealing with the inefficiency of the original BFT algorithm and tolerating malicious peers, while the drawbacks are limited scalability and high latency. In the current blockchain community, Hyperledger Sawtooth supports a pluggable PBFT consensus protocol. Hyperledger Fabric claims to support PBFT but is not yet fully implemented.
- *Proof of Elapsed Time (PoET).* PoET is a type of consensus that uses a trusted execution environment (TEE) to improve the efficiency of the current PoW protocol. It uses the randomly generated elapsed time to determine the right for bookkeeping. PoET provides an excellent solution

to the random miner selection problem, but has the disadvantage of being necessarily dependent on dedicated hardware for security. PoET is primarily promoted and used in Hyperledger Sawtooth [95].

- *Raft.* Raft is a distributed consensus algorithm that functions similarly to Paxos [92]. Compared to Paxos, it is easier to understand and implement in real systems. In Raft, each node has three states: follower, candidate, and leader. The Leader is selected for bookkeeping in a continuous iterative voting process. Raft has the advantage of low algorithm complexity and easy implementation. However, it only supports crash fault tolerance and cannot solve the problem of malicious nodes. Raft is the consensus algorithm mainly used by Hyperledger Fabric and Oracle Blockchain.
- *Others.* In addition, researchers have identified dozens of new consensus algorithms. Other commonly used consensus algorithms include tangle-based solutions, which are widely used in directed acyclic graph (DAG) blockchains (e.g., IOTA). In addition, some platforms adopt customized consensus solutions. For example, the Corda blockchain achieves consensus by confirming the validity and uniqueness of transactions [166].

Blockchain Types

In general, there are three types of blockchain networks: permissionless, permissioned, and hybrid blockchain. This section provides a brief summary of them. A more detailed comparison is shown in Table 2.2.

- *Permissionless Blockchain.* In a permissionless or public blockchain (e.g., Bitcoin or Ethereum), anyone can join the network by submitting or validating transactions. To address the lack of trust among anonymous players, a consensus mechanism is often used to determine who gets the right to package transactions and produce new blocks in a given round. PoW is a good illustration of such a consensus algorithm and has been validated with the popularity of blockchain. However, it has been criticized for being inefficient and consuming too much energy in order to reach a consensus. It is widely believed that in a PoW-based permissionless blockchain, the waste of energy is inevitable in order to establish trust among strangers without any prior knowledge of each other.
- *Permissioned Blockchain.* A permissioned blockchain is operated as a closed ecosystem that can only be accessed by users with permissions. A user can only view the ledger or validate new transactions after being approved by the authority of the blockchain. In this way, malicious or crashed nodes can be identified through more energy-efficient consensus algorithms such as PBFT, PoET, and Raft. The ability of assigning specific network permissions to users and the enhanced performance give permissioned blockchains a great

Table 2.2: Characteristics of the Three Types of Blockchain

	Permissionless Blockchain	**Permissioned Blockchain**	**Hybrid Blockchain**
Participants	• Public • Anonymous	• Private/Consortium • Known identities	• Public + Private
Access Mechanism	• Anyone • Decentralized	• Selected users • Partially decentralized	• Customized
Consensus	• PoW, PoS • Energy-intensive	• PBFT, Raft, PoET • Energy-efficient	• Integrated
Performance	• Low	• High	• Medium
Examples	• Bitcoin • Ethereum	• Hyperledger Fabric	• Aergo

potential for wider industrial application. Hyperledger is one of the most successful blockchain communities and has incubated several permissioned blockchain platforms such as Fabric and Sawtooth [192]. However, there are also some arguments that the "partially decentralized" nature of the permissioned blockchain may lead to compromises in trust [202].

- *Hybrid Blockchain.* It aims to combine the strengths of both permissionless and permissioned blockchains and to customize the degree of decentralization based on specific application needs. A hybrid blockchain enables highly regulated organizations to have greater flexibility and control over which data is kept private versus shared on a public ledger [145]. A typical example is the Aergo platform, which consists of a public chain network using the DPoS consensus and several customized sidechains dedicated to specific applications based on leader-based PoA consensus mechanisms [3]. However, the usability and security of such a hybrid model still requires further validation.

2.3 Motivations and Considerations for the Integration

The integration between blockchain technologies and auction models can promote innovations in traditional auctions. Blockchain can be used to enable a decentralized auction system and improve the trustworthiness of centralized auctions. In this section, we discuss the research and innovation opportunities brought by the integration of blockchain and auction models.

2.3.1 Current Issues of Centralized Auctions.

Auctions can be centralized or decentralized, and they differ in how they achieve their auction goals. Centralized auction applications are operated and owned by a single company (e.g., eBay) and run on a centralized server cluster. The developer can retain full control over the auction application in a centralized auction platform. As a result, centralized auction applications can typically handle higher traffic volumes. More importantly, centralized auctions offer low-cost hosting, fast runtime, easy development, and a tightly controlled user experience. All of these factors have made centralized online auctions a huge success over the past few decades. However, these advantages also come at some serious costs. In general, centralized auctions have the following issues and challenges.

- *Centralized Auctions Are Inflexible.* One problem with centralized auctions is inflexibility. With the current centralized model of online auctions, each platform has several fixed auction formats, rules, policies, and user groups. As a result, both auction buyers and sellers are at risk of vendor lock-in, a situation where the cost of switching to a different vendor is so high that the customer is essentially stuck with the original vendor. Another reason for inflexibility is that platforms need time and money to develop new auction formats to match new technologies and user needs. The limitations of auction formats mean that dynamic user needs cannot be satisfied. As a result, auctions in the current marketplace are usually arranged and operated in an inefficient and inflexible manner. Besides, centralized auctions are subject to censorship from central authorities. The content of the auction is subject to the laws and regulations of the country in which it is held, as well as the platform's own rules and policies [6].
- *Centralized Auctions Are Opaque and Untrustworthy.* Centralized auctions operate in an opaque manner (i.e., a black box). Large auction companies and service providers are by default regarded as trusted parties that can potentially maintain, control, and manage user data, access, and activity. While this can be beneficial for users, it can potentially be used as a source of control to enforce surveillance or lead to abuse of trustworthiness [218]. For example, system administrators can obtain sensitive data in a private auction easily and help some bidders to win the auction.
- *Centralized Auctions Pose Security and Privacy Risks.* Centralized auction platforms collect user data collectively and store it in a certain number of servers to support the hosting of various types of services and applications [218]. Unfortunately, this exposes vulnerabilities and user data to cybercriminals, leading to serious security and privacy concerns. A prime example is eBay's report in 2014 that hackers had infiltrated their systems and stolen the passwords of 145 million users. In addition to account passwords, hackers obtained user private information such as names, email addresses, dates of

birth, physical addresses, and phone numbers [144]. These security incidents may cause negative influence and huge financial losses on auction users.

- *Centralized Auctions Suffer From Single Points of Failure.* Centralized auctions based on the client-server model are prone to single points of failure. These failures can cause the entire auction system to stop functioning due to network or system problems. If the centralized server goes down, the auction application will go offline, and users may not be able to use the application in a timely manner until the error is fixed. For auction use cases that require high availability and reliability (such as luxury jewelry and art auctions), a single point of failure is highly undesirable, which can cause severe property damage to users.
- *Centralized Auctions May Trigger Huge Expenses.* Centralized auctions tend to have higher commission costs. Online auction platforms such as eBay and eBid take a percentage of the final sale price from users to compensate for data processing and marketing costs. For example, eBay's auction commission fee is 12.9% of total sales. In contrast, eBid is cheaper but also requires a base fee of 5% of total sales. When the value of the auctioned item increases, the cost of a centralized auction will increase significantly. This will discourage the widespread adoption of auction applications by regular users [181].

2.3.2 Motivations for the Integration

Blockchain technologies effectively eliminate intermediaries, thereby reducing transaction costs and ensuring trust among auction stakeholders [90]. In general, blockchain technologies can enhance auction models from the following aspects:

- *Immutability of the Auction Transaction.* Every transaction executed on the blockchain is public, verifiable, and immutable. This means that the blockchain can be leveraged as an audit certificate device that prevents participants from cheating during the auction. The winning bidders can also use the blockchain as a transaction proof [24].
- *Automation of the Auction Process.* A smart contract automates the auction process on the blockchain. Almost all auction logic can be predefined in smart contracts to facilitate the exchange of goods or services as well as the token payment.
- *Decentralization of the Auction Management.* There is no need for a specific third-party auctioneer, which ensures trustworthiness and greatly reduces the auction cost. By contrast, traditional centralized auctions can be very expensive and subject to cheating auctioneers; auction houses typically charge 8-20% of the hammer price as a commission [153].

- *Flexibility in the Auction Payment.* Cryptocurrencies embedded in the blockchain can improve the security and flexibility of auction payments. At the same time, a decentralized payment scheme obviates the need for financial intermediaries, making transactions more convenient and less costly.

2.3.3 When Does Integration Make Sense?

While blockchain-based decentralized auctions are exciting and have the potential to change the way many auctions operate, it doesn't mean that blockchain is the right solution for all auction scenarios. In general, blockchain is preferred when buyers and sellers do not have a sufficient level of trust and are unwilling to use a third-party online auction platform [207]. In addition, auction organizers and participants need to consider the trade-off between the benefits and costs of centralized and decentralized auctions. For example, decentralized auctions are more difficult to maintain. Once smart contracts that support the auction logic are deployed on the blockchain, they can no longer be removed or manipulated. Therefore, if auction managers have a critical requirement for application updates or bug fixes, they should be careful about using a blockchain for decentralized auctions. In addition, performance in terms of latency and throughput is typically much better in centralized auction systems than in blockchains, due to the additional complexity introduced by the blockchain consensus mechanism. The trade-off between decentralization and throughput should also be considered when deciding whether to use a blockchain-based auction system.

2.4 Challenges and Future Directions

Despite the great potential of integrating blockchain with auction models, there are several research challenges that need to be addressed. In this section, we highlight and summarize eight open challenges identified in the literature, as shown in Figure 2.1.

2.4.1 Auction Enforcement

Blockchain and smart contracts cannot confirm the veracity of external data, which is known as the blockchain oracle problem. This is a big challenge that prevents the widespread adoption of smart contracts for auction applications on the blockchain. It should be noted that many of the (non-digital) auctioned items and services cannot be managed by the blockchain directly. For instance, in an art auction, while the ownership of artworks can be recorded by the blockchain, the blockchain cannot directly enforce the transfer of off-chain artworks. Basically, a blockchain oracle is a secure middleware that facilitates communication between the blockchain and any off-chain system [28]. Using oracles in an auction fills this

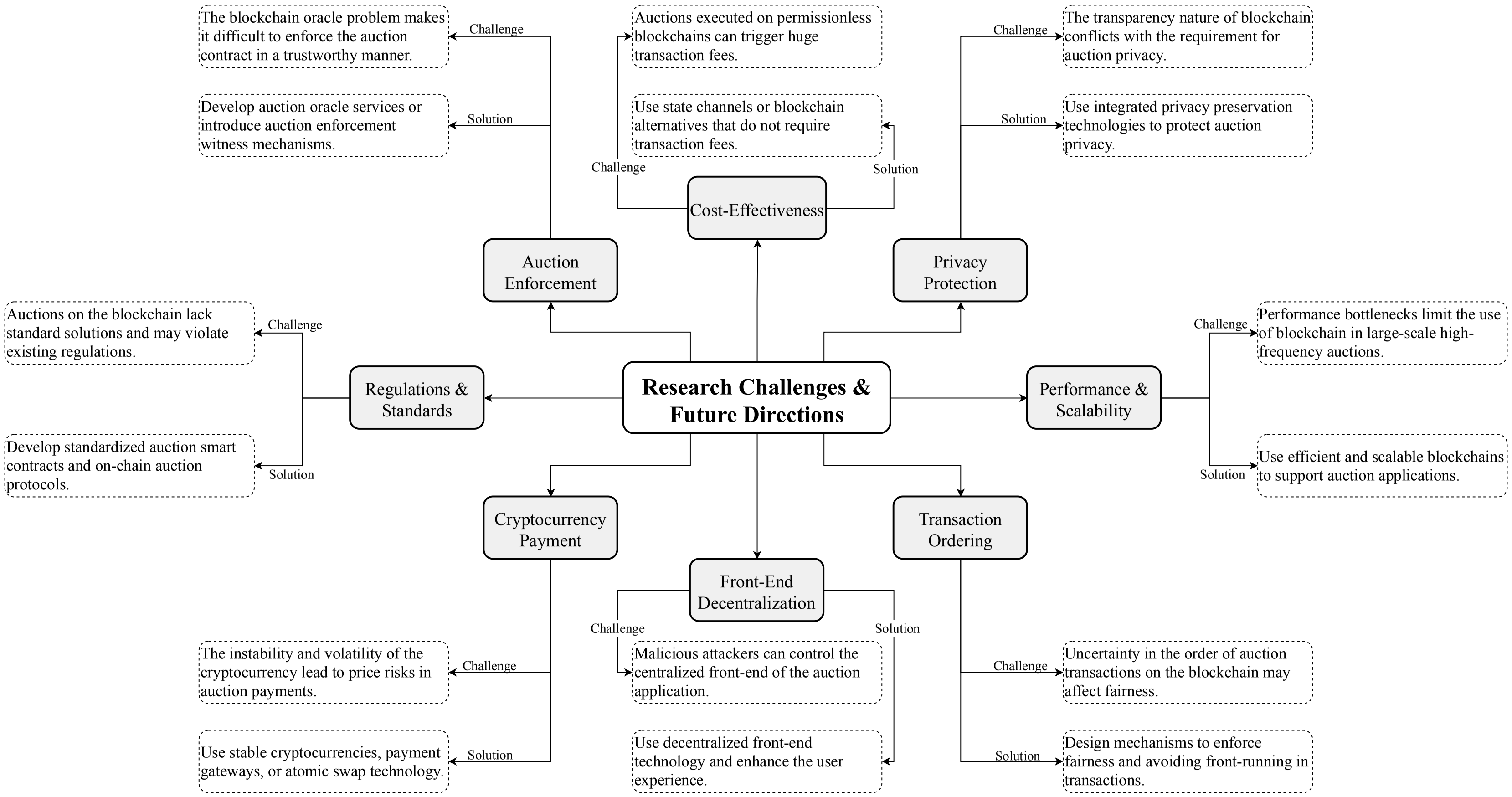

Research Challenges & Future Directions
Auction Enforcement
Challenge
The blockchain oracle problem makes it difficult to enforce the auction contract in a trustworthy manner.
Solution
Develop auction oracle services or introduce auction enforcement witness mechanisms.
Cost-Effectiveness
Challenge
Auctions executed on permissionless blockchains can trigger huge transaction fees.
Solution
Use state channels or blockchain alternatives that do not require transaction fees.
Privacy Protection
Challenge
The transparency nature of blockchain conflicts with the requirement for auction privacy.
Solution
Use integrated privacy preservation technologies to protect auction privacy.
Regulations & Standards
Challenge
Auctions on the blockchain lack standard solutions and may violate existing regulations.
Solution
Develop standardized auction smart contracts and on-chain auction protocols.
Performance & Scalability
Challenge
Performance bottlenecks limit the use of blockchain in large-scale high-frequency auctions.
Solution
Use efficient and scalable blockchains to support auction applications.
Cryptocurrency Payment
Challenge
The instability and volatility of the cryptocurrency lead to price risks in auction payments.
Solution
Use stable cryptocurrencies, payment gateways, or atomic swap technology.
Front-End Decentralization
Challenge
Malicious attackers can control the centralized front-end of the auction application.
Solution
Use decentralized front-end technology and enhance the user experience.
Transaction Ordering
Challenge
Uncertainty in the order of auction transactions on the blockchain may affect fairness.
Solution
Design mechanisms to enforce fairness and avoiding front-running in transactions.

Figure 2.1: Taxonomy of challenges and solutions for blockchain-based auction models.

gap and ensures that the real-world data fed into the blockchain (e.g., whether the auction item/service is delivered as agreed) is accurate and the auction contract is triggered properly [157]. This is why some smart contract-based auction platforms have a built-in oracle component [124]. Current blockchain oracle services are often provided by third-party companies. Some successful solutions include Chainlink, Provable, and Witnet [57]. These oracle services usually require additional commission fees, and a single oracle may suffer from a single point of failure. In [157], a decentralized oracle network is integrated into an auction system. The oracles act as external timers to trigger the start/end of the auction in a trustworthy way.

Another possible solution to the oracle problem is to introduce a decentralized witness mechanism to monitor the delivery of auctioned goods/services. In this case, game theory can be used to design incentive mechanisms to motivate normal blockchain users to join the network and work as witnesses [231]. In addition, a self-enforcing contract witness mechanism is proposed in [143]. The basic idea is that the smart contract can be enforced through the mutual judgment of auction participants. We believe that an efficient and economical oracle design solution will significantly facilitate the enforcement of blockchain-based auction applications.

2.4.2 Cost-Effectiveness

The issue of the high cost is a big challenge for auctions based on permissionless blockchains that require transaction fees. In the current Ethereum blockchain network, any operation that invokes a smart contract requires an execution cost. This cost is usually measured by gas, which represents the unit of computational effort required to perform specific operations on Ethereum [56]. Under certain exchange rates, the gas cost of submitting an auction transaction may be higher than its value, which will discourage normal users from using Ethereum for auctions. In this context, researchers have designed and tested a number of auction smart contracts in the related literature to verify the feasibility and affordability. In [24], the gas consumption of the smart contracts for four basic auctions (i.e., English, Dutch, FPSB, and Vickrey auction) is tested. The result shows that deploying sealed-bid auctions consumes a little bit more gas than open-outcry auctions. Nevertheless, the deployment and implementation costs of the four auctions are kept at a low level. Although smart contracts for basic auctions are less costly, a complex auction smart contract may include multiple operations that incur unexpected large costs. The cost is highly dependent on the design pattern of the auction smart contract and the code optimization. For instance, in an auction scenario where the auction process requires 300 iterations to complete, a total of 12,191,380 gases are required, which could incur a huge transaction cost that equals several thousand dollars [152].

2.4.3 Privacy Protection

In principle, all data stored on the blockchain must be public to all blockchain nodes in order to ensure traceability, verifiability, and immutability. This conflicts with the privacy requirements of most auction applications, especially for those with important trade secrets. Normal users will be discouraged from using the blockchain for auctions if privacy can not be fully guaranteed. As illustrated in Figure 2.2, there are generally two types of privacy concerns for blockchain-based auctions [11]. The first one is identity privacy, which considers participants' privacy and prevents transactions from being associated with specific auction users and their blockchain addresses. The second one is transaction privacy, which covers the privacy of auction information about bids, auction contracts, payments, and other transaction details We find that most researchers target both types of privacy concerns in their models through a combination of various techniques. Based on the relevant literature, commonly used privacy protection solutions and their challenges are summarized in the following text.

Cryptographic Primitives: Cryptographic techniques can effectively protect privacy in blockchain-based auction models. The most common cryptographic primitives may includes multi-party computation (MPC), zero-knowledge proof (ZKP), commitment scheme, asymmetric encryption, homomorphic encryption, and digital signature. Since these cryptographic techniques differ in terms of effects

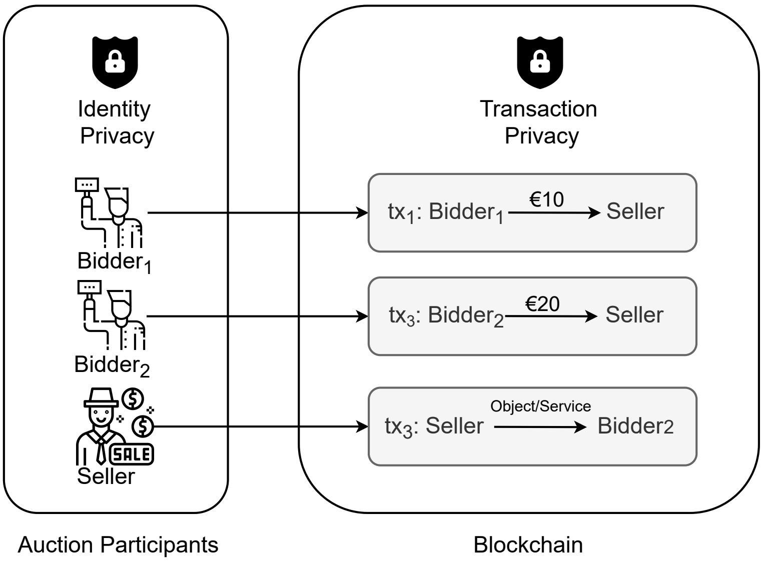

Identity Privacy
Transaction Privacy
Bidder1
Bidder2
Seller
tx1: Bidder1 €10 → Seller
tx3: Bidder2 €20 → Seller
tx3: Seller Object/Service → Bidder2
Auction Participants
Blockchain

Figure 2.2: There are two privacy concerns identified for blockchain-based auctions: identity privacy and transaction privacy. The former concerns the privacy of auction participants, while the latter concerns the privacy of various auction transactions (e.g., bids, payments, and contract details).

and application scenarios, some studies choose to integrate multiple algorithms on the blockchain to build a secure and privacy-preserving auction system. For example, ZKP, MPC, AE, and CS and their variant algorithms have been widely combined in recently proposed frameworks in [65, 1, 40, 171]. Such an integrated model can reduce the potential risk of using one single encryption algorithm, thus presenting an overall good privacy-preserving effect. On the other hand, while cryptographic primitives can protect auction privacy, they suffer from high computational complexity and high transaction costs when implemented on the blockchain. It is reported that a non-interactive ZKP verification roughly takes more than 3 million gas on the Ethereum blockchain [66]. The huge transaction fees make it impractical for auction users to use these algorithms and join the auction. To effectively protect privacy, the performance of the cryptographic algorithms used in the blockchain needs to be significantly improved [172]. Recent studies have focused on designing lightweight cryptographic protocols that are weaker than traditional ones (e.g., MPC/ZKP) [23, 40, 133, 137]. These protocols can perform specific auction tasks and achieve optimized on-chain performance.

Cryptographic protocols can also be implemented off-chain as separate modules to reduce on-chain execution costs. However, this increases the risk of data corruption during the transmission and communication phase. To solve this problem, Benhamouda *et al.* [18] proposed an approach that integrates secure MPC protocols into the Hyperledger Fabric blockchain architecture. This integration model combining blockchain and cryptographic protocols can significantly reduce the risk of data transmission and, therefore, may become a future research and development trend.

Permissioned Blockchain: Permissioned blockchains usually have an extra authentication mechanism for permission management and, therefore, can provide privacy protection against non-member users. For example, Hyperledger Fabric implements the “channel” technology, which is essentially a private ledger between specific network members; nodes within the same “channel” can share data, but nodes outside the channel cannot access it. In [47], a hybrid blockchain-based auction architecture is proposed, in which a permissioned blockchain is used to publish sensitive bids and a permissionless blockchain is used to make the auction accountable. It should be noted that this solution offers privacy protection against non-member peers, but still suffers from possible data leakage from malicious nodes within the same network [18]. Therefore, it is often used in combination with other encryption techniques.

2.4.4 Performance & Scalability

Performance is one of the major bottlenecks of blockchain in auction applications. As one of the world’s largest e-auction platforms, eBay needs to process more than 2 billion transactions per day [131]. However, current popular permissionless blockchain platforms (e.g., Bitcoin and Ethereum) can not support high-frequency auction transactions. Due to the requirement to establish trust among anonymous entities, a compute-intensive and time-intensive consensus mechanism (i.e., PoW) is often required. For instance, the throughput of Bitcoin and Ethereum are only around 7 and 15 transactions per second (tps), respectively [209]. According to the investigation in Appendix A, most existing blockchain-based auction models are implemented using Ethereum. As a result, their performance bottleneck would be around 15 tps in a public chain network. We argue that such a throughput value is sufficient for small-scale auctions (e.g., high-value auctions among a few participants) and static auctions requiring only a few rounds of bidding. However, dynamic auction models that include iterative bidding and intensive computations may need other high-performance blockchain alternatives.

Scalability is another big concern for blockchain-based auctions, especially when the number of auction participants increases dramatically. Nowadays, some large-scale e-auction platforms need to handle an increasing number of users and intensive auction transactions. For instance, eBid serves a daily traffic volume of around 60,000 visitors [170]. However, when conducting an auction on the blockchain, the maximum number of auction participants is based on the condition of the underlying blockchain, which is actually limited by the maximum size of each block [62]. Therefore, in order to support auction applications with large-scale users, more scalable blockchain solutions need to be carefully designed and developed. The use of the permissioned blockchain is a feasible and popular solution to this problem. Permissioned blockchains can generally achieve better performance and scalability due to the use of efficient consensus algorithms (such as PBFT or Raft) and customizable block parameters. Hyperledger Fabric is expected to reach a throughput of more than 3,000 tps. FastFabric [69] is a project that aims to extend Hyperledger Fabric with architectural improvements that allow it to support even 20,000 tps. However, it should be noted that the improved performance of the permissioned blockchain is at the expense of decentralization. The absence of a stable and widely recognized cryptocurrency also limits the widespread use of permissioned blockchains.

2.4.5 Transaction Ordering & Fairness

One of the significant challenges of decentralized auctions is the time synchronization. In a decentralized system, each node user has its own clock. There is no such an absolute clock, so it is difficult to enforce a precise time window to manage particular auction applications. Permissionless blockchains typically use PoW-based consensus algorithms to determine the order of transactions on the blockchain and avoid the double-spending problem. However, this approach introduces uncertainty in the order of auction transactions, i.e., auction users do not know whether their transactions will be prioritized or deferred. For example, different bidders may submit concurrent operations regarding a competing auction smart contract; one decides that the auction has ended, and the other is still trying to bid. Sometimes transactions may experience delays due to network congestion, and some transactions may even be canceled due to the process of the consensus mechanism. This uncertain waiting process can lead to the exposure of trading intentions, making front-running easy to occur in blockchain-based auctions. In traditional finance, front-running is a type of cheating, where information that will affect the price of an asset is known in advance from non-public information. In blockchain-based auctions, it means that while an auction transaction is waiting to be packaged, other users can profit by setting a higher blockchain fee to preempt the transaction [159]. Front-running is unfair and undermines the trading strategies of normal auction participants, harming their trading interests. In summary, the successful implementation of a blockchain-based decentralized auction application must deal with front-running issues to ensure the transaction's fairness. This is especially critical for auction models that are sensitive to the order of bids.

2.4.6 Front-End Decentralization

From a software development perspective, auction applications require excellent front-end components to assist users and improve the user experience. Decentralized smart contracts allow any compliant auction transaction to be executed securely and continuously as long as the blockchain exists. However, while smart contracts and the underlying blockchain are fully decentralized, the front-end of most on-chain auction applications is still implemented using traditional centralized Internet architecture. This allows attackers to influence the user experience by taking control of the front-end web page. An example is Whisky Auctioneer's claim that in 2020 that they had to shut down their auction website and stopped the online auction of thousands of bottles of rare whisky due to a constant Distributed Denial of Service (DDoS) attack [156]. Imagine if the auction application was truly decentralized, then attackers would not be able to prevent most users from accessing the front-end pages through DDoS attacks. An attacker could also do other malicious acts, such as making users connect to unaudited malicious contract code on the blockchain by controlling the front-end, even though the original auction smart contract is both audited and secure. To have a truly decentralized experience, users need to be able to control their front-end. This is important because it protects users from malicious attackers and achieves the goal of a fully decentralized auction application.

2.4.7 Cryptocurrency Payment

Following the end of an auction, the exchange of goods and money between buyers and sellers is expected to happen. Cryptocurrency is often leveraged to complete the auction payment due to its easy and secure transaction properties.

Besides, auction payments can be enforced automatically through the Ethereum token Ether in a smart contract. With such a design, payments can be processed within the blockchain, and transactions containing the corresponding values can be processed between different wallets [62]. On the other hand, the price volatility of cryptocurrencies is a big challenge. Due to the speculative nature of cryptocurrencies, their market values are constantly fluctuating. This makes it difficult for auction sellers to accept cryptocurrencies as the payment method without considering the price risk. Buyers who expect the cryptocurrency value to increase will also be hesitant to use their own tokens as auction payments [36]. This issue can be partially addressed by introducing a cryptocurrency payment gateway [142].

Market liquidity is another concern regarding using cryptocurrencies for auction payments. There are already cryptocurrencies that are designed to support application-specific auctions, e.g., GreenCoin [46] for energy trading and Xcoin [59] for spectrum trading. However, the trading market of these emerging cryptocurrencies is quite small and therefore lacks liquidity. This means that in some cases, cryptocurrencies may not be considered equivalent to fiat money. Another issue is that different blockchain platforms support different cryptocurrencies. In this case, an atomic swap [22] is an efficient solution for auction payments when users hold different cryptocurrencies.

2.4.8 Regulations & Standards

There is no authority in a decentralized blockchain network to avoid possible transaction disputes. In an auction application, decentralized users may generate transaction data in different formats. It would be a huge challenge to ensure that the information uploaded by auction users complies with the relevant laws and regulations. For instance, a key part of the EU General Data Protection Regulation (GDPR) lies in the citizen's right to data erasure, i.e., the GDPR claims that individuals have the right to delete the data associated with them [19]. However, due to the immutable nature of the blockchain, it is difficult to remove on-chain sensitive information once uploaded to the blockchain. Currently, different countries and regions are actively developing new blockchain industry regulations to promote blockchain applications. The compliance with current laws and regulations needs to be carefully considered when designing blockchain-based auction applications.

Another pressing challenge is standardization. Currently, different blockchain platforms have different architectures and design patterns, and there are hundreds of auction models to support different application scenarios. There is an urgent need for a standardized solution to set, maintain and merge standards across blockchain platforms to enable seamless integration. As one of the largest blockchain communities, Ethereum has developed several standards (e.g., ERC-20 for token development) to help maintain project interoperability across different implementations [55]. Standardized solutions for auction applications have great potential to address challenges such as interoperability, user experience, social acceptance, scale, governance, cost consumption, digital identity, privacy protection, and developer shortcomings [220]. We believe that the development and operations of standardized auction smart models will be an active research direction in the near future.

2.5 Conclusion

In this chapter, we reviewed existing auction models and blockchain technologies, and provided a conceptual schema to analyze research and innovation opportunities from their integration. Specifically, we provided an overview of main application areas for blockchain-based auction models, e.g., energy trading, wireless communication, and service allocation (this can be found in Appendix A). There are many open research challenges identified for blockchain-auction models. Here we list the most important challenges to be further investigated:

- Auction Enforcement: The blockchain oracle problem makes it difficult to enforce the auction contract in a trustworthy manner.
- Cost-Effectiveness: Auctions executed on permissionless blockchains can trigger huge transaction fees.
- Privacy Protection: The transparency nature of blockchain conflicts with the requirement for auction privacy.
- Performance & Scalability: Performance bottlenecks limit the use of blockchain in large-scale high-frequency auctions.
- Transaction Ordering & Fairness: Uncertainty in the order of auction transactions on the blockchain may affect fairness.
- Front-End Decentralization: Malicious attackers can control the centralized front-end of the auction application.
- Cryptocurrency Payment: The instability and volatility of the cryptocurrency lead to price risks in auction payments.
- Regulations & Standards: Auctions on the blockchain lack standard solutions and may violate existing regulations.

In this thesis, we concentrate our efforts on the first four challenges, i.e., auction enforcement, cost-effectiveness, privacy protection, and performance & scalability. We leave the study of the other four challenges to our future work.

In summary, recent research on the integration of blockchain and auction models is quite extensive. Scientific communities have recognized the great potential of integrating the two to solve problems in various application scenarios. While there are still many challenges, such an integration trend will be beneficial to both industry and academia. We believe that the presented survey will offer theoretical support and practical guidance for researchers and auction practitioners.

Chapter 3

AWESOME: An Auction and Witness Enhanced SLA Management Framework for Decentralized Cloud Marketplaces

In recent decades, the world has witnessed cloud computing as an essential technology that changes the traditional application Development and Operations (DevOps) lifecycle. However, current cloud software DevOps and Service-Level Agreement (SLA) management often face challenges of: 1) selecting the best fitting service providers, customizing services and planning capacities for large-scale distributed applications; 2) guaranteeing high-quality and trustworthy SLAs among multiple service providers; and 3) enhancing the interoperability of cloud services across different providers. This chapter proposes a novel framework called Auction and Witness Enhanced trustworthy SLA for Open, decentralized service MarkEtplaces (AWESOME) to build a trustworthy cloud marketplace and address the above challenges. The proposed framework contains four subsystems: a customizable graphical user interface, an auction-based service selection model, a witness committee management mechanism, and a smart contract factory orchestration. We prototype the AWESOME decentralized application (DApp) using the Ethereum blockchain. The experimental results demonstrate that our model and DApp are economical and feasible.

This chapter is based on:

- **Zeshun Shi**, Veno Ivankovic, Huan Zhou, Siamak Farshidi, Jayachander Surbiryala, and Zhiming Zhao. "AWESOME: An Auction and Witness Enhanced SLA Model for Decentralized Cloud Marketplaces". *Journal of Cloud Computing* (2022): 11(1), pp.1-25.
- **Zeshun Shi**, Siamak Farshidi, Huan Zhou, and Zhiming Zhao. "An Auction and Witness Enhanced Trustworthy SLA Model for Decentralized Cloud Marketplaces". *In ACM International Conference on Information*

Technology for Social Good (GoodIT), pp. 109-114. ACM, 2021.

In the rest of this chapter, we first introduce the related works in Section 3.1. Then, in Section 3.2 we present the details of system requirements, objectives, actors, on-chain and off-chain interactions, and the overall AWESOME system architecture. Next, we introduce the design choices of our AWESOME DApp and show the details of how the DApp works with a business process model in Section 3.3. Section 3.4 shows the experimental results to demonstrate the feasibility of the AWESOME framework and DApp. Finally, we conclude the whole chapter in Section 3.5.

3.1 Related Work

There are already several frameworks that target cloud services and resources allocation using blockchain and auction models [219, 44, 215, 34, 72, 186, 48, 67]. These models will be discussed in detail in Appendix A.2.3. It should be noted that most of these models focus on designing on-chain auction algorithms to improve allocation efficiency and privacy; unfortunately, none of them consider the execution of auction contracts. iExec [103] and Golem [68] are two popular projects that are closely related to our model. The former aims to build a blockchain-based distributed cloud environment, while the latter tries to build a worldwide supercomputer. They both propose to employ oracles to verify the success of a computing task and to trigger the execution of the smart contract if the conditions are met. However, most existing oracle services are provided by third-party companies (e.g., Chainlink, Provable, and Witnet) [57] and are therefore subject to single points of failure.

In summary, there is an urgent need to establish a secure, trustworthy, and cost-effective auction model in the federated cloud services scenario. Although blockchain-based decentralized auction models have great potential to tackle this problem, most existing solutions only focus on optimizing bidding processes regardless of the auction agreement enforcement. A comparison of our model with related studies is shown in Table 3.1. Our work is among the first to combine bidding and auction enforcement with blockchain.

Table 3.1: Comparison with existing blockchain-based cloud auction models.

Ref.	Topic	Auction Algorithm	Auction Privacy	Auction Enforcement
DeCloud [219]	Edge/Cloud service trading	●	●	○
CloudAgora [48]	Cloud storage and computing sharing	●	●	○
ChainFaaS [67]	Serverless computing	◐	○	○
AStERISK [186]	Shared economy service allocation	●	●	○
Chen et al. [34]	Cloud VM allocation	●	●	○
Gu et al. [72]	Cloud storage resource trading	●	○	○
Debe et al. [44]	Fog service trading	●	○	○
Yu et al. [215]	Edge service crowdsensing	●	○	○
iExec [103]	Decentralized cloud computing	◐	●	◐
Golem [68]	Worldwide supercomputer	◐	●	◐
Our model	Decentralized cloud service trading	●	●	●

Notes: Filled (or half-filled) circles indicate that the properties are (partially) addressed, while empty circles mean that properties are not considered.

3.2 The AWESOME Framework

In this section, we first analyze the requirements using two industrial use cases and discuss the design objectives of the system. Then, we describe our AWESOME framework in detail, including actor identification, on-chain vs. off-chain interactions, and system components & workflows.

3.2.1 Requirements Analysis

In industrial innovations (e.g., crowd journalism and disaster early warning) and scientific applications (e.g., research data management), cloud services are playing an increasingly important role in real-time data processing (e.g., multimedia acquired by mobile devices), running simulations (e.g., for predicting possible disasters), and for enabling extensive scale collaborations (e.g., for running distributed scientific workflows). Therefore, it is necessary to employ multiple data centers or providers to handle decentralized collaboration between resource providers and customers in several industrial use cases.

1. Use case 1. Decentralized cloud marketplace for social media (taken from EU ARTICONF project[1]): crowd journalism for real-time news reporting during live sports, music events, or natural disasters. Individual citizen journalists make photos or videos of the news and trade them via the news platform. The system has to detect fake news from those crowdsourced contents by running real-time processing at multiple cloud providers or engaging human experts to review them.
2. Use case 2. Decentralized service marketplace for medical data management (taken from EU CLARIFY project[2]): sharing and utilizing pathology data

[1] https://articonf.eu/

[2] http://www.clarify-project.eu/

provided by hospitals or individuals from different countries, where various medical data access constraints are often applied. When a machine learning application for studying breast cancer must use data from multiple hospitals, the application developer has to select cloud providers from a decentralized marketplace that meet the application needs (e.g., geolocation, capacity, and price).

We can therefore highlight the following requirements and challenges from those use cases:

- Provider selection, service customization, and capacity planning challenges. The developer has to select cloud services from different providers (very often multiple ones) due to distributed data locations (e.g., sensors or repositories), various data access constraints (e.g., for medical data), and performance constraints (e.g., for real-time decisions in early warning). The various price and reputation models make the selection time-consuming and challenging to be optimal.
- SLA interoperability and guarantee challenge. The time-critical application constraints, e.g., processing media contents during crowd news reporting and real-time decision-making, require the profound optimization of the application logic and the quality guarantee of the cloud services. However, the diverse SLA terms among providers and the uncertainties in the SLA guarantee make performance optimization difficult.
- Difficulties in setting up business processes in a decentralized marketplace. The business logic in a decentralized marketplace is often realized by smart contracts, which are supposed to be immutable after being deployed on blockchains. However, any careless design or mistake may cause unexpected loss.
- Virtual infrastructure automation challenge. When an application involves multiple providers or data centers, the provisioning of the virtual infrastructure, deployment of the software platform and application components, monitoring, and adaptation of the application need to be ideally automated. However, the diverse Application Programming Interfaces (APIs) from different providers and the interoperability issues across those providers make automated provisioning and deployment a challenge. This leads to a high level of complexity in monitoring runtime infrastructure quality, detecting SLA violations, and adapting the infrastructure when violations happen.

To tackle these challenges in a decentralized cloud marketplace, we propose to build the AWESOME framework. The AWESOME software architecture consists of novel technologies in DApp DevOps, game theory, blockchain, and smart contracts.

3.2.2 System Objectives

We aim to provide guidelines, tools, and templates that can aid developers in developing a DApp for a specific business that can benefit from a decentralized architecture. The system needs to provide service customers and providers a platform that can facilitate easy SLA generation and operation, and allow decentralized witnesses to monitor such SLAs. Furthermore, the system needs to be generic and modular. DApp developers who use the system could customize the model for their own business needs and adapt it to other blockchain use cases. Specifically, the objectives of the AWESOME framework can be summarized as follows:

- Objective 1: Improve the customer/provider selection in a decentralized ecosystem by developing an automated service auction framework to enable dynamic business relations between a consumer(s) and providers and establish SLAs.
- Objective 2: Improve the service quality and SLA's trustworthiness between consumer(s) and providers by establishing a decentralized witness mechanism to monitor the SLA violations and automate the procedure for SLA compensation and payment.
- Objective 3: Improve the model usability by developing easy-to-use customizable DApp GUIs for general cloud users to interact with different smart contracts.
- Objective 4: Improve the continuous DevOps of DApps by providing an integrated contract factory to improve smart contracts' security and efficiency.

3.2.3 Actor Identification

Actors which interact with the AWESOME framework are human roles, external systems, or devices that exchange information with the DApp. With this in mind, we identify the following actors:

- **Service Customer**: Service customers use the DApp to find providers that can offer them services. They should be able to make listings on the platform and sign an SLA with a service provider. They pay for these services but can receive compensation in case of SLA violations.
- **Service Provider**: Service providers use the DApp to list their available services on the platform for auction. They earn monetary rewards for these services but may be penalized in case of SLA violations.
- **Auction Witness**: Witnesses can use the DApp to monitor SLAs and receive monetary rewards for their efforts. The judgment from the witness committee can ensure that auctions and cloud services are delivered as agreed in SLAs.

- **AWESOME Operator**: An AWESOME operator could modify the developed framework for a specific business use case. An operator needs to read the provided documentation, edit the user interfaces, blockchain APIs, and smart contract templates, and then deploy a custom decentralized service marketplace.[3]

In our model, customers are motivated to receive services at a low cost, and providers are incentivized to provide high-quality services in return for service fees. These two parties complete transactions and make profits by means of customized on-chain auctions. The payoffs of both parties are always positive; otherwise, an auction will never be settled. The incentive for witnesses, on the other hand, can include the following three parts: 1) a reward for their monitoring efforts; 2) a penalty if their reports about SLA violations fail to match the results of others (based on the majority rule); and 3) a blockchain transaction fee. We specify in our model that the monitoring reward is large enough so that the witness's payoff is always positive. In this way, witnesses are always motivated to participate in our model and earn their rewards.

3.2.4 On-Chain vs. Off-Chain Interactions

After identifying system objectives and actors, we can now design the system architecture as a whole. It is important to formalize what transactions and data should be placed on-chain and off-chain when developing a blockchain-based system [138]. In practice, the information that should be included in the blockchain is that with critical trust requirements. This is because on-chain information is immutable and enforces non-repudiation. In addition, not only "big data" is not suitable for the blockchain, but even "not-tiny" data should not be stored on the blockchain. This is mainly due to cost and scalability considerations; the computing power and data storage space available on the blockchain is limited. One of the typical solutions is to store raw data off-chain due to its size while storing small critical data and hashes on-chain. In terms of computation, blockchains are not the best for complex, intensive computations; however, they provide a benefit in their interoperability properties as many systems can access it [212]. The authors in [138] suggested that as a general guideline, data with transparent and immutable requirements for DApps should be managed on the blockchain system. In this section, we follow this idea to demonstrate our design choices.

On-Chain Activities

The data and activities the system should be kept on-chain include:

- Auction: To support transparent trade, the auction of service tasks should be conducted on the blockchain to maintain fairness and prevent fraud.

[3] For simplicity, customer, provider, witness, and operator are used in the following text.

Implementing decentralized auctions on the blockchain can also avoid the cheating auctioneers of traditional centralized auctions and save commission fees.

- Witness Reports: In order to avoid tampering with the SLA reports, they should be placed on-chain. While this may lead to witnesses viewing each other's reports and reporting accordingly, the reports should be transparent. One effective way to address this issue and protect privacy is to submit the hashes of the reports in a specified time window.
- SLAs: It is essential to include SLA details between the service provider, customer, and witnesses in the blockchain, as all these data have trust requirements. However, since SLAs may be larger textual files that could give a large load to the blockchain, a possible solution is to place SLA metadata that can unlock the SLA off-chain while keeping the cryptographic hash on-chain.
- Payment Enforcement: In case of an SLA violation, a smart contract should be used to facilitate payments to providers, customers, and witnesses automatically. Blockchain cryptocurrencies can support the secure and fair enforcement of money payments.

Off-Chain Activities

The data and activities the system should keep off-chain include:

- User Interfaces: Due to data loads, the way in which providers, customers, and witnesses interact with the platform should mostly be off-chain.
- Cloud Services. Cloud services offered and used by providers and customers should be off-chain.
- Pre-Monitoring Communications: The platform should facilitate communications between providers and customers before entering an SLA agreement so that they can privately agree upon service and monitoring terms.

3.2.5 Overall System Components

Based on the above analysis, we designed the system architecture of AWESOME. The AWESOME framework consists of four subsystems in response to the proposed objectives, as shown in Figure 3.1:

1. **DApp Graphical User Interface (DGUI)** provides a flexible and customizable DApp interactive environment for different AWESOME users to connect on-chain and off-chain activities. It is designed to provide a bridge between customers, providers, and witnesses who do not have IT

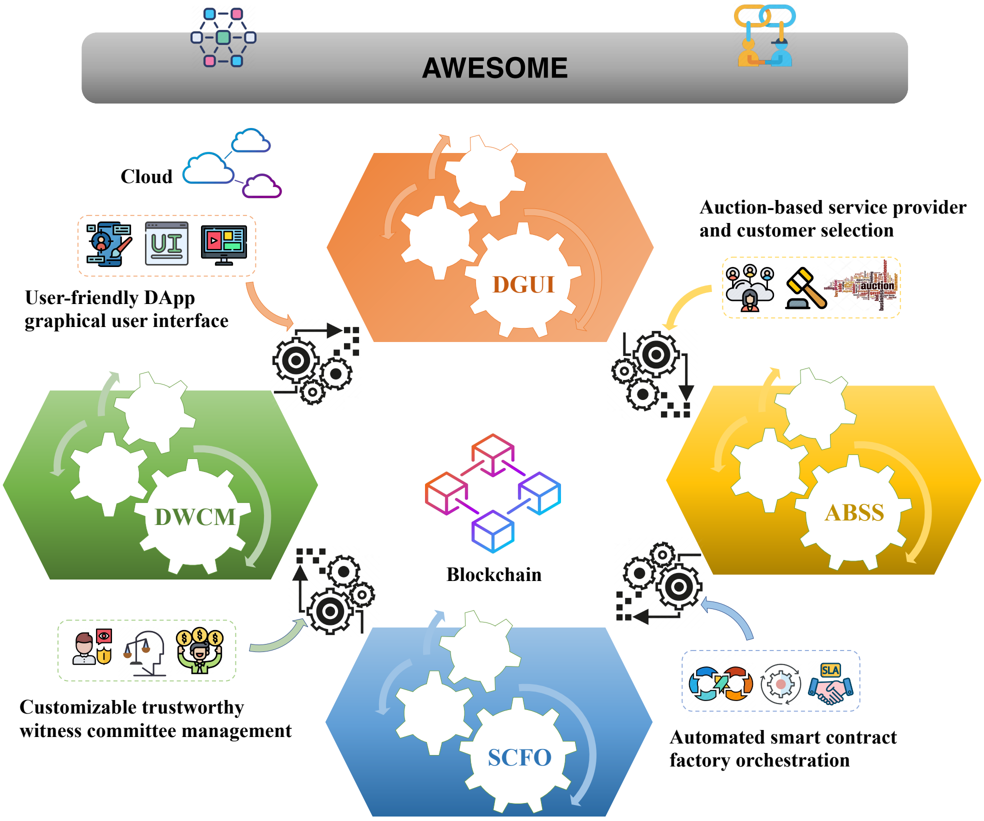

AWESOME
Cloud
User-friendly DApp graphical user interface
DGUI
Auction-based service provider and customer selection
auction
DWCM
Blockchain
ABSS
Customizable trustworthy witness committee management
SCFO
SLA
Automated smart contract factory orchestration

Figure 3.1: The overall architecture of the AWESOME framework.

development knowledge and assist them in calling function interfaces between different smart contract agents. Furthermore, the usability of the AWESOME DApp is ensured through a customizable interface design for business needs. More specifically, *DGUI* is designed with interfaces regarding 1) auctions by customers and providers; 2) SLA monitoring activities by witnesses; and 3) DApp management and maintenance by operators.

2. **Auction-Based Service Selection (ABSS)** provides an auction-based service customer/provider selection solution. This subsystem will first diagnose the use case requirements and help the user select the most suitable auction model and algorithm to achieve desirable results. Then, the management of the auction process and the enforcement of the service fee payment (in the form of cryptocurrency) are executed on the blockchain, ensuring that the whole auction is open and trustworthy. Finally, *ABSS* also audits bidder candidates to prevent malicious actors from joining the auction.

3. **Decentralized Witness Committee Management (DWCM)** provides a trustworthy incentive framework to manage decentralized auction witnesses. First of all, an appropriate number of witness candidates will be selected in an unbiased way to perform off-chain monitoring of cloud SLAs. *DWCM*

will then invoke a game theory incentive mechanism based on customized payoff functions to enable selected witnesses to make correct judgments to win more profits. The subsystem will also audit the witnesses' reputations to reward/restrict their participation in future monitoring activities.

4. **Smart Contract Factory Orchestration (SCFO)** provides tools and APIs for AWESOME operators to set the necessary smart contracts on the blockchain. More specifically, the subsystem is responsible for automating the process of planning, provisioning blockchain infrastructure, and deploying AWESOME business smart contracts. In addition, the *SCFO* subsystem also monitors and diagnoses smart contracts and the underlying blockchain infrastructure at runtime to provide effective adaptation solutions.

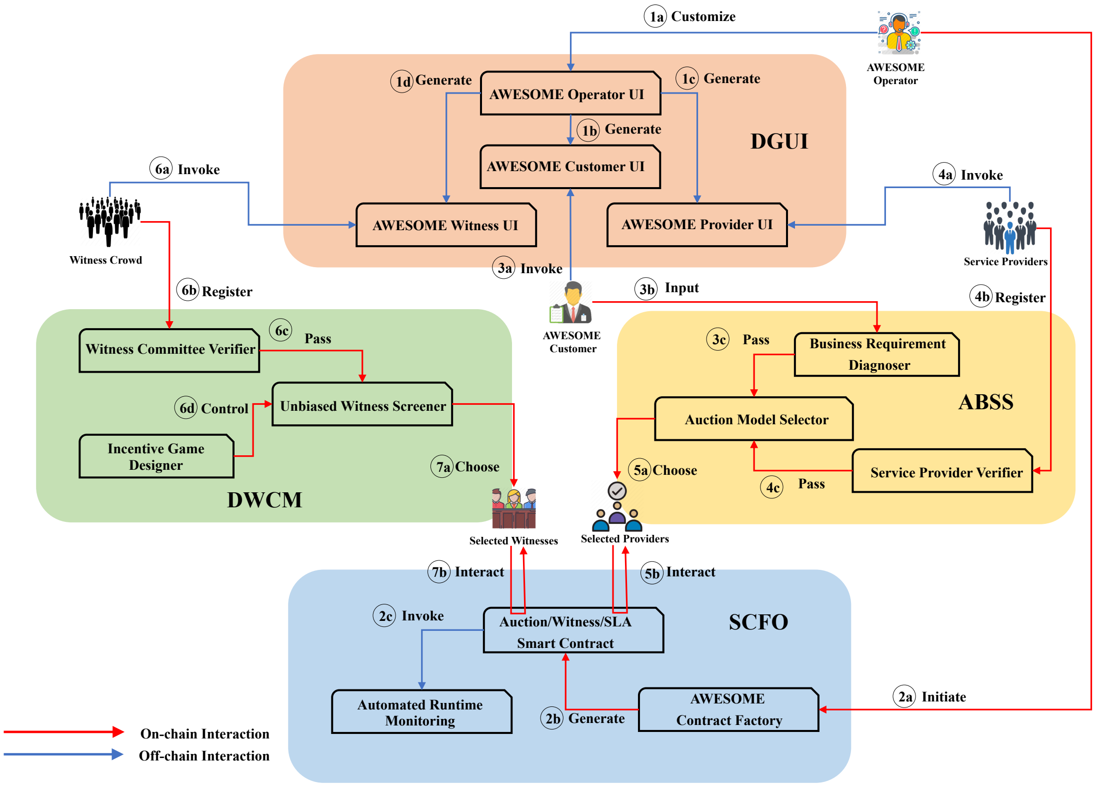


Figure 3.2: The process flow of the AWESOME framework and the related stakeholders in the decentralized service ecosystem.

As shown in Figure 3.2, the overall workflow of the AWESOME framework can be described as the following steps (using a reverse auction example). First of all, an AWESOME operator calls the *DGUI* subsystem to customize and generate customer, provider, and witness user interfaces (UIs) for the current use case (steps 1a, 1b, 1c, and 1d). The AWESOME operator then calls the *SCFO* subsystem to initiate a contract factory (step 2a). This contract factory automatically generates the required auction, witness, and SLA smart contracts to ensure trustworthy interaction between different participants (step 2b). Meanwhile, it also invokes a runtime monitor for these contracts and returns the monitoring result to the AWESOME operator (step 2c). Next, an AWESOME customer invokes the UI to transfer the specific business requirement to *ABSS* subsystem (steps 3a and 3b). Based on this requirement, an auction model is selected and configured to wait for providers to submit their bids (step 3c). The decentralized service providers then start registering and verifying in the *ABSS* subsystems through their UIs (steps 4a, 4b, and 4c). When there are enough bidder candidates, smart contracts automatically start the auction process to find qualified providers (step 5a). Then, the witnesses invoke their UIs to register and verify in the *DWCM* subsystem (steps 6a, 6b, and 6c). The *DWCM* subsystem performs game design and an unbiased witness screener to choose the appropriate witness to form the witness committee (steps 6d and 7a). Finally, the selected providers collaborate to provide cloud services, and selected witnesses monitor the SLAs to win profits (steps 5b and 7b). The service fee and witness fee will be paid and enforced with cryptocurrency using smart contracts when the cloud service ends.

In the entire AWESOME workflow, *DGUI* provides a customizable graphical interaction environment to support user-to-user interactions in business processes. *ABSS* selects candidate providers through an effective auction mechanism. *DWCM* ensures trustworthy SLA enforcement through truth-telling witness monitoring. Finally, *SCFO* provides automated smart contract support for the entire process. The four subsystems form a dynamic ecosystem that provides services to AWESOME users collaboratively.

3.3 The AWESOME DApp Demonstration

This section presents a prototype system of the AWESOME framework. The DApp of the AWESOME prototype helps AWESOME operators and users trade cloud services in a decentralized service marketplace. This DApp will contain customizable auction and witness models for service provisioning and SLA violation monitoring. Specifically, we first examine different design choices and implementation options. Then we leverage a use case to demonstrate how the different roles would utilize the DApp.

3.3.1 Design Choices

We discuss the design choices regarding the auction models, the blockchain infrastructure, and the smart contract protocols.

Auction Models

In order to support dynamic business requirements, we should allow that both customers and providers can be initiators and bidders. This produces a more customizable system that aids in defining a broader range of auction possibilities. Therefore, at the highest level, the two types of auctioning that the system supports are: 1) forward auctions, where the provider is the initiator and the customers submit competing bids; 2) reverse auctions, where the customer is the initiator, and the providers submit competing bids. Specifically, AWESOME is designed to support the following eight auction models:

- **English Auction**: This is an ascending bid auction, where the price is gradually raised until only one final bidder remains, and that bidder wins the service at the finalized price.
- **Dutch Auction**: This is a descending bid auction, also known as a clock auction or open-outcry descending-price auction. The seller starts at a high price and lowers it until a bidder accepts the price.
- **First Price Sealed-Bid Auction**: In this auction, bidders submit sealed bids simultaneously to the seller, and the highest bidder wins and pays the value of their bid.
- **Second Price Sealed Bid Auction**: In this auction, bidders submit sealed bids and the highest bidder wins again, but the price they pay is the value of the second-highest bid. It is also known as the Vickrey auction, which encourages truthful bidding in terms of mechanism design [116].

We can also implement the symmetrical version of these four types for reverse auctions.

- **Reverse English Auction**: In this auction, the price is decremented by competing providers until no one bids at a lower price. The provider offering the lowest price wins the auction and provides the service at that price.
- **Reverse Dutch Auction**: This auction begins at a very low price for the service and gradually increases until a service provider accepts to provide the service at that specific price.
- **Reverse First Price Sealed-Bid Auction**: In this auction, providers submit sealed bids simultaneously to the customer. The one with the lowest bid wins and pays the value of their bid.
- **Reverse Second Price Sealed Bid Auction**: In this auction, providers again submit sealed bids, the lowest bid again wins, but the price he should pay is the value of the second-lowest bid.

Blockchain Infrastructure

In general, blockchains can be permissionless or permissioned. Our modular AWESOME framework is not limited to the underlying blockchain infrastructure. AWESOME aims to build a decentralized cloud marketplace using both permissionless and permissioned blockchains. Some existing popular platforms may include:

- **Ethereum**: It is an open-source permissionless blockchain platform with smart contract and cryptocurrency functions. It provides a decentralized mechanism to handle peer-to-peer smart contract transactions through its proprietary Ethereum Virtual Machine. We are currently using the Ethereum blockchain to develop AWESOME smart contracts and DApps.
- **Permissioned blockchain platforms**, e.g., Hyperledger Fabric, Hyperledger Sawtooth, Hyperledger Iroha, Hyperledger Besu, and IOTA, can also be implemented to support our decentralized cloud marketplace. The main reason for choosing a permissioned blockchain is the availability of a more efficient consensus mechanism, which increases scalability and reduces wasted resources. These platforms have been investigated in many research studies and commercial projects [232].

Smart Contract Protocols

To meet the requirements of the AWESOME framework to build a decentralized cloud marketplace, we designed three smart contracts (i.e., auction contract, witness contract, and SLA contract) to support trustworthy and fair interactions between different stakeholders, as shown in Figures 3.3 to 3.5. Specifically, the auction contract is responsible for managing the auction process. The witness contract is responsible for the registration and management of auction witnesses, as well as the calculation of witness reporting results and corresponding witness fees. Furthermore, the SLA contract is used to build and manage a trustworthy SLA lifecycle. It should be noted that we leverage a contract factory to manage and generate subcontracts instead of developing different contracts separately in the AWESOME framework, as this is a more secure and efficient way [136].

The sequence diagram in Figure 3.6 shows the interaction between the contract factory and different subcontracts. First, an AWESOME operator calls the contract factory to create a new auction contract. Next, an auction contract with a customized auction rule for business requirements is built to support a transparent and automated auction process. In this case, service providers can register and submit their bids for services on the blockchain. The auction contract then selects the winning providers based on the highest k bids and generates k SLA contracts for each provider. When the auction is settled (note that the services have not been delivered yet), the AWESOME operator calls the contract

Protocol: Auction Contract

An auction contract is a smart contract used for cloud auctions in a decentralized cloud marketplace. It is defined as a tuple $AC = (Rule, p_{reserve}, \mathbf{b}, \mathbf{T}, N, \mathbf{b}')$.
Input:

- *Rule* is the bidding rule for the cloud auction. It can be one of the auction models presented in Section 3.3.1.
- The reserve price of the auction initiator $p_{reserve}$.
- A set of bids from different bidders, $\mathbf{b} = (b_1, b_2, \ldots, b_n)$.
- A set of time windows to specify different auction phases $\mathbf{T} = \{T_1, T_2, T_3, T_4, T_5\}$.

Output:

- A set of selected winner bidders $N = \{1, 2, \ldots, k\}$ based on the auction *Rule*, and their corresponding bids $\mathbf{b}' = \left(b'_1, b'_2, \ldots, b'_k\right)$.

Process:

1. *Setup Auction*: The auction initiator (i.e., customer or provider) starts to 1) define the auction requirement, 2) specify an auction *Rule*, and 3) upload his/her $p_{reserve}$.
2. *Bidder Register*: Other customers or providers register as bidders in AC between $[T_1, T_2]$ to participate in the auction.
3. *Check Bidder Number*: The auction initiator checks the number of bidders between $[T_2, T_3]$; if there are enough bidders, proceed to the next step, otherwise terminate.
4. *Submit Bid*: Registered providers submit their bids $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ to the AC and prepay a deposit between $[T_4, T_5]$.
5. *(Reveal Bids and Reserve Price)*: These two steps are optional and only for sealed-bid auctions. Bidders and the auction initiator reveal their encrypted (using hash functions) $\mathbf{b}$ and $p_{reserve}$ submitted in previous steps.
6. *Place Bids*: Based on the auction *Rule*, winner bidders $N = \{1, 2, \ldots, k\}$ and their bids $\mathbf{b}' = \left(b'_1, b'_2, \ldots, b'_k\right)$ are calculated and published.
7. *Withdraw Deposit*: Bidders and the auction initiator withdraw their prepaid deposits after T_5.

Figure 3.3: Protocol: Auction Contract

factory again to generate a witness contract that contains customized incentive mechanisms to encourage truth-telling witnesses. More details about a game theory-based witness payoff design are discussed in our previous research [231]. Then, different winner providers can start to deliver cloud services off-chain while the witnesses start to monitor all the services; if the QoS satisfies the requirements in the SLA contract, there is no violation; otherwise, there is a violation. The result of service monitoring is also returned to the auction contract to determine the status of the auction.

3.3.2 Use Case Demonstration

We use the business process model in Figure 3.7 to demonstrate how our AWESOME DApp works. There are three parties of stakeholders who interact with the

Protocol: Witness Contract

A witness contract is a smart contract used for monitoring the SLAs. It is defined as a tuple of three elements $WC = (\mathbf{W}, F_{\text{witness}}, \varphi)$.

Input:

- A set of monitoring results $\mathbf{W} = \left(W^1, W^2, \dots, W^k\right)$ for k SLAs from m witnesses, where $W^k = \left\{w_1^k, w_2^k, \dots, w_m^k\right\}$ is the monitoring results reported for cloud SLA k.
- The payment for each witness F_{witness}, which is used to motivates witnesses to monitor the federated cloud services. It is transferred in advance by AC.
- The penalty function $\varphi(w)$ for witness i, which defines the penalty to be deducted from his/her F_{witness}. For a particular SLA, the closer the results reported by witness j are to those of others, the less penalty he receives.
- A set of time windows to specify different monitoring phases $\mathbf{T} = \{T_6, T_7, T_8, T_9\}$.

Output:

- $\mathbf{SLA}_{\text{violate}} = \left(SLA^1_{\text{violate}}, SLA^2_{\text{violate}}, \dots, SLA^{\text{k}}_{\text{violate}}\right)$ is a set of SLA violation reports for k SLAs, where $\mathbf{SLA}^{\text{j}}_{\text{violate}} = \{0, 1\}$ is used to denote the finial result of whether SLA j is violated.

Process:

1. *Witness Register*: Normal blockchain users register as witnesses in WC between $[T_6, T_7]$ to participate in the SLA monitoring.
2. *Check Auction Settled*: Check the current auction state between $[T_7, T_8]$; if the auction is settled successfully, proceed to the next step, otherwise terminate.
3. *Submit Reports*: Registered witnesses submit their encrypted (using hash functions) reports $\mathbf{W} = \left(W^1, W^2, \dots, W^k\right)$ to WC and prepay a deposit between $[T_8, T_9]$.
4. *Reveal Reports*: Witnesses reveal their encrypted W submitted in the previous step, using their private keys and the real messages.
5. *Calculate Witness Fee*: WC calculates the SLA_{violate} for k SLAs and the payment of each witness based $\varphi(w)$ and F_{witness}.
6. *Withdraw Witness Fee*: Witnesses withdraw their corresponding witness fees after T_9.

Figure 3.4: Protocol: Witness Contract

AWESOME DApp to complete a reverse auction. In this auction, the customer acts as the purchaser of the cloud service and the initiator of the auction. The providers need to compete for bids to get the right to sell their services. The entire AWESOME business process is described as follows. When the AWESOME DApp is launched, it first invokes the AWESOME contract factory to generate the auction contract to support the auction process management. At this time, a customer can register, set up, and post an auction through the customer UI, as illustrated in Figure 3.8a. This auction invitation is displayed on the AWESOME DApp to make it visible to providers. When noticing the auction invitations, different providers can register as bidders and submit their bids through the provider UI, as shown in Figure 3.8b. When enough bids are received to meet the customer's requirement, the auction is settled, and the winning providers are selected.

Protocol: SLA Contract

An SLA contract is a smart contract for defining service agreements between service customer and provider. It is defined as a tuple $\mathbf{SLAC} = \big(c_\mathrm{i}, SP_\mathrm{i}, QoS_\mathrm{i}, F^\mathrm{i}_\mathrm{service}, F^\mathrm{i}_\mathrm{witness}, \varGamma(SLA_\mathrm{violate})\big)$.

Input:

- A customer c_i and a service provider SP_i. SP_i promises to provide the service quality of QoS_i, and c_i promises to purchase the services at the price of $F^\mathrm{i}_\mathrm{service}$.
- $F^\mathrm{i}_\mathrm{witness}$ is the witness fee of the cloud SLA_i, which is paid by c_i and SP_i together.
- The SLA compensate rule $\varGamma(SLA^\mathrm{i}_\mathrm{violate})$; if SLA_i is violated, $F^\mathrm{i}_\mathrm{service}$ will be refunded (in proportion) to c_i. Otherwise $F^\mathrm{i}_\mathrm{service}$ will be transferred from **SLA** to SP_i.

Output:

- A set of sub SLAs for different services $\mathbf{SLA} = \{SLA_1, SLA_2, \ldots, SLA_\mathrm{k}\}$, where $SLA_\mathrm{i} = \big(c_\mathrm{i}, SP_\mathrm{i}, QoS_\mathrm{i}, F^\mathrm{i}_\mathrm{service}, F^\mathrm{i}_\mathrm{witness}, \varGamma(SLA_\mathrm{violate})\big)$ indicates that c_i and SP_i signed SLA_i together.

Process:

1. *Publish Service*: A service party i (i.e., customer or provider) publishes service details and specifies the QoS_i.
2. *Setup SLA*: The SLA is setup based on $F^\mathrm{i}_\mathrm{service}$, $F^\mathrm{i}_\mathrm{witness}$, and $\varGamma(SLA^\mathrm{i}_\mathrm{violate})$.
3. *Accept SLA*: Other service parties choose to accept the SLA or not; if the SLA is accepted, **SLAC** outputs a settled SLA. Otherwise, proceed to the next step.
4. *Cancel SLA*: Service parties cancel the SLA preset in **SLAC**.
5. *Check SLA Violation*: Service parties check whether an SLA is violated or not.
6. *Withdraw Service Fee*: After the service is delivered, the service fee is calculated and allocated based on $\varGamma(SLA^\mathrm{i}_\mathrm{violate})$.

Figure 3.5: Protocol: SLA Contract

After that, the AWESOME contract factory will generate SLA contracts to prepare for service delivery and monitoring. The customer and providers need to sign and confirm the SLA contracts in AWESOME DApp, respectively. At the same time, a witness contract is also generated. At this point, the auction initiator (i.e., the customer) needs to define the rules for witnesses, including the number of witnesses, the minimum consensus percentage required to confirm a violation, the time window for submitting reports, and the rewards and penalties for each witnesses' report. This is illustrated in Figures 3.9a and 3.9b. Then, witnesses can register and interact with the contract through the Witness UI. The cloud service is officially launched only when all sub-SLAs are confirmed, and there are enough witnesses for SLA monitoring. Next, the customer and providers perform off-chain cloud service provisioning and consumption. Witnesses perform continuous SLAs monitoring and report the monitoring results to the AWESOME DApp, as shown in Figure 3.10. Based on the results reported by the witness committee, SLA violation is confirmed: when there is a violation, the service fee prepaid by the customer is refunded; while when the service is completed as agreed in the SLA,

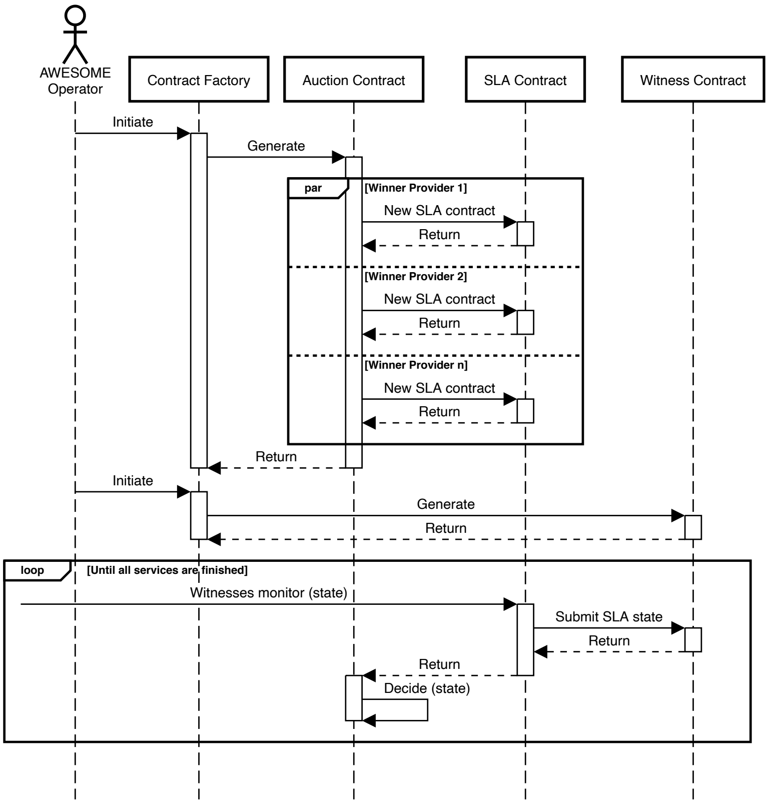


Figure 3.6: The sequence diagram of the AWESOME smart contracts interactions.

the providers can withdraw their corresponding service fees. Finally, witness fees are calculated and allocated based on the monitoring results of the witness game.

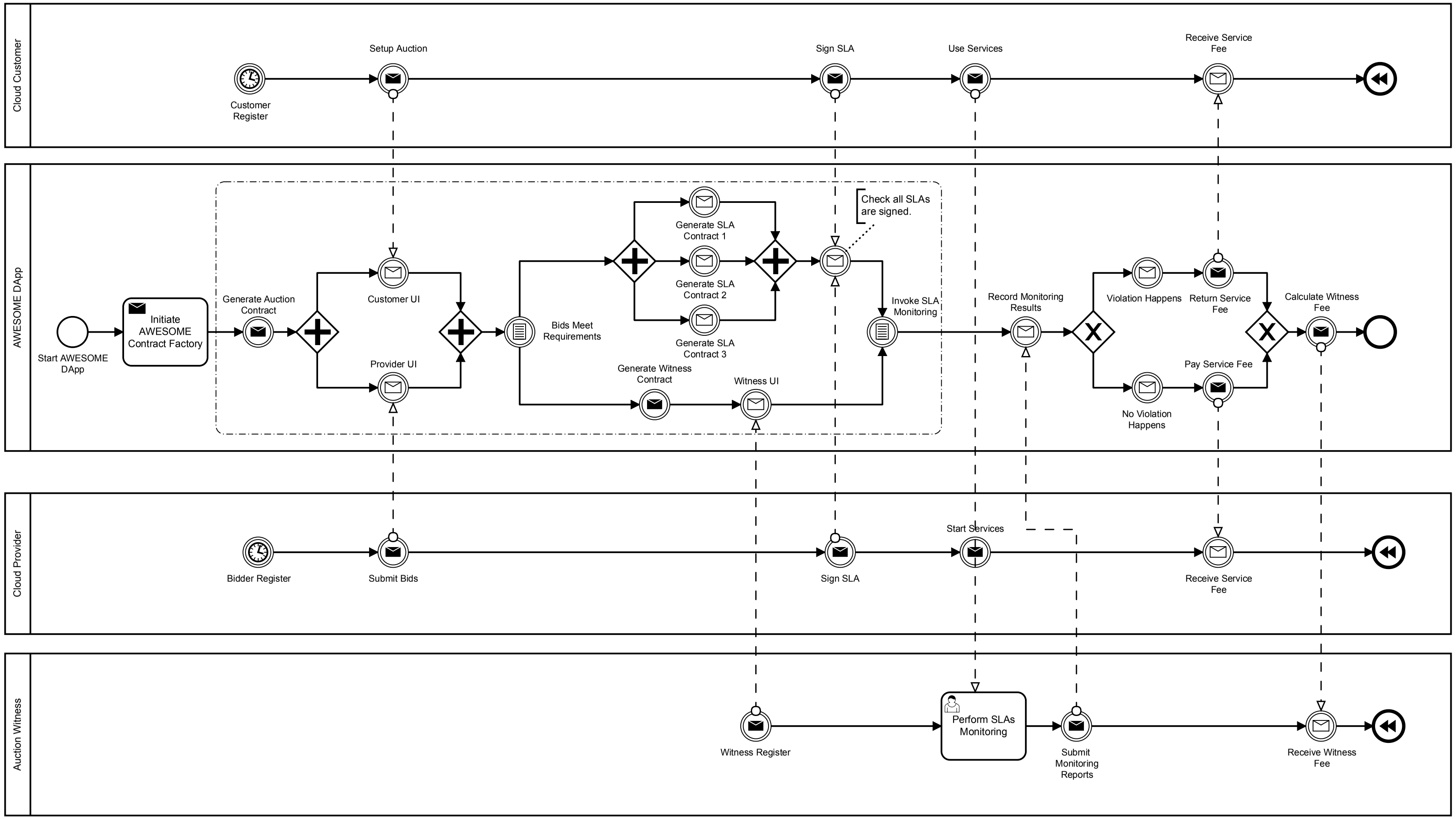
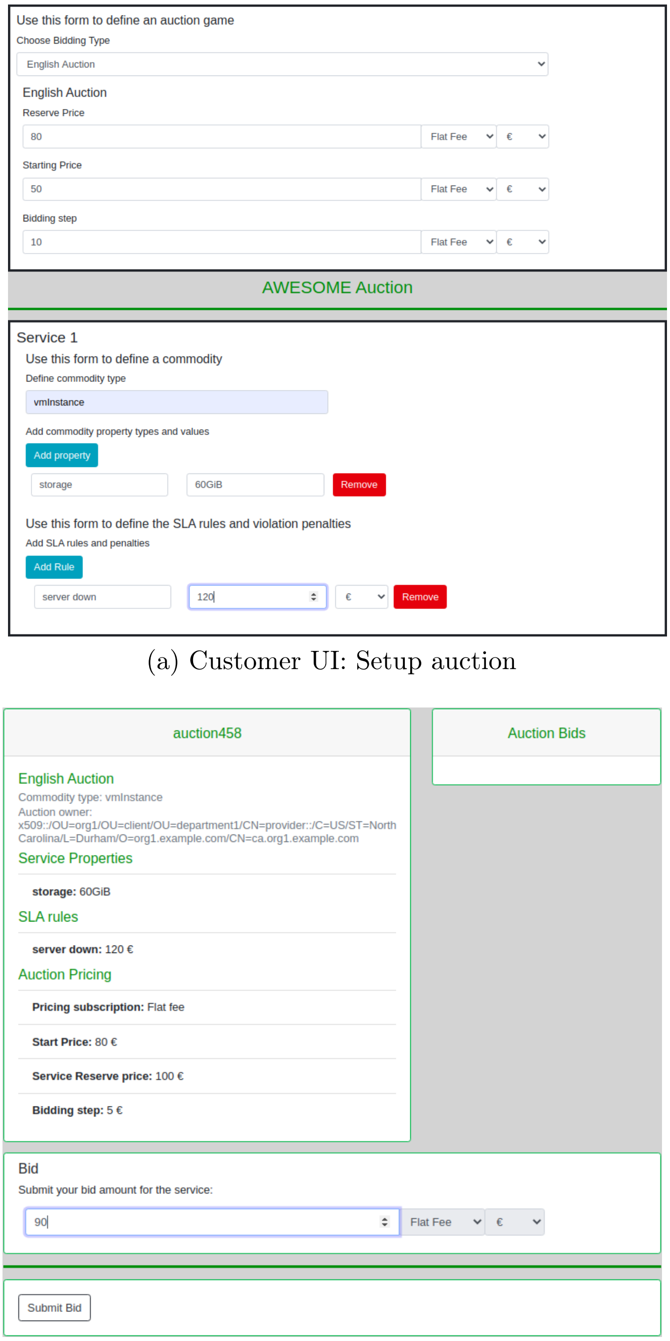


Figure 3.7: The business processes in an AWESOME service trading activity.



Use this form to define an auction game

Choose Bidding Type: English Auction

English Auction

Reserve Price: 80 Flat Fee €

Starting Price: 50 Flat Fee €

Bidding step: 10 Flat Fee €

AWESOME Auction

Service 1

Use this form to define a commodity

Define commodity type: vmInstance

Add commodity property types and values

storage 60GiB

Use this form to define the SLA rules and violation penalties

Add SLA rules and penalties

server down 120 €

(a) Customer UI: Setup auction

auction458

English Auction

Commodity type: vmInstance

Auction owner: x509::/OU=org1/OU=client/OU=department1/CN=provider::/C=US/ST=North Carolina/L=Durham/O=org1.example.com/CN=ca.org1.example.com

Service Properties

storage: 60GiB

SLA rules

server down: 120 €

Auction Pricing

Pricing subscription: Flat fee

Start Price: 80 €

Service Reserve price: 100 €

Bidding step: 5 €

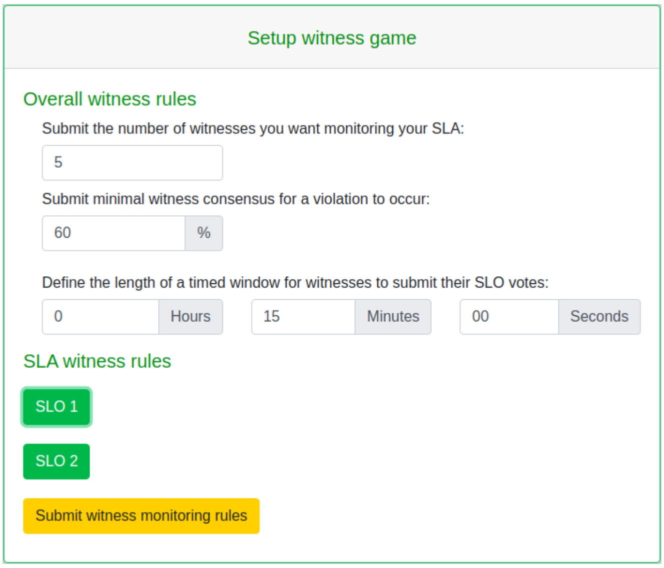
Auction Bids

Bid

Submit your bid amount for the service: 90 Flat Fee €

(b) Provider UI: Submit bid

Figure 3.8: The auction initiator (i.e., the customer in this auction) sets the auction rules, and bidders (i.e., providers in this auction) submit their bids.



Setup witness game

Overall witness rules

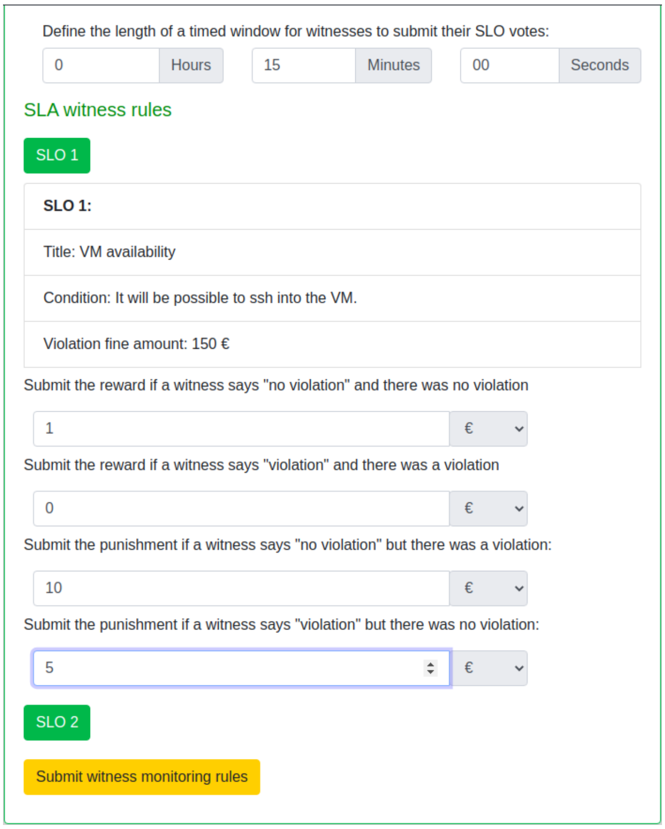
Submit the number of witnesses you want monitoring your SLA: 5

Submit minimal witness consensus for a violation to occur: 60 %

Define the length of a timed window for witnesses to submit their SLO votes: 0 Hours 15 Minutes 00 Seconds

SLA witness rules

(a) Customer UI: Setup witness game



Define the length of a timed window for witnesses to submit their SLO votes: 0 Hours 15 Minutes 00 Seconds

SLA witness rules

SLO 1:

Title: VM availability

Condition: It will be possible to ssh into the VM.

Violation fine amount: 150 €

Submit the reward if a witness says "no violation" and there was no violation: 1 €

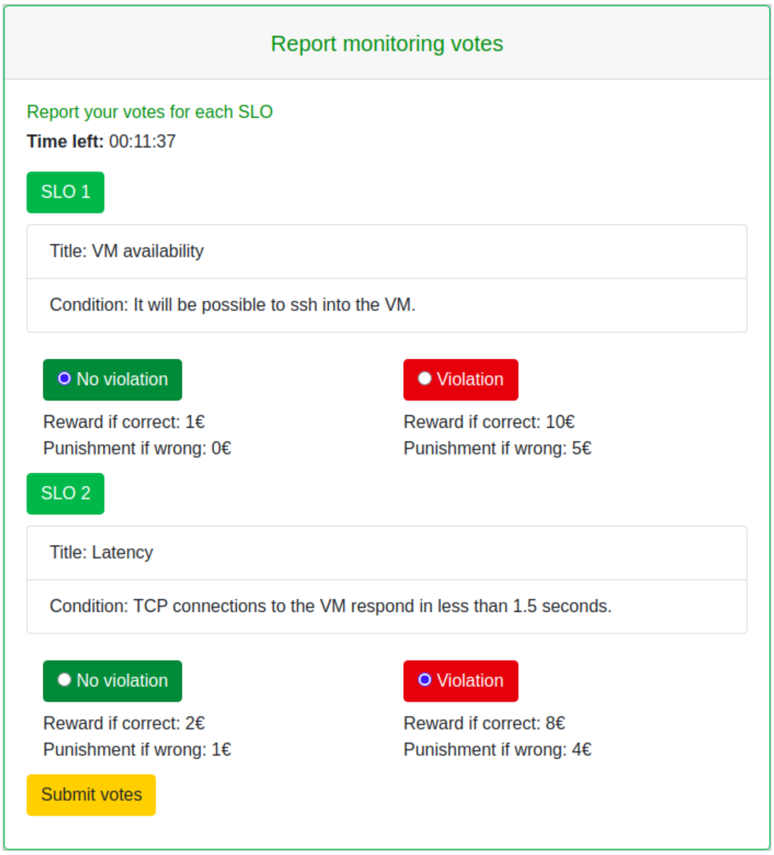
Submit the reward if a witness says "violation" and there was a violation: 0 €

Submit the punishment if a witness says "no violation" but there was a violation: 10 €

Submit the punishment if a witness says "violation" but there was no violation: 5 €

(b) Customer UI: Define rewards and penalties

Figure 3.9: The auction initiator (i.e., the customer in this auction) sets the monitoring rules for witnesses.



Report monitoring votes

Report your votes for each SLO

Time left: 00:11:37

SLO 1

Title: VM availability

Condition: It will be possible to ssh into the VM.

◉ No violation
Reward if correct: 1€
Punishment if wrong: 0€

○ Violation
Reward if correct: 10€
Punishment if wrong: 5€

SLO 2

Title: Latency

Condition: TCP connections to the VM respond in less than 1.5 seconds.

○ No violation
Reward if correct: 2€
Punishment if wrong: 1€

◉ Violation
Reward if correct: 8€
Punishment if wrong: 4€

Submit votes

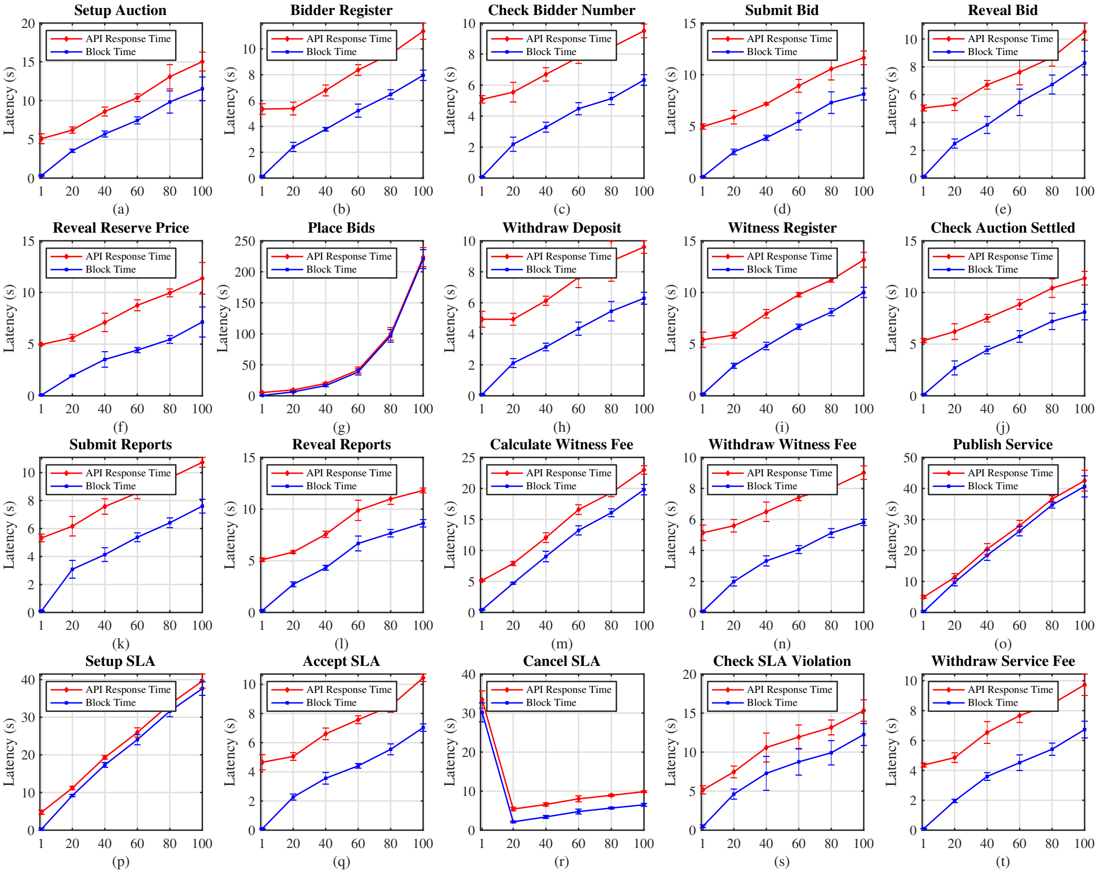
Figure 3.10: Witness UI: A witness starts to submit the report for SLA violations.

3.4 Experiments and Validations

In this section, we present the evaluation and validation of the AWESOME framework. We identified two key metrics that affect the performance of our AWESOME framework: latency and cost.

3.4.1 Latency Analysis

To evaluate the performance of the AWESOME DApp and the feasibility of the model, we first simulated cloud auction and SLA management scenarios with different player numbers. Then, the latency is tested in the local Ethereum blockchain and calculated in two ways: 1) the response time of the AWESOME DApp; 2) the difference between the block timestamps [190]. The former reflects the latency of executing transactions in our DApp, and the latter demonstrates the transaction processing time of the underlying blockchain. In addition, we simulated two cases of blockchain mining network congestion. The “best” case implies that there are enough miners to process transactions promptly. In contrast, the “average” situation indicates seconds of delays for transaction processing due



Setup Auction (a)
Bidder Register (b)
Check Bidder Number (c)
Submit Bid (d)
Reveal Bid (e)
Reveal Reserve Price (f)
Place Bids (g)
Withdraw Deposit (h)
Witness Register (i)
Check Auction Settled (j)
Submit Reports (k)
Reveal Reports (l)
Calculate Witness Fee (m)
Withdraw Witness Fee (n)
Publish Service (o)
Setup SLA (p)
Accept SLA (q)
Cancel SLA (r)
Check SLA Violation (s)
Withdraw Service Fee (t)

Figure 3.11: Variation of execution latency of functional interfaces in AWESOME smart contracts for different number of players (under the best mining network).

to network congestion.

Figure 3.11 consists of 20 plots, showing the performance of 20 function interfaces (introduced in Figures 3.3 to 3.5) in a "best" blockchain mining network. Overall, the response time of the AWESOME DApp API is a few seconds longer than the blockchain block time, which is in line with our expectations. It can be seen that the execution time of most of the function interfaces increases linearly with the number of players. One exception is *Place Bids*, which has an exponentially increasing trend. This is because this function requires on-chain calculations to place bids and therefore takes significantly more time when the number of players increases or the auction data becomes complex. Some other functions that take longer time include *Setup Auction*, *Calculate Witness Fee*, *Publish Service*, and *Setup SLA*. Except for these functions, the latency of all others holds at a low level (10 to 15 seconds with 100 players).

Similarly, Figure 3.12 shows the performance of 20 functional interfaces on an "average" mining network. It can be found that the DApp API response time and the blockchain block time have both increased significantly; the two latency

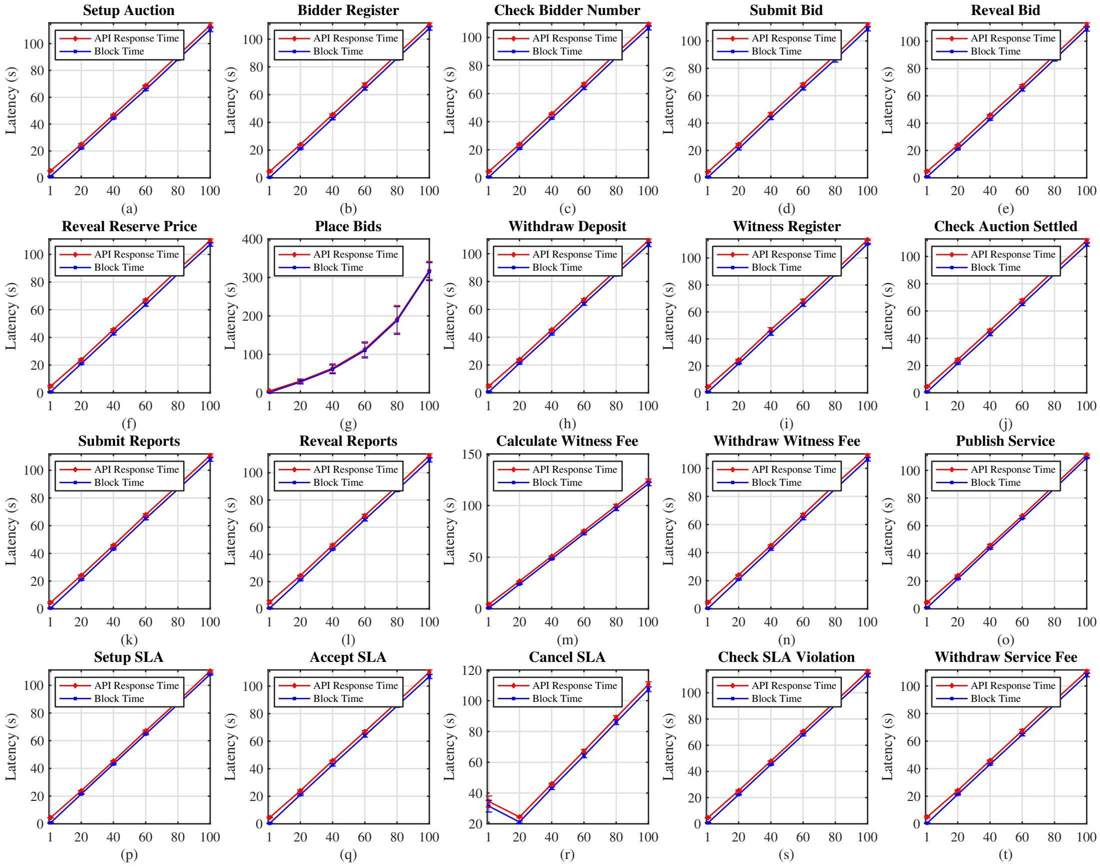


Figure 3.12: Variation of execution latency of functional interfaces in AWESOME smart contracts for different number of players (under the average mining network).

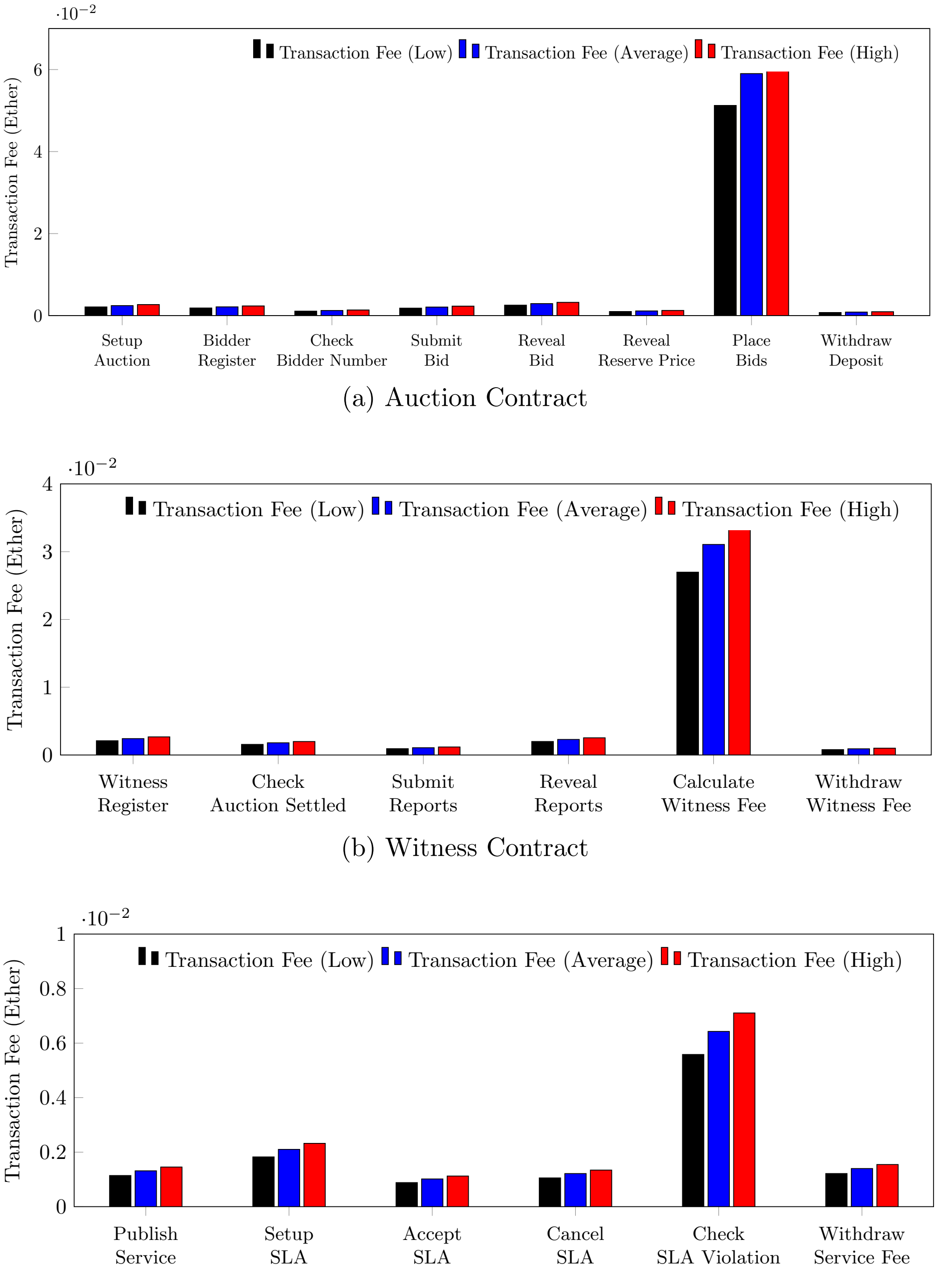
also tend to be close to each other because of the huge increase in the total time needed to process blockchain transactions. Nevertheless, all latency is maintained within a few minutes when the number of players increases. A strange observation is that the latency is high in the *Cancel SLA* function when the player number is 1. By analyzing our experimental workflow, we found that this is because he needs to process a large number of SLAs generated from the previous step. This also proves that the latency is influenced by the player numbers and the complexity of the data to be processed. In summary, we conclude that the AWESOME framework has a good performance in terms of execution latency. Some individual functions that require complex calculations have longer time delays, but they may not be time-critical in the auction process. We also conclude that the main factor affecting DApp latency is the performance of the underlying blockchain; namely, the latency is heavily dependent on the congestion of the blockchain mining network.

3.4.2 Cost Analysis

Next, we measured the gas consumption and transaction fee (Ether) of each function interface in AWESOME smart contracts, as shown in Figure 3.13 and Table 3.2. These functions have built-in access control mechanisms; only specific stakeholder groups can access and call them. Specifically, three transaction submission speed modes (i.e., low, average, and high) were tested.[4] By analyzing the testing results, we can find that the transaction fees of most function interfaces are maintained at a relatively low level (less than 0.01 ether), except for only three special cases, namely *Place Bids* (auction contract), *Calculate Witness Fee* (witness contract), and *Check SLA Violation* (SLA contract). These function interfaces require specific computational tasks on-chain and therefore need more transaction fees to pay for miners.

Table 3.3 further shows the transaction fee of each AWESOME user (converted into USD). Overall, the customer is the beneficiary and initiator of the service auction and should therefore bear more commission fees. Providers have lower transaction costs since they only join the model as bidders. Besides, the transaction fee per witness is economical (about $15-20), which ensures that they have sufficient incentive to join the SLA monitoring activities. It is worth noting that although the customer pays over $200 to initiate the auction, this fee is fixed and independent of the final service price. In contrast, some popular online auction platforms charge a percentage of the sale price as a commission (e.g., eBay charges 12.9% of the sale price). Therefore, our model has a price advantage, especially when the price of the auctioned cloud services becomes very expensive. On the other hand, if the commission is higher than the cost of the cloud service itself, our model will not be suitable for a public blockchain like Ethereum. A fee-free permissioned blockchain (e.g., Hyperledger Fabric), in this case, can be used as an alternative and the whole model is still valid.

[4] The estimated transaction confirmation duration for three modes are 16 minutes, 2 minutes and 19 seconds, and 30 seconds, respectively. Data was collected on April 30, 2021, at https://etherscan.io/gastracker



(a) Auction Contract

(b) Witness Contract

(c) SLA Contract

Figure 3.13: The transaction fee of each function interface in AWESOME smart contracts.

Table 3.2: The gas consumption and access control mechanisms of function interfaces in AWESOME smart contracts.

Smart Contract	Access Control	Function Interface	Gas Consumption	Transaction Fee (Ether)		
				Low	Average	High
Auction Contract	Customer	Setup Auction	63694	0.002101902	0.002420372	0.002675148
	Provider	Bidder Register	55835	0.001842555	0.00212173	0.00234507
	Customer	Check Bidder Number	32420	0.00106986	0.00123196	0.00136164
	Provider	Submit Bid	54555	0.001800315	0.00207309	0.00229131
	Provider	Reveal Bid	76605	0.002527965	0.00291099	0.00321741
	Customer	Reveal Reserve Price	29630	0.00097779	0.00112594	0.00124446
	Customer	Place Bids	1552698	0.051239034	0.059002524	0.065213316
	Customer & Provider	Withdraw Deposit	22242	0.000733986	0.000845196	0.000934164
Witness Contract	Witness	Witness Register	63630	0.00209979	0.00241794	0.00267246
	Customer	Check Auction Settled	47298	0.001560834	0.001797324	0.001986516
	Witness	Submit Reports	28159	0.000929247	0.001070042	0.001182678
	Witness	Reveal Reports	60330	0.00199089	0.00229254	0.00253386
	Customer	Calculate Witness Fee	817390	0.02697387	0.03106082	0.03433038
	Witness	Withdraw Witness Fee	24017	0.000792561	0.000912646	0.001008714
SLA Contract	Provider	Publish Service	34587	0.001141371	0.001314306	0.001452654
	Provider	Setup SLA	55275	0.001824075	0.00210045	0.00232155
	Customer	Accept SLA	26721	0.000881793	0.001015398	0.001122282
	Provider	Cancel SLA	31926	0.001053558	0.001213188	0.001340892
	Provider	Check SLA Violation	169151	0.005581983	0.006427738	0.007104342
	Customer & Provider	Withdraw Service Fee	36785	0.001213905	0.00139783	0.00154497

Table 3.3: The transaction fee of each AWESOME user in a specific auction event.

User	Gas Consumption	Transaction Fee			USD		
		Low	Average	High	Low	Average	High
Customer	2628878	0.086752974	0.099897364	0.110412876	230.94335702592	265.93477475712	293.92790894208
Provider	536961	0.017719713	0.020404518	0.022552362	47.17129358304	54.31845927744	60.03619183296
Witness	176136	0.005812488	0.006693168	0.007397712	15.47330805504	17.81774866944	19.69330116096

3.5 Conclusion

In this chapter, we proposed a novel AWESOME framework for building a decentralized cloud marketplace. The framework is enhanced by auction models and witness mechanisms, and support interactions between service providers, customers, and witnesses to complete trustworthy auctions and SLA enforcement. Compared with classic cloud service models, our framework leverages blockchain and smart contracts for decentralized service auctions; transaction efficiency is ensured through customizable auction models, and auction trustworthiness is guaranteed by the monitoring of decentralized witnesses. We also prototyped an AWESOME DApp using the Ethereum blockchain. It contains customizable GUIs and advanced smart contract protocols to support the SLA business management process. We evaluated the latency and cost of our model and DApp. The experimental results demonstrated that our model is economical and feasible.

Chapter 4

Towards an Incentivized AWESOME Framework: A Bayesian Game Approach for Federated Cloud Services

In the previous chapter, we have demonstrated the feasibility of AWESOME framework for cloud service auction and SLA management. The effectiveness of the model is mainly achieved by blockchain and smart contract technologies. However, the framework design so far is not fully satisfactory. This is because the current framework lacks user incentives that are crucial for the continuous maintenance of the AWESOME ecosystem. Effective incentives are needed to support cloud auctions and attract witnesses to join the model to make trustworthy SLA violation judgments. In this chapter, we propose an incentivized AWESOME auction model using Bayesian game theory. Specifically, the AWESOME framework is enhanced with two Bayesian Nash Equilibriums (BNEs); the first BNE enables the selection of cost-effective providers to construct the federated cloud services, while the second BNE ensures consistent and trustworthy monitoring of federated SLAs. Moreover, a timed message submission (TMS) algorithm is proposed to protect the auction privacy during the message submission phase. This chapter validates the equilibrium results of two BNEs and implements the proposed model on the Ethereum blockchain. The analytical and experimental results demonstrate the trustworthiness and cost-effectiveness of our model.

This chapter is based on:

- **Zeshun Shi**, Huan Zhou, Cees de Laat, and Zhiming Zhao. “A Bayesian Game-Enhanced Auction Model for Federated Cloud Services Using Blockchain”. *Future Generation Computer Systems* (2022): 136, pp.49-66.

The remainder of the chapter is organized as follows. Section 4.1 briefly explain the problem statement. Section 4.2 introduces the overall architecture and process of the proposed federated cloud auction model. Section 4.3 dives into the key techniques, including a federated cloud partition model, two Bayesian games with the analysis of BNEs, and the TMS algorithm. Section 4.4 presents the evaluation and implementation details. Finally, the chapter is concluded in Section 4.5.

4.1 Problem Statement

The AWESOME framework introduced in the previous section can be used to build a decentralized cloud marketplace. The framework supports interactions between service providers, customers, and witnesses to accomplish trusted auctions and SLA enforcement. However, challenges regarding enhancing mutual trust relationships and increasing the level of user participation have not yet been fully addressed. The current solution is to simply use blockchain and smart contracts as trustworthy devices for stakeholders, but there is no effective incentive mechanism for large-scale user interactions. There is an urgent need to design effective incentives to motivate users to participate in auctions and SLA monitoring, and to support the continous maintenance of the AWESOME framework.

In our previous work [231], we introduced the idea of a decentralized SLA monitoring mechanism for blockchain using the complete information game theory. However, such a design is not fully satisfactory. First, there is a basic assumption in a complete information game that knowledge about other players is available to all players. This is a strong assumption that is not practical in the real world. Besides, the model can only implement naive incentives to determine whether the service is violated through a boolean value without the ability to monitor federated SLAs at the same time. In economics, Bayesian games are often used to model economic situations where players may not know various features of the environment. In this context, the incomplete information (Bayesian) game is more suitable for our cloud auction and SLA management scenario, where each player knows his own utility, but not the information (type and utility) about other players.

4.2 Bayesian Game-Enhanced Federated Cloud Auction Model

Based on current research gaps, we propose our Bayesian game-enhanced federated cloud auction model in detail. In practice, cloud customers usually use cloud services from federated service providers to improve fault tolerance and reliability. There are many providers with similar functions on the market. In contrast to the spot instance pricing that requires customers to bid for resources, providers need to offer flexible pricing strategies and submit bids to sell their services in our model. It should be noted that the AWESOME framework aims to design a generic framework to support different auction scenarios. In this chapter, a reverse auction model for federated cloud services is chosen as a use case to demonstrate the design of incentive mechanisms.

4.2.1 Architecture Overview

Figure 4.1 shows the architecture overview of the proposed auction model. Generally, there are three roles involved: 1) a cloud customer who pays and consumes the federated cloud services. In the auction model, the customer works as a service purchaser and publishes auction invitations; 2) multiple cloud providers that collaborate to provide federated cloud services. They act as bidders in the auction model and fight for the right to sell services through bidding activities; and 3) a new role called auction witnesses is introduced to monitor the federated SLAs and ensure the successful delivery of the auctioned cloud services.[1]

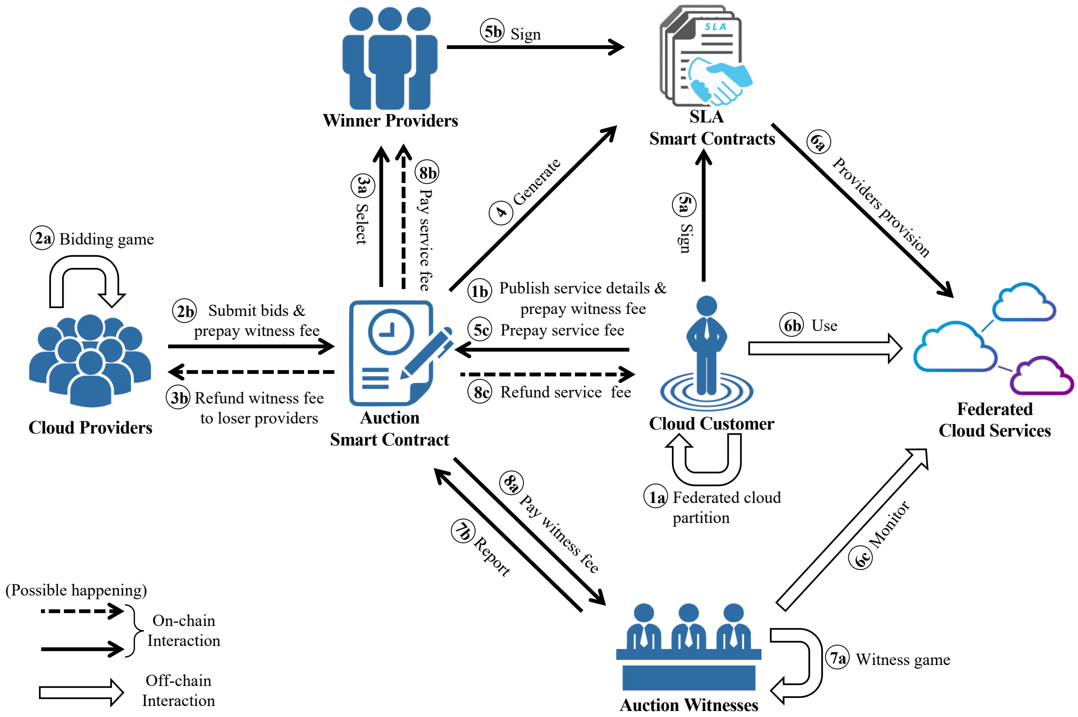


Figure 4.1: Architecture overview of the Bayesian game-enhanced federated cloud auction model.

There are two types of smart contracts involved. The auction smart contract is the main contract to manage the entire auction process (e.g., setting parameters, auction rules, and execution orders). SLA smart contracts are sub-contracts generated by the auction contract to specify the service details between the customer and each provider. Here SLA smart contracts work as the auction

[1] For simplicity, customer, provider, and witness are used in the following text.

agreements in traditional auction activities, and contain the terms and obligations for both sellers and buyers to finish the deal. Smart contracts can be used as audit evidence by both parties, making the service terms immutable, open, and transparent. Besides, there are two types of fees involved in the auction payment. Witness fees are leveraged to encourage potential blockchain users to join the monitoring activity. They also act as deposits to ensure that the customer and providers cannot easily deny their obligations. By contrast, service fees are the costs of providing cloud services, which are generated by the providers' bids and paid by the customer in the form of cryptocurrency. It should be noted that blockchain cannot act as an enforcement device unless the money (i.e., tokens) is transferred to the smart contract in advance. Therefore, these two types of fees require prepayments to enforce the money transfer at the right time.

4.2.2 Model Process

In Figure 4.1, an agent (usually the customer) first deploys the auction smart contract on the blockchain. Before the auction starts, the customer needs to consider how many providers are needed to form the cloud federation. An off-chain federated cloud partition model is therefore used to accomplish this goal (step 1a). After that, the customer can publish the bid invitation on the auction smart contract (step 1b). This invitation contains a detailed description of the service requirement, e.g., provider numbers, virtual machine (VM) specifications, uptime, throughput, and the reserve price. A certain amount of witness fees need to be prepaid at the same time. When providers notice the auction invitation posted by the customer, they can register as bidders to participate in the auction. Providers can start bidding when the number of registered providers meets the requirement within a specified time window. Similar to the customer, providers need to prepay part of the witness fee when submitting bids (steps 2a and 2b).[2] When the sum of k (the number of providers required by the customer) bids is less than the customer's reserve price, the requirements from both parties are met. The auction smart contract then judges this auction as a valid auction and selects k providers with the lowest bids as the winning providers (step 3a). Otherwise, the auction is invalid (the customer's reserve price is too low or the providers' bids are too high) and the witness fee prepaid by both parties will be refunded (step 3b).

Once the winning providers are selected, the auction smart contract generates k SLA smart contracts (step 4) and waits for both parties to sign (steps 5a and 5b). Next, the customer needs to prepay the service fee generated by the bidding of providers (step 5c). After all SLA smart contracts are signed, a witness registration window opens and allows normal blockchain users to register as auction witnesses to earn commission fees. SLA smart contracts become active only when

[2] The total amount of the witness fee is shared by both the customer and selected providers.

a sufficient number of witnesses are registered. The providers then start to provide the corresponding services for the customer in accordance with the SLA smart contracts, and the witnesses start to monitor the services (steps 6a, 6b, and 6c). After the federated cloud services are completed, witnesses need to report the monitoring results (steps 7a and 7b). The auction smart contract then calculates and pays the witness fee to witnesses (step 8a) and finalizes the ownership of the service fee based on the reported SLAs violation result; if there is no violation, the service fees are paid to all providers (step 8b). Otherwise, the part of the prepaid service fee will be refunded to the customer if any of the sub-services are in violation (step 8c).

4.3 Key Techniques

In this section, we describe three key techniques in our federated cloud auction model in detail. This model enables the automatic enforcement of the auction, the results of which can convince both auction parties. Specifically, we first model the partition of federated cloud services as a graph partition problem (steps $1a$ in Figure 4.1). Then, the effectiveness and trustworthiness of the auction are guaranteed through two BNEs among providers and witnesses (steps $2a$ and $7a$ in Figure 4.1). Finally, a timed message submission (TMS) algorithm is proposed to protect the auction privacy during the message submission phase. The algorithm also allows the model to satisfy the basic assumption of the static Bayesian game (steps $2b$ and $7b$ in Figure 4.1). The notions and symbols used in this chapter are listed in Table 4.1 for easy reference.

4.3.1 Federated Cloud Partition

We consider a problem scenario where a customer used to use the cloud service from a single provider. Now, this customer wants to switch to a federated cloud solution to improve flexibility and reliability. Before using auction models to select different providers, the customer needs to consider how many providers are needed and how to allocate existing resources to those providers. We assume that the total budget and VM topology are fixed. In this case, the original VM topology can be simulated as an infrastructure graph.

Definition 1. ***Infrastructure Graph.*** *Let $G = (M, E)$ be an undirected weighted infrastructure graph for a specific cloud service, where M is a set of vertices and E is a set of edges.*

In a federated infrastructure graph, $M = \{m_1, m_2, \ldots, m_M\}$ denotes a set of cloud VMs. Each VM m_i is assigned to a value $p(m_i)$ that indicates the unit price of this VM regarding usage time. E is a set of edges and each $e = \{i, j\} \in E$ represents the network communication cost between m_i and m_j.

Table 4.1: The list of symbols and notations used in this chapter.

Symbol/Notation	**Description**
$G = (M, E)$	An undirected weighted infrastructure graph for a specific cloud service.
$G' = G_1 \cup G_2 \cup \cdots \cup G_k$	A partitioned infrastructure graph such that k blocks are disjoint and have (nearly) balanced $VMSize$ and $Budget$.
$M = \{m_1, m_2, \ldots, m_M\}$	A set of cloud VMs and each $m_i \in M$ represents a VM.
$p(m_i)$	The time unit service price of VM m_{i}.
E	A set of edges and each $e = \{i, j\} \in E$ represents the network communication cost between m_i and m_j.
J	A vector of resource attributes for a cloud VM.
$\mathbf{r} = \left(r_1, r_2, \ldots, r_{\lvert J \rvert}\right)$	The resource capacity of the VM is a vector, where r_h is a value of the h-th resource attribute in J.
$c_{uv} \in \{0, 1\}$	The binary decision variable for edges of the G, which is 1 if $e = \{u, v\}$ is a cut edge, otherwise it is 0.
$x_{v,k} \in \{0, 1\}$	The binary decision variable for vertices of the G, which is 1 if v is in block k, otherwise it is 0.
$VMSize$	The VM size requirement of each partition.
$Budget$	The budget requirement of each partition.
b_{i}	The bidding function of provider i is $b_{\mathrm{i}} \geq 0$, which is monotonically increasing and differentiable.
v_{i}	The expected value of provider i for the service to be auctioned.
$b_{\mathrm{i}}^{*}\left(v_{\mathrm{i}}\right)$	The BNE of the provider i in the BBG.
BBG	An n-player static Bayesian game of incomplete information for bidding on the federated cloud services.
$N = \{1, 2, \ldots, n\}$	A set of players. Each player is a provider who can offer bids for the cloud service that the customer needs.
$T = \left[v^{\min}, v^{\max}\right]$	The continuous type space of different providers. v_{i} is independently and identically distributed in this interval.
$A = [0, \infty)$	The non-negative continuous action space (bid) of each provider.
p	A common belief of all providers that v_{i} is independently and identically distributed in $\left[v^{\min}, v^{\max}\right]$.
$u = (u_1, u_2, \ldots, u_{\mathrm{n}})$	A set of payoff functions for service providers, where $u_{\mathrm{i}} : A \times T \to R$.
$\mathrm{C_P}$	A provider needs to pay a blockchain transaction fee $\mathrm{C_P}$ to submit new blockchain transactions.
$s = (s_1, s_2, \ldots, s_{\mathrm{n}})$	The strategy profile of the Bayesian game.
WBG	An m-player static Bayesian game of incomplete information for monitoring the service j.
$M = \{1, 2, \ldots, m\}$	A set of players. Each player is a witness and all the witnesses form the witness committee for service j.
$WT = \{H, D, R\}$	The type space of witness members.
$WA = [0, 1]$	The continuous action space for witnesses to monitor the service.
P	A common belief of all witness members that there are three types of witnesses, and the proportions are p_{H}, p_{D}, and p_{R}.
$U = (U_1, U_2, \ldots, U_{\mathrm{m}})$	A set of payoff functions for witnesses, where $U_{\mathrm{i}} : WT \times WA \to R$.
$w_{\mathrm{i}}^{\mathrm{j}}$	The monitoring report given by witness i for the SLA j.
$w_{\mathrm{R}}, w_{\mathrm{H}}, w_{\mathrm{H}}$	The monitoring results of three types of witnesses.
$SLA_{\mathrm{violate}}^{\mathrm{j}}$	The SLA violation indicator for service j.
$\varphi\left(w_{\mathrm{i}}^{\mathrm{j}}\right)$	The penalty function of witness i for monitoring service j.
ε	The intensity factor of the penalty function $\varphi\left(w_{\mathrm{i}}^{\mathrm{j}}\right)$.
$f\left(w_{\mathrm{i}}^{\mathrm{j}}\right)$ and $g\left(w_{\mathrm{i}}^{\mathrm{j}}\right)$	The intrinsic psychological cost functions for honest and dishonest witnesses.
h and l	The preference intensity coefficients for honest and dishonest witnesses.
$\mathrm{C_W}$	A witness needs to pay a blockchain transaction fee $\mathrm{C_W}$ to submit new blockchain transactions.
$F_{\mathrm{witness_i}}^{\mathrm{j}}$	The witness fee of witness i for monitoring service j.
α	The growth factor of the witness i's blockchain transaction fee.

Definition 2. ***Resource Capacity.*** *Let J denote a vector of resource attributes for a cloud VM. Thus, the resource capacity of this VM is a vector $\mathbf{r} = \left(r_1, r_2, \ldots, r_{|J|}\right)$, where r_h is the value of the h-th resource attribute in J.*

A cloud VM can have several resource attributes to determine its pricing, e.g., vCPU, memory, instance storage, and network bandwidth. Suppose there are two VMs with similar functions. Their resource capacities are (2, 500, 100) and (1, 1000, 200), and the three values represent vCPU, memory, and storage, respectively. In this case, the first VM has more vCPU processors, whereas the second VM has higher memory and storage attributes. It is difficult to compare which one is more valuable. In order to evaluate the value of VMs based on their available resources, we use the Simple Additive Weighting (SAW) [5] method to convert the different resource attributes into a single value.

First, the values of different resource attributes can be normalized to eliminate incompatibility. Let the resource capacity of m_i be $\mathbf{r}_i = \left(r_{i,1}, r_{i,2}, \ldots, r_{i,|J|}\right)$, then any $r_{i,h} > 0$ can be normalized by the following function:

$$\beta\left(r_{i,h}\right) = \begin{cases} \frac{r_{i,h} - r_{i,h}^{\min}}{r_{i,h}^{\max} - r_{i,h}^{\min}} & \text{if } r_{i,h}^{\max} \neq r_{i,h}^{\min} \\ 1 & \text{if } r_{i,h}^{\max} = r_{i,h}^{\min} \end{cases} \tag{4.1}$$

Here $r_{i,h}^{\max}$ and $r_{i,h}^{\min}$ are the maximum and minimum values of the h-th resource attribute on m_i. Then, the time unit price of m_i can be converted into a single value, forming the attribute of each node:

$$p(m_i) = \sum_{h=1}^{|J|} \left(\omega_h \times \beta\left(r_{i,h}\right)\right) \tag{4.2}$$

Here $\omega_h \in (0, 1)$ is the price coefficient that determines the unit price of the h-th resource attribute. In this way, Equation (4.2) can represent the total value of the VM with different resource attributes. It should be noted that ω_h could differ in different providers. However, as a partition user (i.e., the customer), it is impossible to know this value for specific providers until the auction is completed (because the providers are not yet selected). Here, the cloud partition model aims to estimate how many providers are most beneficial based on the existing market pattern and total budget. Therefore, the user can use an expected value of the possible price coefficients among all candidate cloud providers.

We now consider partitioning the *Infrastructure Graph* into different blocks with a mixed-integer linear programming (MILP) model. A balanced partition target for both VM size and budgets is set so that all the sub-blocks can be regarded as similar auction objects. At the same time, the communication costs across different providers should be minimized since these costs are usually considered the most expensive ones [91].

Definition 3. ***K-Partitioned Infrastructure Graph.*** *Let $G' = G_1 \cup G_2 \cup \cdots \cup G_k$ denote a partitioned infrastructure graph such that k blocks are disjoint and have (nearly) balanced VM size and budget. Meanwhile, the total weight of cut edges[3] is minimized.*

In order to get G', we first introduce binary decision variables for edges and vertices of the G. More precisely, for each edge $e = \{u, v\} \in E$, we introduce the variable $c_{uv} \in \{0, 1\}$, which is 1 if e is a cut edge, otherwise it is 0. In addition, for each $v \in V$ and block k, we introduce the variable $x_{v,k} \in \{0, 1\}$, which is 1 if v is in block k, otherwise it is 0. There are upper and lower bounds on VM size and budget. Users can adjust the imbalance rate of VM size and budget for different partitions based on $VMSize_{max}$, $VMSize_{min}$, $Budget_{max}$, and $Budget_{min}$ parameters. Finally, $t(v)$ is the service time of VM v. We assume that the budget is proportional to the service time $t(v)$ and the time unit price $p(v)$. Thus, the objective of an infrastructure graph partition model can be described as:

$$\text{Minimize} \sum_{\{u,v\} \in E} c_{uv} \cdot e(\{u, v\}) \tag{4.3}$$

$$\text{Subject to:} \forall \{u, v\} \in E, \forall k : c_{uv} \geq x_{u,k} - x_{v,k} \tag{4.4}$$

$$\forall \{u, v\} \in E, \forall k : c_{uv} \geq x_{v,k} - x_{u,k} \tag{4.5}$$

$$\forall k : \sum_{v \in V} x_{v,k} \leq VMSize_{max} \tag{4.6}$$

$$\forall k : \sum_{v \in V} x_{v,k} \geq VMSize_{min} \tag{4.7}$$

$$\forall k : \sum_{v \in V} x_{v,k} p(v) t(v) \leq Budget_{max} \tag{4.8}$$

$$\forall k : \sum_{v \in V} x_{v,k} p(v) t(v) \geq Budget_{min} \tag{4.9}$$

$$\forall v \in V : \sum_{k} x_{v,k} = 1 \tag{4.10}$$

Objective Equation (4.3) expresses the goal of this model. Constraints Equation (4.4) & Equation (4.5) ensure that c_{uv} satisfies the basic assumption of a cut edge. Constraints Equation (4.6) & Equation (4.7) guarantee that partitioned blocks are balanced regarding the VM size. Constraints Equation (4.8) & Equation (4.9) make sure the block budget does not exceed the upper/lower bounds. Finally, constraint Equation (4.10) sets that each node can only be set to one block.

[3] A cut edge is an edge that connects different partitions of the graph. Therefore, cut edges can be an approximation of the cross-cloud communication cost incurred by the partitioning.

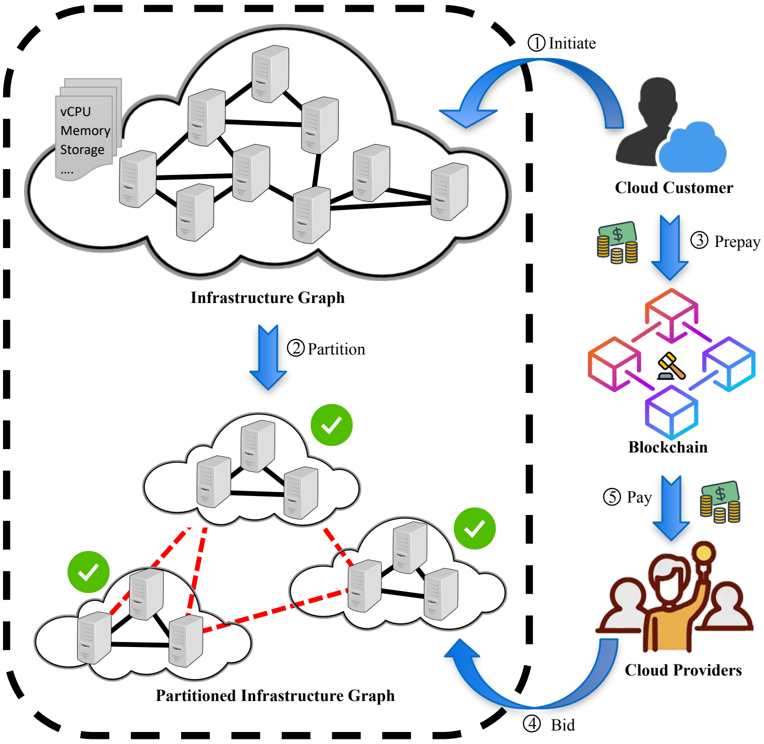


Figure 4.2: An illustration of the federated cloud partition and auction.

An example of a federated cloud partition is illustrated in Figure 4.2. The original infrastructure graph with nine VMs is partitioned into three balanced blocks, and each block has the same VM size and budget boundary (steps 1 and 2). Meanwhile, the total communication cost across different providers (red dotted lines) is minimized. After the partition, the customer should know how many providers are needed to form the federated cloud services. The customer can then post an auction request on the blockchain. After the winning providers are selected through bids (step 4), the customer needs to prepay the service fee using cryptocurrencies (step 3). This fee will be automatically enforced at the end of the auction (step 5). It should be noted that here we consider a balanced infrastructure graph partition. In fact, our auction model is still applicable when dealing with an unbalanced partition scenario. The difference is that in an unbalanced partition, the auction needs to be completed in each sub-block independently since the requirements of blocks are different.

4.3.2 Bayesian Game-Based Auction Enhancement

Bayesian games are often used to model economic situations where players may not know various features of the environment. In this section, two Bayesian games are leveraged to model incomplete information sharing among different cloud providers and auction witnesses.

Bidding Bayesian Game

Some basic assumptions for the bidding Bayesian game are described as follows. There are n providers simultaneously bidding on the cloud services. Since they do not know the types of other players, it is impossible to know the true bids of others. The bidding function of provider i is $b_{\mathrm{i}} \geq 0$, which is monotonically increasing and differentiable. Moreover, v_{i} is the expected value of provider i for the service to be auctioned. It is the private information of provider i and can be regarded as the provider's type. We further define v_{i} is independently and identically distributed[4] on $\left[v^{\min}, v^{\max}\right]$, the cumulative density function is F, and its probability density function is f. Thus, we can formulate this bidding problem as a static Bayesian game of incomplete information.

Definition 4. *Let* $\mathrm{BBG} = (\mathrm{N}, \mathrm{T}, \mathrm{A}, \mathrm{p}, \mathrm{u})$ *denote an n-player static Bayesian game of incomplete information for bidding on the federated cloud services, where:*

- $N = \{1, 2, \ldots, n\}$ *is a set of players. Each player is a provider who can offer bids for the cloud service that the customer needs.*
- $T = \left[v^{\min}, v^{\max}\right]$ *is the continuous type space of different providers. Providers only observe their own types.*
- $A = [0, \infty)$ *is the non-negative continuous action (i.e., bid) space of each provider.*
- p *is a common belief of all providers that* v_{i} *is independently and identically distributed on* $\left[v^{\min}, v^{\max}\right]$ *with cumulative density function* F.
- $u = (u_1, u_2, \ldots, u_{\mathrm{n}})$ *is a set of payoff functions for service providers, where* $u_{\mathrm{i}} : A \times T \to R$.

We propose that in our BBG model, the providers who submit the k-lowest bids win this auction. The rewards they can get are their own bidding prices, and their utilities are the differences between the bids and the internal expected values. In practice, a provider usually does not submit the bid as the expected value v_{i}. Instead, a higher bid is often submitted to maximize profits. At the same time, providers must consider submitting bids lower than other $n-k$ providers' to ensure they will be selected. A provider also needs to pay a blockchain transaction fee $\mathrm{C_P}$ to submit new blockchain transactions.[5] Thus the utility function u_{i} of provider i can be described as:

[4] In practice, the distribution of v_{i} for different providers may be different. This brings system noises and increases the difficulty of estimating the provider's bids. Therefore, we assume that bidders are symmetric in our model, which is the same assumption as in most auction studies.

[5] We define $\mathrm{C_P}$ as a constant since the transaction fees are similar among different providers. The same setting is applied to $\mathrm{C_W}$.

$$u_{\mathrm{i}}\left(b_{\mathrm{i}}, v_{\mathrm{i}}\right)=\begin{cases} b_{\mathrm{i}}-v_{\mathrm{i}}-\mathrm{C_P} & \text{if } \sum_j \mathbb{1}\left\{b_{\mathrm{i}}<b_{\mathrm{j}}, \forall j \neq i\right\}>n-k \\ -\mathrm{C_P} & \text{otherwise} \end{cases} \tag{4.11}$$

Definition 5. *Let the strategy of player i be function $s_i: T \rightarrow A$, thus the strategy profile $s=\left(s_1, s_2, \ldots, s_{\mathrm{n}}\right)$ is a BNE if for any $i \in N, s_i$ assigns an optimal action that maximizes player i's expected payoff.*

Theorem 1. *If the bid of each provider satisfies the following function, the strategy profile $s=\left(b_1^*\left(v_1\right), b_2^*\left(v_2\right), \cdots, b_{\mathrm{n}}^*\left(v_{\mathrm{n}}\right)\right)$ is a unique BNE for the BBG.*

$$b_{\mathrm{i}}^*\left(v_{\mathrm{i}}\right)=v_{\mathrm{i}}+\frac{\int_{v_{\mathrm{i}}}^{v^{\max}}(1-F(v))^{\mathrm{n-k}} d v}{\left[1-F\left(v_{\mathrm{i}}\right)\right]^{\mathrm{n-k}}} \tag{4.12}$$

Proof. Assume the BNE of the BBG can be expressed as $b_{\mathrm{i}}^*\left(v_{\mathrm{i}}\right)=b\left(v_{\mathrm{i}}\right)$ for any $i \in N$, where b is an increasing and differentiable function. A specific provider i must submit a bid which is lower than other $n-k$ players. Thus, the expected utility of provider i can be described as:

$$\mathbb{E}\left[u_{\mathrm{i}}\left(b_{\mathrm{i}}, v_{\mathrm{i}}\right)\right] \tag{4.13}$$

$$=\left(b_{\mathrm{i}}-v_{\mathrm{i}}-\mathrm{C_P}\right) \cdot \operatorname{Pr}\left\{\sum_j \mathbb{1}\left\{b_{\mathrm{i}}<b_{\mathrm{j}}^*\left(v_{\mathrm{j}}\right)=b\left(v_{\mathrm{j}}\right), \forall j \neq i\right\}>n-k\right\}+$$

$$\left(-\mathrm{C_P}\right) \cdot\left(1-\operatorname{Pr}\left\{\sum_j \mathbb{1}\left\{b_{\mathrm{i}}<b_{\mathrm{j}}^*\left(v_{\mathrm{j}}\right)=b\left(v_{\mathrm{j}}\right), \forall j \neq i\right\}>n-k\right\}\right) \tag{4.14}$$

$$=\left(b_{\mathrm{i}}-v_{\mathrm{i}}\right) \cdot \operatorname{Pr}\left\{\sum_j \mathbb{1}\left\{v_{\mathrm{j}}>b^{-1}\left(b_{\mathrm{i}}\right), \forall j \neq i\right\}>n-k\right\}-\mathrm{C_P} \tag{4.15}$$

$$=\left(b_{\mathrm{i}}-v_{\mathrm{i}}\right) \cdot\left[1-F\left(b^{-1}\left(b_{\mathrm{i}}\right)\right)\right]^{\mathrm{n-k}}-\mathrm{C_P} \tag{4.16}$$

Therefore, the objective is to obtain the expression of b_{i} that maximizes the expected utility. Using the first order condition, we get:

$$\begin{aligned}\left(v_{\mathrm{i}}-b_{\mathrm{i}}\right) \cdot f\left(b^{-1}\left(b_{\mathrm{i}}\right)\right) \cdot(n-k)\left[1-F\left(b^{-1}\left(b_{\mathrm{i}}\right)\right)\right]^{\mathrm{n-k-1}} \cdot\left[b^{-1}\left(b_{\mathrm{i}}\right)\right]^{\prime} \\ +\left[1-F\left(b^{-1}\left(b_{\mathrm{i}}\right)\right)\right]^{\mathrm{n-k}}=0\end{aligned} \tag{4.17}$$

Bringing in the equilibrium point of provider i, we get $b_{\mathrm{i}}=b_{\mathrm{i}}^*\left(v_{\mathrm{i}}\right)$. Since $b_{\mathrm{i}}^*\left(v_{\mathrm{i}}\right)=b\left(v_{\mathrm{i}}\right)$, we can replace b_{i} with $b\left(v_{\mathrm{i}}\right)$ in the above equation and get:

$$\begin{aligned}\left(v_{\mathrm{i}}-b\left(v_{\mathrm{i}}\right)\right) \cdot f\left(v_{\mathrm{i}}\right) \cdot(n-k) \cdot\left[1-F\left(v_{\mathrm{i}}\right)\right]^{\mathrm{n-k-1}} / b^{\prime}\left(v_{\mathrm{i}}\right)+ \\ \left[1-F\left(v_{\mathrm{i}}\right)\right]^{\mathrm{n-k}}=0\end{aligned} \tag{4.18}$$

This equation can be rearranged into:

$$\frac{d}{dv_{\mathrm{i}}}b\left(v_{\mathrm{i}}\right)\cdot\left[1-F\left(v_{\mathrm{i}}\right)\right]^{\mathrm{n-k}}=-v_{\mathrm{i}}\cdot f\left(v_{\mathrm{i}}\right)\cdot\left(n-k\right)\cdot\left[1-F\left(v_{\mathrm{i}}\right)\right]^{\mathrm{n-k-1}} \tag{4.19}$$

By integrating both sides of the equation from v_{i} to v^{max}, we get:

$$b\left(v_i\right)=\frac{\int_{v^{max}}^{v_i}v_i\cdot\left[\left(1-F\left(v_i\right)\right)^{n-k}\right]'dv_i}{\left[1-F\left(v_i\right)\right]^{n-k}} \tag{4.20}$$

By simplifying the above equation, the proof of Theorem 1 is finished. □

From Equation (4.12) we know that when n and k are fixed, $b_{\mathrm{i}}^{*}\left(v_{\mathrm{i}}\right)$ depends on v_{i}, v^{max}, and $F\left(v_{\mathrm{i}}\right)$. Since we assume that all the providers are rational, they will bid according to the BNE points of Equation (4.12) to maximize their utility. Consequently, our auction model receives a set of bidding prices where all bidders' utilities are maximized. The most suitable providers (with the k lowest v_{i}) are selected and the effectiveness of the auction is guaranteed.

Next, we discuss the case of different distributions of v_{i}, which is often assumed to be uniform, normal, or log-normal distribution in related research [217]. In order to show the explicit equation, we suppose all v_{i} follows a uniform distribution in $\left[v^{\mathrm{min}},v^{\mathrm{max}}\right]$. Using the cumulative density function of uniform distribution $F(v_{\mathrm{i}})=\frac{v_{\mathrm{i}}-v^{\mathrm{min}}}{v^{\mathrm{max}}-v^{\mathrm{min}}}$, the Equation (4.12) can be further simplified as:

$$b_{\mathrm{i}}^{*}\left(v_{\mathrm{i}}\right)=\frac{n-k}{n-k+1}v_{\mathrm{i}}+\frac{1}{n-k+1}v^{\mathrm{max}} \tag{4.21}$$

The above equation is obtained when the number of target providers k is known before different providers start to bid. In this case, the bid of provider i only needs to be lower than any other $n-k$ providers' to ensure the successful bidding. By contrast, in a first-price auction model where only one provider (with the lowest bid) can be selected, provider i's bid must be lower than all other $n-1$ providers' bids. The customer can also select k providers by performing a k-round first-price auction. At this time, provider i's BNE is:

$$b_{\mathrm{i'}}^{*}\left(v_{\mathrm{i}}\right)=\frac{n-1}{n}v_{\mathrm{i}}+\frac{1}{n}v^{\mathrm{max}} \tag{4.22}$$

In fact, both bidding strategies (Equation (4.21) and Equation (4.22)) can select k providers to form the cloud federation. However, here we only choose Equation (4.21) in our auction model, which is mainly because: 1) the provider's utility in Equation (4.21) is higher than that in Equation (4.22) (as shown in Equation (4.23)), so providers are more willing to participate; and 2) Equation (4.22) requires multiple loops of the auction process, which is time-consuming and expensive to execute on the blockchain.

$$b_{\mathrm{i}}^{*}\left(v_{\mathrm{i}}\right)-b_{\mathrm{i'}}^{*}\left(v_{\mathrm{i}}\right)=\frac{k-1}{n(n-k+1)}\cdot\left(v^{\mathrm{max}}-v_{\mathrm{i}}\right)\geq0 \tag{4.23}$$

Witness Bayesian Game

Once the customer and federated providers reach an agreement on the auction detail, they start to follow and execute it. Providers need to provide the federated cloud services according to the requirements in SLAs, and the customer needs to pay the corresponding service fees. However, any individual can violate the previous agreement. The providers may not provide the quality of service (QoS) they promised, and the customer may also refuse to pay service fees using the excuse of service violations. Therefore, a trustworthy witness mechanism is proposed to monitor and control the enforcement of federated SLAs. We assume the witness committee monitors different cloud services independently; their reporting results for one service do not affect the results for others. Thus we can model the monitoring process of the service j as follows.

Definition 6. *let* $\mathrm{WBG} = (\mathrm{M}, \mathrm{WT}, \mathrm{WA}, \mathrm{P}, \mathrm{U})$ *denote an m-player static Bayesian game of incomplete information for monitoring the service* j *, where:*

- $M = \{1, 2, \ldots, m\}$ *is a set of players. Each player is a witness and they form the witness committee for service* j*.*
- $WT = \{H, D, R\}$ *is the type space of witness members, where* H*,* D*, and* R *represent honest, dishonest, and rational witness, respectively. Witnesses only observe their own types.*
- $WA = [0, 1]$ *is the continuous action space for witnesses to monitor the service. Specifically, we denote* $w_{\mathrm{i}}^{\mathrm{j}}$ *as the monitoring result given by witness* i *for the SLA* j*.*
- P *is a common belief of all witness members that there are three types of witnesses in total, and the proportions are* p_{H}*,* p_{D}*, and* p_{R}*, respectively.*
- $U = (U_1, U_2, \ldots, U_m)$ *is a set of utility functions for witnesses, where* $U_{\mathrm{i}} : WT \times WA \to R$ *is the payoff function determining the rewards of witness* i*.*

Definition 7. *Based on the monitoring result of service* j*, the SLA violation is confirmed only when the majority of witnesses report violations. Otherwise, it is treated as no violation happens. Since* $w_{\mathrm{i}}^{\mathrm{j}} \in [0, 1]$*, the witness* i *reports the violation of SLA* j *when* $w_{\mathrm{i}}^{\mathrm{j}} > 1/2$*.*

$$SLA_{\mathrm{violate}}^{\mathrm{j}} = \begin{cases} 0 & \text{if } \sum_{i} \mathbb{1}\left\{0 \leq w_{\mathrm{i}}^{\mathrm{j}} \leq \frac{1}{2}\right\} > \frac{m}{2} \\ 1 & \text{if } \sum_{i} \mathbb{1}\left\{\frac{1}{2} < w_{\mathrm{i}}^{\mathrm{j}} \leq 1\right\} > \frac{m}{2} \end{cases} \tag{4.24}$$

Here $SLA_{\mathrm{violate}}^{\mathrm{j}} = \{0, 1\}$ is used to denote the finial result of whether SLA j is violated; $SLA_{\mathrm{violate}}^{\mathrm{j}} = 1$ means SLA j is violated while $SLA_{\mathrm{violate}}^{\mathrm{j}} = 0$ means not. The above definition shows that when most witnesses report violations, the model determines that the SLA j is violated. Next, we further design that witnesses should be penalized if their reports fail to match the reports of others. Thus the penalty function of witness i can be described as $\varphi\left(w_{\mathrm{i}}^{\mathrm{j}}\right)$.

$$\varphi\left(w_{\mathrm{i}}^{\mathrm{j}}\right) = \frac{\varepsilon}{n-1} \sum_{i \neq k} \left(w_{\mathrm{i}}^{\mathrm{j}} - w_{\mathrm{k}}^{\mathrm{j}}\right)^2 \tag{4.25}$$

Here ε is defined as the intensity factor of the penalty function. Since each type of witness will give out the same report based on their strategy profile, the penalty function $\varphi\left(w_{\mathrm{i}}^{\mathrm{j}}\right)$ can be further described as:

$$\begin{aligned} \varphi\left(w_{\mathrm{i}}^{\mathrm{j}}\right) = \varepsilon\Big[(1 - p_{\mathrm{H}} - p_{\mathrm{D}}) \left(w_{\mathrm{i}}^{\mathrm{j}} - w_{\mathrm{R}}\right)^2 + p_{\mathrm{H}} \left(w_{\mathrm{i}}^{\mathrm{j}} - w_{\mathrm{H}}\right)^2 + \\ p_{\mathrm{D}} \left(w_{\mathrm{i}}^{\mathrm{j}} - w_{\mathrm{D}}\right)^2 \Big] \end{aligned} \tag{4.26}$$

Here w_{R}, w_{H}, and w_{D} are used to denote the monitoring results from three types of witnesses. We further assume honest and dishonest witnesses have a psychological cost to tell a lie/truth, while rational witnesses have no psychological burden. The quadratic functions $f\left(w_{\mathrm{i}}^{\mathrm{j}}\right)$ and $g\left(w_{\mathrm{i}}^{\mathrm{j}}\right)$ can well represent the psychological burden of two types of witnesses.

$$f\left(w_{\mathrm{i}}^{\mathrm{j}}\right) = \begin{cases} h \cdot \left(1 - w_{\mathrm{i}}^{\mathrm{j}}\right)^2 & \text{if } SLA_{\mathrm{violate}}^{\mathrm{j}} = 1 \\ h \cdot (w_{\mathrm{i}}^{\mathrm{j}})^2 & \text{if } SLA_{\mathrm{violate}}^{\mathrm{j}} = 0 \end{cases} \tag{4.27}$$

$$g\left(w_{\mathrm{i}}^{\mathrm{j}}\right) = \begin{cases} l \cdot \left(1 - w_{\mathrm{i}}^{\mathrm{j}}\right)^2 & \text{if } SLA_{\mathrm{violate}}^{\mathrm{j}} = 0 \\ l \cdot (w_{\mathrm{i}}^{\mathrm{j}})^2 & \text{if } SLA_{\mathrm{violate}}^{\mathrm{j}} = 1 \end{cases} \tag{4.28}$$

Here $f\left(w_{\mathrm{i}}^{\mathrm{j}}\right)$, $g\left(w_{\mathrm{i}}^{\mathrm{j}}\right) : [0,1] \to R_{+}$. h and l are the preference intensity coefficients for honest and dishonest witnesses. We design that honest and dishonest witnesses have inherent psychological costs, which means that when honest witnesses tell lies or dishonest witnesses tell the truth, their psychological burden will increase. Especially, we set $h = l = 1$ in the following text to simplify the calculation. Thus, we can describe the utility function of three types of witnesses as follows, where $\mathrm{C_W}$ is the blockchain transaction fee and $F_{\mathrm{witness_i}}^{\mathrm{j}}$ is the witness fee of witness i for service j. We specify that the witness fee is large enough so that the witness's utility is always positive ($U_{\mathrm{i}} > 0$). In this way, witnesses always have an incentive to participate in our model and receive a reward.

$$U_{\mathrm{i}}\left(w_{\mathrm{i}}^{\mathrm{j}}, SLA_{\mathrm{violate}}^{\mathrm{j}}, R\right) = F_{\mathrm{witness_i}}^{\mathrm{j}} - \varphi\left(w_{\mathrm{i}}^{\mathrm{j}}\right) - \mathrm{C_W} \tag{4.29}$$

$$U_{\mathrm{i}}\left(w_{\mathrm{i}}^{\mathrm{j}}, SLA_{\mathrm{violate}}^{\mathrm{j}}, H\right) = F_{\mathrm{witness_i}}^{\mathrm{j}} - \varphi\left(w_{\mathrm{i}}^{\mathrm{j}}\right) - f\left(w_{\mathrm{i}}^{\mathrm{j}}\right) - \mathrm{C_W} \tag{4.30}$$

$$U_{\mathrm{i}}\left(w_{\mathrm{i}}^{\mathrm{j}}, SLA_{\mathrm{violate}}^{\mathrm{j}}, D\right) = F_{\mathrm{witness_i}}^{\mathrm{j}} - \varphi\left(w_{\mathrm{i}}^{\mathrm{j}}\right) - g\left(w_{\mathrm{i}}^{\mathrm{j}}\right) - \mathrm{C_W} \tag{4.31}$$

Theorem 2. *In a WBG, when the monitoring result of each type of witness satisfies the following equations, the strategy profile* $s = (w^*_{\mathrm{R}}, w^*_{\mathrm{H}}, w^*_{\mathrm{D}})$ *constitutes a unique BNE.*

$$w^*_{\mathrm{R}} = \begin{cases} \frac{p_{\mathrm{H}}}{p_{\mathrm{H}}+p_{\mathrm{D}}} & \text{if } SLA^{\mathrm{j}}_{\mathrm{violate}} = 1 \\ \frac{p_{\mathrm{D}}}{p_{\mathrm{H}}+p_{\mathrm{D}}} & \text{if } SLA^{\mathrm{j}}_{\mathrm{violate}} = 0 \end{cases} \tag{4.32}$$

$$w^*_{\mathrm{H}} = \begin{cases} 1 - \frac{\varepsilon \cdot p_{\mathrm{D}}}{(\varepsilon+1)(p_{\mathrm{H}}+p_{\mathrm{D}})} & \text{if } SLA^{\mathrm{j}}_{\mathrm{violate}} = 1 \\ \frac{\varepsilon \cdot p_{\mathrm{D}}}{(\varepsilon+1)(p_{\mathrm{H}}+p_{\mathrm{D}})} & \text{if } SLA^{\mathrm{j}}_{\mathrm{violate}} = 0 \end{cases} \tag{4.33}$$

$$w^*_{\mathrm{D}} = \begin{cases} 1 - \frac{p_{\mathrm{H}}+(\varepsilon+1) \cdot p_{\mathrm{D}}}{(\varepsilon+1)(p_{\mathrm{H}}+p_{\mathrm{D}})} & \text{if } SLA^{\mathrm{j}}_{\mathrm{violate}} = 1 \\ \frac{p_{\mathrm{H}}+(\varepsilon+1) \cdot p_{\mathrm{D}}}{(\varepsilon+1)(p_{\mathrm{H}}+p_{\mathrm{D}})} & \text{if } SLA^{\mathrm{j}}_{\mathrm{violate}} = 0 \end{cases} \tag{4.34}$$

Proof. Since the situation about SLA violation is symmetric, we only consider $SLA^{\mathrm{j}}_{\mathrm{violate}} = 1$ to prove Theorem 2. Here the iterated elimination of strictly dominated strategies (IESDS) method [189] is leveraged to narrow down and solve the BNE of the WBG. Specifically, with IESDS the original game can be divided into an n-round game to remove dominated strategies. In each round, witness i wants to maximize the utility by reporting different $w^{\mathrm{j}}_{\mathrm{i}}$. The monitoring result of round $t+1$ is determined by round t when the payoff is maximized.

$$w_{\mathrm{R}}(t+1) = \arg \max_{w^{\mathrm{j}}_{\mathrm{i}} \in w_{\mathrm{R}}(t)} \left[F^{\mathrm{i}}_{\mathrm{witness}} - \varphi\left(w^{\mathrm{j}}_{\mathrm{i}}\right) - \mathrm{C_W} \right] \tag{4.35}$$

$$w_{\mathrm{H}}(t+1) = \arg \max_{w^{\mathrm{j}}_{\mathrm{i}} \in w_{\mathrm{H}}(t)} \left[F^{\mathrm{i}}_{\mathrm{witness}} - \varphi\left(w^{\mathrm{j}}_{\mathrm{i}}\right) - f\left(w^{\mathrm{j}}_{\mathrm{i}}\right) - \mathrm{C_W} \right] \tag{4.36}$$

$$w_{\mathrm{D}}(t+1) = \arg \max_{w^{\mathrm{j}}_{\mathrm{i}} \in w_{\mathrm{D}}(t)} \left[F^{\mathrm{i}}_{\mathrm{witness}} - \varphi\left(w^{\mathrm{j}}_{\mathrm{i}}\right) - g\left(w^{\mathrm{j}}_{\mathrm{i}}\right) - \mathrm{C_W} \right] \tag{4.37}$$

Specifically, the utility would be maximized when the first-order condition is satisfied. For three types of witnesses, let $w^{\mathrm{j}}_{\mathrm{i}} = w_{\mathrm{R}}$, $w^{\mathrm{j}}_{\mathrm{i}} = w_{\mathrm{H}}$, and $w^{\mathrm{j}}_{\mathrm{i}} = w_{\mathrm{D}}$, respectively. We get:

$$w_{\mathrm{R}}(t+1) = \frac{p_{\mathrm{H}}}{p_{\mathrm{H}} + p_{\mathrm{D}}} w_{\mathrm{H}}(t) + \frac{p_{\mathrm{D}}}{p_{\mathrm{H}} + p_{\mathrm{D}}} w_{\mathrm{D}}(t) \tag{4.38}$$

$$w_{\mathrm{H}}(t+1) = \frac{\varepsilon \cdot (1 - p_{\mathrm{H}} - p_{\mathrm{D}})}{\varepsilon - \varepsilon \cdot p_{\mathrm{H}} + 1} w_{\mathrm{R}}(t) + \frac{\varepsilon \cdot p_{\mathrm{D}} \cdot w_{\mathrm{D}}(t) + 1}{\varepsilon - \varepsilon \cdot p_{\mathrm{H}} + 1} \tag{4.39}$$

$$w_{\mathrm{D}}(t+1) = \frac{\varepsilon \cdot (1 - p_{\mathrm{H}} - p_{\mathrm{D}})}{\varepsilon - \varepsilon \cdot p_{\mathrm{D}} + 1} w_{\mathrm{R}}(t) + \frac{\varepsilon \cdot p_{\mathrm{H}}}{\varepsilon - \varepsilon \cdot p_{\mathrm{D}} + 1} w_{\mathrm{H}}(t) \tag{4.40}$$

From the above combined equations we can see that the maximum/minimum value of w_{R} in round $t+1$ is determined by the maximum/minimum value of w_{H} and w_{D} in round t. The cases of w_{H} and w_{D} are similar. We then use the proof by contradiction to show the uniqueness of the Nash Equilibrium. Assume in the IESDS process, w_{R}^{*}, w_{H}^{*}, w_{D}^{*} finally converge to an interval $[\min(s^{*}), \max(s^{*})]$ separately instead of a single point. Thus, $\min(w_{\mathrm{R}}^{*})$, $\max(w_{\mathrm{R}}^{*})$, $\min(w_{\mathrm{H}}^{*})$, $\max(w_{\mathrm{H}}^{*})$, $\min(w_{\mathrm{D}}^{*})$, and $\max(w_{\mathrm{D}}^{*})$ should all satisfy the combined equations. At the Nash equilibrium state the witnesses should report the same result at round t and $t+1$. However, given the fixed ε, p_{H}, and p_{D}, the above combined equations only have one unique set of solution, as shown in equation Equation (4.41), Equation (4.42), and Equation (4.43).

$$w_{\mathrm{R}}^{*} = \frac{p_{\mathrm{H}}}{p_{\mathrm{H}} + p_{\mathrm{D}}} \tag{4.41}$$

$$w_{\mathrm{H}}^{*} = 1 - \frac{\varepsilon \cdot p_{\mathrm{D}}}{(\varepsilon + 1)\left(p_{\mathrm{H}} + p_{\mathrm{D}}\right)} \tag{4.42}$$

$$w_{\mathrm{D}}^{*} = 1 - \frac{p_{\mathrm{H}} + (\varepsilon + 1)p_{\mathrm{D}}}{(\varepsilon + 1)\left(p_{\mathrm{H}} + p_{\mathrm{D}}\right)} \tag{4.43}$$

This means $w_{\mathrm{R}}^{*} = \min(w_{\mathrm{R}}^{*}) = \max(w_{\mathrm{R}}^{*})$, $w_{\mathrm{H}}^{*} = \min(w_{\mathrm{H}}^{*}) = \max(w_{\mathrm{H}}^{*})$, and $w_{\mathrm{D}}^{*} = \min(w_{\mathrm{D}}^{*}) = \max(w_{\mathrm{D}}^{*})$ in the Nash Equilibrium. The same conclusion can also be obtained in the situation when $SLA_{\text{violate}} = 0$. Therefore, the original assumption (i.e., the strictly dominated strategies converge to an interval) does not hold. Theorem 2 is proved. □

In the above modeling, witness i only monitors the SLA j. When k federated SLAs need to be monitored together, the overall payoff of witness i then changes to Equation (4.44) (with the rational type). Since we assume the witness committee monitors different sub-cloud services independently, the BNE of the monitoring result for a specific provider does not change.

$$U_{\mathrm{i}}\left(w_{\mathrm{i}}, SLA_{\text{violate}}, R\right) = k \cdot F_{\text{witness}_{\mathrm{i}}}^{\mathrm{j}} - \sum_{j \in J} \varphi\left(w_{\mathrm{i}}^{\mathrm{j}}\right) - \alpha \cdot \mathrm{C_W} \tag{4.44}$$

Here α is the growth factor of the witness i's blockchain transaction fee. It is also noted that the witness committee monitors each sub-service separately in the above modeling process. This means that the monitoring result of each witness is a k-dimension vector, and each value of the vector indicates whether a sub-cloud service is violated. By contrast, each witness can also make a general judgment about whether the whole federated cloud service is violated or not. In this case, the monitoring result is a value from 0 to 1 instead of a vector, and the federated cloud services formed by k providers are actually regarded as one service. However, this design may be unfriendly to providers who do not violate the SLA when the whole judgment is a violation.

4.3.3 Timed Message Submission (TMS) Algorithm

In the BBG and WBG of the proposed federated cloud auction model, we assume that both providers and witnesses submit messages (bids and reports) to the blockchain simultaneously. The simultaneity and data privacy during the submission phase are crucial to:

- Satisfy the basic assumption of game theory. The static Bayesian game requires all players to submit messages to the blockchain simultaneously. However, this is very difficult to achieve in reality.
- Protect bid privacy and avoid possible plagiarism among submitters. Since data on the blockchain is public and transparent, if users who submit later can see the predecessors' message, their judgment may be affected.

Algorithm 1 TMS algorithm – Phase 1

```
Input:
Length of the registered witnesses array: $len(RW)$;
Length of the SLA array: $len(SLAs)$;
Sealed message array of witness i: $sealedMessages_{\text{i}}$;
Submission deposit of the witness i: $Deposit_{\text{i}}$.
Output:
Sealed message map for all witnesses: $sealedMap$;
// Phase 1: sealed message array submission
function SUBMITMESSAGES($sealedMessages_{\text{i}}$)
    require ($now < submitEnd$)
    require ($RW[address_{\text{i}}].registered == true$)
    require ($msg.value >= Deposit_{\text{i}}$)
    if $len(sealedMessages_{\text{i}}) == len(SLAs)$ then
        $submissionDeposit[address_{\text{i}}] = msg.value$
        $sealedMap[address_{\text{i}}] \leftarrow sealedMessages_{\text{i}}$
        $witnessNum++$
        if $witnessNum == len(RW)$ then
            return $sealedMap$
        end if
    end if
end function
```

In this context, we propose the TMS algorithm to handle the above-mentioned challenges. Our algorithm consists of two phases: 1) sealed message array submission; and 2) message array reveal and deposit refund. Correspondingly, there are two function interfaces named “submitMessages” and “RevealMessages” in the auction smart contract. The proposed algorithm can be leveraged by both providers and witnesses. Here we use the witness's case as an example to explain the algorithm details.

Algorithm 1 and Algorithm 2 demonstrate the two phases of the TMS algorithm. In the first phase, only registered witnesses can submit the sealed message array. This array contains k values, and each value is a hash of the witness's judgment and the private key. When the current time is in the submission time window and the size of the message array meets the requirement, the witness can submit a deposit and store the sealed message array in $sealedMap$. After all witnesses submit their sealed message arrays, the function outputs a sealed message map, where keys are the addresses of witnesses, and values are the arrays of hash values. Considering the irreversibility of the hash function, it is impossible for witnesses to detect the true messages of others in this phase.

Algorithm 2 TMS algorithm – Phase 2

```
Input:
Message array of witness $i$: $realMessages_{\mathrm{i}}$;
Private key of witness $i$: $witnessKey_{\mathrm{i}}$;
Sealed message map obtained in the previous phase: $sealedMap$;
Output:
Revealed message map for all witnesses: $revealedMap$
// Phase 2: message array reveal and deposit refund
function REVEALMESSAGES($realMessages_{\mathrm{i}}, witnessKey_{\mathrm{i}}$)
    require ($now > submitEnd$ && $now < revealEnd$)
    require ($sealedMap[address_{\mathrm{i}}] \; != \; null$)
    require ($len(realMessages_{\mathrm{i}}) == len(SLAs)$)
    for $j = 0$; $j < len(SLAs)$; $j++$ do
        require ($realMessages_{\mathrm{i}}[j] \in [0, 1]$ )
        if $Hash(realMessages_{\mathrm{i}}[j], witnessKey_{\mathrm{i}}) == sealedMap[address_{\mathrm{i}}][j]$ then
            $SLAsNum++$
        end if
    end for
    if $SLAsNum == len(SLAs)$ then
        $revealedMap[address_{\mathrm{i}}] \leftarrow realMessages_{\mathrm{i}}$
        $address_{\mathrm{i}}.transfer(submissionDeposit[address_{\mathrm{i}}])$
        $submissionDeposit[address_{\mathrm{i}}] = 0$
        $revealNum++$
        if $revealNum == len(RW)$ then
            return $revealedMap$
        end if
    end if
end function
```

Then, the algorithm comes to the second phase. At this time, each witness needs to submit a message array (with true values instead of hashes) and a right witness key to reveal and verify the previously submitted sealed message array. Only witnesses who have submitted the sealed message in the first stage can call this function. In this stage, when the length and range of the real message array entered by the witness meet the requirement, the function checks whether the hash of each message value and witness key is equal to the sealed one submitted in the previous phase. We design that the smart contract can confiscate the deposit as a penalty if a submitter does not disclose the information within a specific time window. In this way, the submitter is forced to open the sealed message in time. When all values are revealed successfully, the real message array of the witness is stored in the smart contract for further processing. Then, when all witnesses are verified successfully, the algorithm ends and the "revealedMap" is exported. With these two steps, data privacy and simultaneity submission are guaranteed within a time window. The algorithm further discourages irresponsible submitters from causing losses to others by not revealing the message in time.

4.4 Evaluation and Implementation

In this section, we design and implement experiments to test the proposed federated cloud auction model. Our evaluation is in the following parts. We first evaluate each of the three key techniques introduced in Section 4.3, namely the cloud partition model, the BNE strategies of two Bayesian games, and the TMS algorithm. Then, the entire smart contract implementation and the cost details are presented.

4.4.1 Cloud Partition Benchmark

To benchmark the proposed federated cloud partition model, we prepared four graph datasets. Table 4.2 shows the statistical information of the selected graphs, in which both synthetic and real-world datasets are used. We first used three types of synthetic workflows provided by the workflow generator in the Pegasus community [20]. These datasets simulate real scientific applications in seismology (CyberShake), biology (Genomes), and astronautics (Montage) fields. The datasets provide information about the application patterns and task dependencies performed on benchmark workstations, and are therefore suitable for testing our graph partition model. We also used a real-world dataset that contains cluster traces from the Alibaba Cluster Trace Program [10]. The trace is sampled from a real production cluster with long-running applications and batch workloads on each machine. The cut edge proportion is used as a metric to evaluate the graph partitioning performance. We use Gurobi 8.1.1 as the MILP solver to find the optimal solution of the partitioned combinations. In addition, a popular graph partitioning tool called METIS is used as the benchmark for comparison. METIS could partition unstructured graphs into user-specified parts using either recursive bisection or k-way partitioning algorithms, both of which produce high-quality partitions [111].

Figure 4.3 consists of 12 plots, where each column represents the result of one dataset. The four plots above show the performance with different partition blocks

Table 4.2: Statistics of the tested graph datasets.

Dataset	Vertices	Edges	Degree	Category
CyberShake	100	192	3.84	Synthetic (Pegasus)
Genomes	100	122	2.44	Synthetic (Pegasus)
Montage	100	234	4.68	Synthetic (Pegasus)
Alibaba	100	178	3.56	Real-world (Clusters trace 2018)

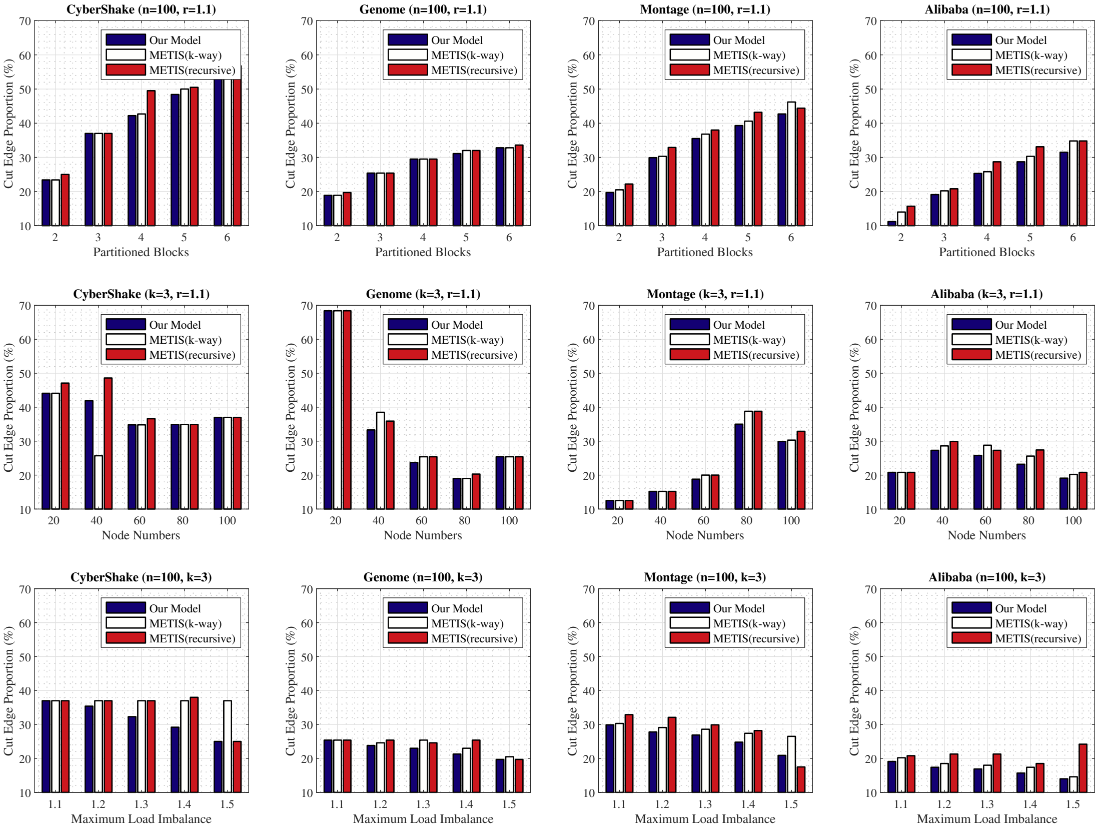


Figure 4.3: Performance analysis of the federated cloud partition model.

when the number of nodes and the load imbalance are fixed (n=100, r=1.1). It can be concluded that the proportion of cut edges tends to increase linearly with the increase in the number of partitioned blocks. Our model performs better in the CyberShake, Montage, and Alibaba datasets than in the Genome case. The middle four plots present the result of increasing the node numbers when the number of partitions and the load imbalance are fixed (k=3, r=1.1). Although an increase in graph nodes can increase the total number of edges, there is no such an increasing trend in the proportion of cut edges. Finally, the four plots below show the performance when the nodes and partition blocks are fixed (n=100, k=3). In general, when relaxing the constraint of the maximum load imbalance, our model obtains better performance to a linear trend. The case of METIS is more complex, and both partitioning methods' performance fluctuates under different configurations. It is worth noting that in the two special cases (i.e., CyberShake (n=40, k=3, r=1.1) and Montage (n=100, k=3, r=1.5)) the METIS method outperforms our model. After analysis, we found that METIS could not obtain feasible solutions in both cases; despite the lower value of the cut edge proportion, the load imbalance actually exceeds the maximum limit and thus leads to partition failures. This also proves the advantage of our model from the side. In summary, the number of nodes and partitions, the maximum load imbalance, and the model solving methods all affect the partition performance. Our model shows different degrees of improvement in different datasets and experimental settings compared to the two METIS approaches. The maximum performance improvement in CyberShake, Genome, Montage, and Alibaba datasets are 12%, 5.2%, 5.6%, and 10.2%, respectively.

Another finding is from Equation (4.12) and Figure 4.3, the fewer partitioned blocks result in lower bids and less cut edges. Therefore, the cloud customer seems to prefer to partition fewer blocks. However, in real life, the customer's decision on partitioning can be very complex and depends on many factors. Choosing fewer partition blocks means putting all their eggs in fewer baskets, which increases the risk of single points of failure. More partitioned blocks may also have advantages, such as better flexibility, reliability, and scalability — which is why federated cloud services are needed. Therefore, customers need to consider the trade-off between partitioning cost and QoS requirements.

Besides, we notice that the execution time is acceptable (within a few minutes) for most of the tested graphs. However, when dealing with large-scale graphs (e.g., more than several hundred nodes or when the graph density is very high), the model may take more than several hours to solve. This is mainly because the proposed MILP model is an NP-hard problem aimed at obtaining an optimal solution, which is a trade-off compared to METIS where fast partitioning results (but not optimal) can be obtained within seconds. The goal of the proposed federated cloud partition model is to find an optimal solution that helps customers to choose the right number of providers. Based on this consideration, we value partition quality as a more important metric than execution time in the current model. We leave the algorithmic optimization of the model execution time to future work.

4.4.2 Bayesian Nash Equilibrium Analysis

In this section, we analyze and validate BNE strategies of the BBG and WBG in our model. Figure 4.4 shows the equilibrium strategies of providers (bidders) when the types and distributions are different. There are three rows of plots, representing three types of bidders, namely $v_{\mathrm{i}} = 0.1$, $v_{\mathrm{i}} = 0.5$, and $v_{\mathrm{i}} = 0.9$. Different bidders will submit bids based on their own expected values. Usually, bidders with higher

v_{i} can offer higher bidding prices. The three columns correspond to three different distributions of V, namely $V \sim \mathcal{U}(0,1)$, $V \sim \mathcal{N}(0.5,1)$, and $V \sim log{-}\mathcal{N}(0.5,1)$. The x-axis of each plot represents the number of bidding providers (n), and the y-axis represents the number of providers to be selected (k). The color of the square is the final bid $b_{\mathrm{i}}^{*}(v_{\mathrm{i}})$ submitted by provider i at the equilibrium point, where the redder the color, the higher the bid. It should be noted that there is no equilibrium when $k > n$.

First, when v_{i} and k are fixed, $b_{\mathrm{i}}^{*}(v_{\mathrm{i}})$ will decrease with the increase of n. This is because the bidding becomes more competitive when n increases, and bidders must submit a lower bid to defeat their competitors. Similarly, when n and v_{i} are fixed, $b_{\mathrm{i}}^{*}(v_{\mathrm{i}})$ will increase as k increases due to the bidding competition is weakened when k becomes larger. Thus, bidders can increase profits by submitting higher bids. When k approaches to n, $b_{\mathrm{i}}^{*}(v_{\mathrm{i}})$ reaches its highest point. Besides, in the BBG bidders needs to detect other bidders' types from the current distribution of V to determine their own bids. For example, when $v_{\mathrm{i}} = 0.1$ for bidder i, the probabilities of the other bidders' v_{j} locating between 0 to 0.1 for uniform, normal, and log-normal distribution are 10%, 3.6%, and 0.25%, respectively. The smaller probability means that potential competitors (people with lower bids than bidder i) are less likely to appear. So bidder i can submit a higher bid to increase profits.

Figure 4.5 consists of 12 plots showing the equilibrium strategies of three witnesses types ($WT_{\mathrm{i}} = R$, $WT_{\mathrm{i}} = H$, and $WT_{\mathrm{i}} = D$). The x-axis of each plot is the proportion of honest witnesses p_{H}, while the y-axis is the proportion of

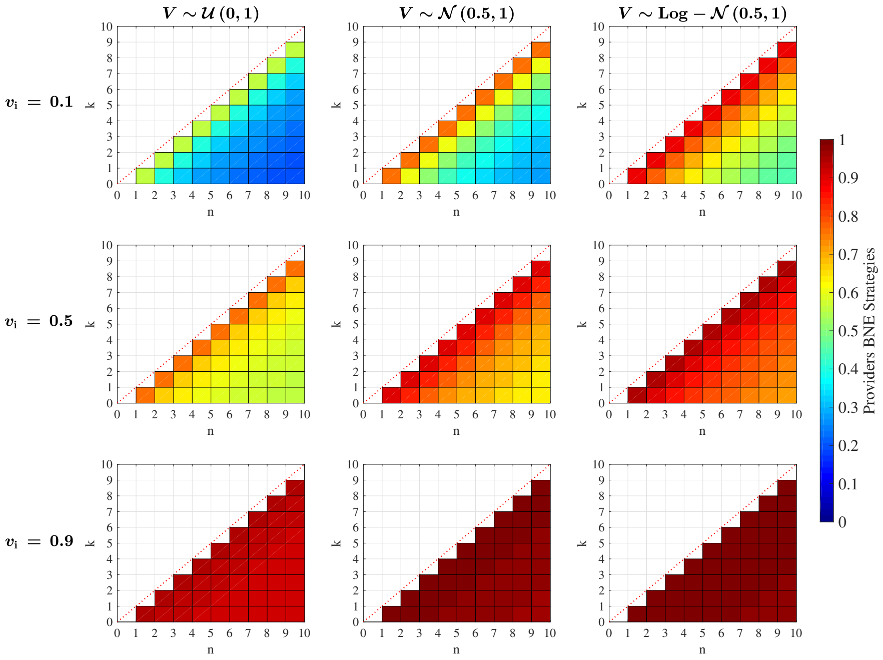


Figure 4.4: Equilibrium strategies of the providers.

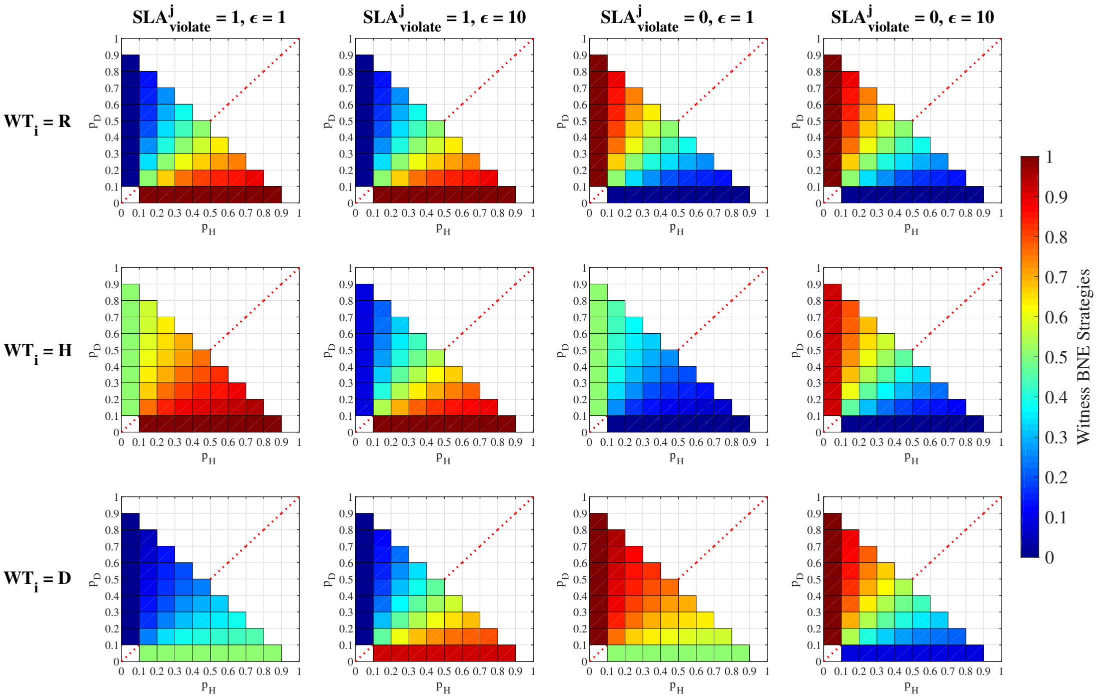


Figure 4.5: Equilibrium strategies of the auction witnesses.

dishonest witnesses p_{D}. The color in the square represents the monitoring result w_{i}^{*} reported by witness i at the equilibrium point. The equilibrium strategy does not exist when $p_{\mathrm{H}} + p_{\mathrm{D}} > 1$. Besides, The first two columns indicate the situation of an SLA violation, while the last two columns indicate there are no violations. We set the penalty function factor $\varepsilon = 1$ in the first and third columns, and $\varepsilon = 10$ in the second and fourth columns. When $SLA^{\mathrm{j}}_{\mathrm{violate}} = 1$, it can be observed that the w_{i}^{*} of rational witnesses are only influenced by the current p_{H} and p_{D}. As p_{H} increases, they tends to report the true result where $SLA^{\mathrm{j}}_{\mathrm{violate}} > 1/2$. Otherwise when p_{D} is larger, they will report $SLA^{\mathrm{j}}_{\mathrm{violate}} < 1/2$. Next, the w_{i}^{*} of honest and dishonest witnesses are affected by both their intrinsic psychological cost and penalty function factor ε. When $\varepsilon = 1$, the penalty function is not enough to restrict their behaviors. Therefore, the honest/dishonest witnesses will follow their nature to tell the truth/lie. However, when ε is 10, witnesses' reports need to be consistent with others' to reduce the huge penalty. In conclusion, when ε is large enough and $p_{\mathrm{H}} > p_{\mathrm{D}}$, all three types of witnesses will report the consistent true SLA violations in the Nash equilibrium in order to improve their utility. The consistency and trustworthiness of monitoring, in this way, are guaranteed.

4.4.3 TMS Algorithm Evaluation

In Section 4.3.3 we proposed that the TMS algorithm can be applied to both providers and witnesses for submitting bids and monitoring results while protecting

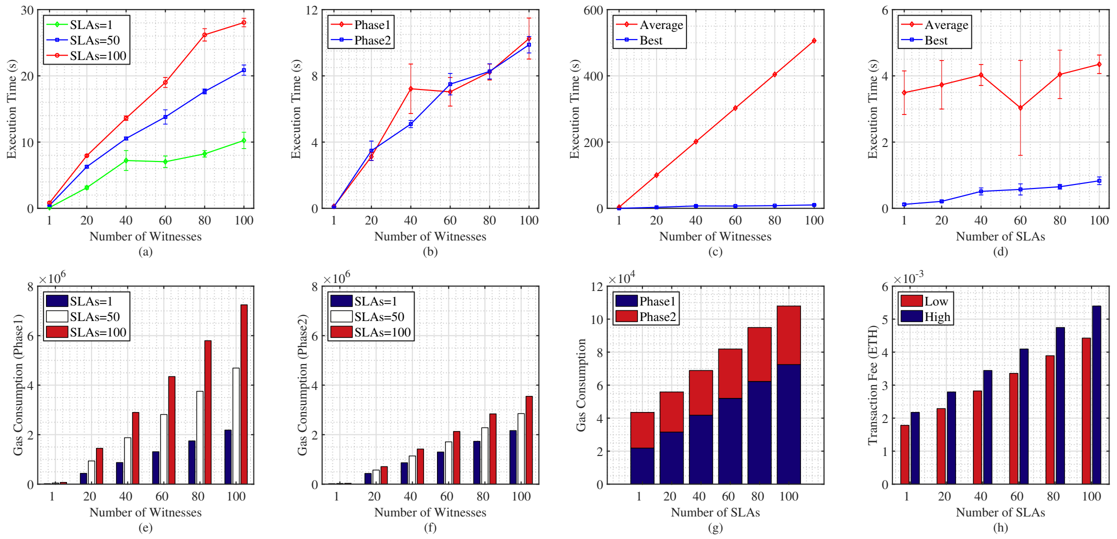


Figure 4.6: Execution time and cost evaluation of the TMS algorithm.

privacy. In this section, we evaluate the performance of the TMS algorithm using the case of witnesses. The on-chain execution overhead, including execution time and cost, is tested in a local Ethereum blockchain. Specifically, time is calculated as the difference between block timestamps [190], and the cost is the gas consumption to perform transactions on the Ethereum blockchain. Different numbers of witnesses, as well as the SLAs they need to monitor, are tested. Two mining network congestion situations of the blockchain are simulated: “Best” means that there are enough miners who will process the transactions in time, while “Average” means that mining is congested and there is a delay in transaction processing.

Figure 4.6 consists of eight plots. In the four plots above, plots 4.6a and 4.6d show that the execution time of the TMS algorithm increases linearly with the growth of witnesses and SLA numbers. Specifically, the algorithm execution time is less affected by the number of SLAs compared to the number of witnesses. When the witness number increases to 100, the execution time increases significantly. In contrast, when the SLA is increased by a factor of 100, the execution time increases only a little. Besides, plot 4.6b shows that the execution time of phases 1 and 2 are similar. Plot 4.6c indicates that the congestion of the blockchain mining network plays a critical role in the algorithm performance; it takes only a few seconds when the network is in the “best” condition but can last for several minutes when the network is congested (i.e., “average” condition).

The lower four plots of Figure 4.6 demonstrate the total cost of the TMS algorithm and the average cost per user. First, plots 4.6e and 4.6f show that the total gas consumption for both phases of the TMS algorithm increases linearly with the number of witnesses and SLAs. By comparing the two graphs, it can be seen that when the number of witnesses is very high, the gas consumption

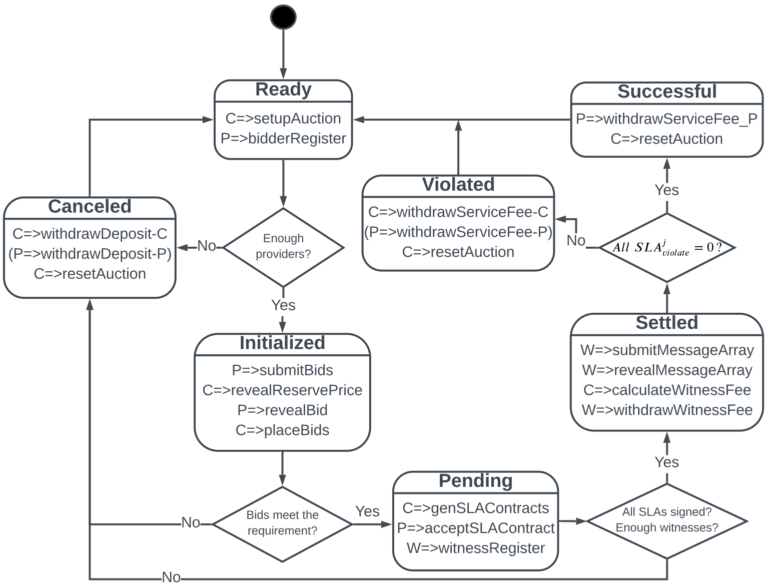


Figure 4.7: State transition diagram of the AWESOME smart contract.

of phase 1 is about twice that of phase 2. When the number of SLAs increases, the total cost of phase 1 increases more significantly. Next, plot 4.6g shows a linear increase in the average cost per user as the number of SLAs changes. When the SLA increases to 100 times, the gas consumption increases only by less than three times. In addition, plot 4.6h shows the predicted transaction costs of the TMS algorithm on the Ethereum main chain for two transaction speed cases.[6] The results show that the transaction fee increases by 21.95% when the requested transaction speed is changed to “high". However, the amount of Ether spent per user remains low, and in the worst case (i.e., SLA number is 100 and transaction speed is high), the algorithm spends less than $3. In summary, we argue that the execution overhead of the TMS algorithm is acceptable for each user compared to its improvement in model trustworthiness.

4.4.4 Smart Contract Implementation and Evaluation

According to the architecture and payoff functions of our federated cloud auction model, we implement a prototype system based on the Ethereum blockchain with Solidity programming language.[7] Figure 4.7 shows the state transitions of the auction smart contract. The rectangles in the figure represent different composite states, where the upper part shows the current state of the auction, and the lower part are the actions that can occur in this state. The C, P, and W next to the actions represent the customer, provider, and witness. We design different

[6] https://etherscan.io/gastracker

[7] https://solidity.readthedocs.io/en/latest/

function interfaces to be initiated by different auction roles, and the initiator is the beneficiary with the greatest benefit at the current stage. The deployer of the smart contract (usually the customer) needs to check the current conditions and determine the confirmation of the state transfer. For money refunding, instead of automatically transferring money from the smart contract to users, we design that users need to withdraw the money by themselves. This is because when transferring money to multiple addresses, an attacker can trap the contract into an unusable state. In contrast, in the “withdrawal” mode, an attacker can only cause his or her own withdrawals to fail without affecting the rest of the contract’s work [185, 184]. Also, we omit the introduction of states in the SLA smart contracts for simplicity. SLA smart contracts are sub-contracts generated by the auction smart contract and are not the focus of this chapter on the auction problem. Their functions and state machines can be customized by users themselves.

The auction smart contract can be expressed in seven states: “Ready”, “Initialized”, “Pending”, “Settled”, “Violated”, “Successful”, and “Canceled”. The “Ready” state is automatically enabled when the auction smart contract is deployed on the blockchain. Within this state, the customer can set up an auction and promote the required services. Providers can then register as bidder candidates; when enough bidders register, the auction is “Initialized". In this state, registered providers can submit and reveal their bids. The customer also needs to reveal the reserve price and place all the bids by order. If there are enough bids to meet the customer’s requirement, the auction comes into the “Pending” state. This means that the bidding phase has finished and the auction enforcement is pending. The customer must now invoke the interface to automatically generate SLA smart contracts for winning providers and wait for their acceptance. The auction is only “Settled” when all SLA smart contracts are signed and there are enough registered witnesses to monitor the SLAs. In this stage, witnesses monitor and submit their results and the customer calculates the witness fee for each witness according to the payoff functions. Finally, if all SLAs are performed as agreed, the auction state changes to “Successful". The providers can withdraw their own service fees. By contrast, if there are any violations occur, the auction state then converts to “Violated", and the customer can withdraw the prepaid service fee for specific violated providers. It should be noted that if any of the above auction conditions are not met, the sale is “Canceled". To retrieve the prepaid deposit, the customer and providers can use the “withdrawDeposit” interface. The customer can also use the “resetAuction” interface to reset the auction state to the “Ready” stage and wait for the next auction round.

It should be noted that the above process may cause a waste of resources for the customer and providers when they have already reached an agreement and there are not enough witnesses. However, we argue that such design is necessary and reasonable; if we let witnesses register in advance, this may also result in a waste of resources for witnesses, i.e., one may register as a witness (with a transaction fee on the blockchain) but not perform the monitoring task to win profits because the

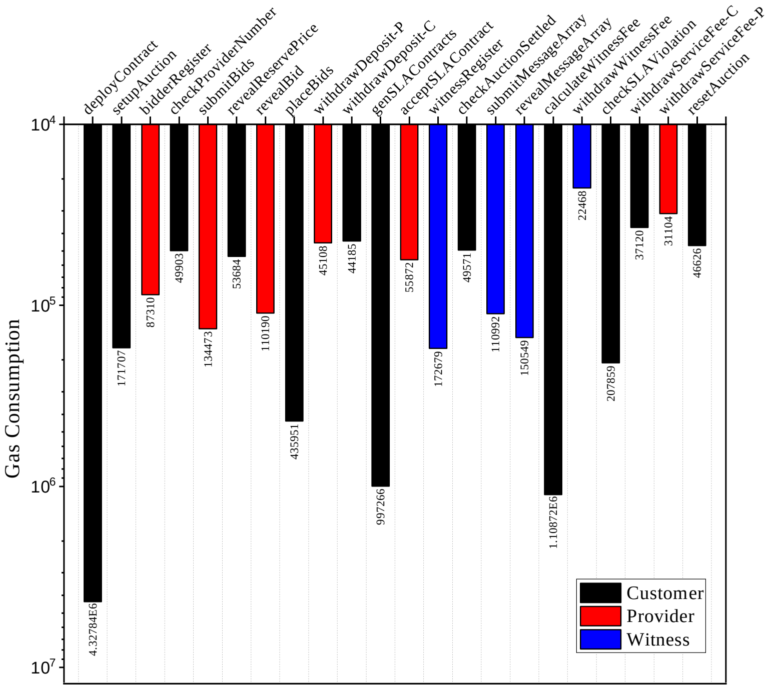


Figure 4.8: Gas consumption of each function interface.

bidding fails. We chose the current design because the trustworthy enforcement of our model is based on witnesses' monitoring, and thus more witnesses should be incentivized to participate. To reduce the possibility of the waste of resources for the customer and providers in the current model, some measures can be taken. For example, an option could be added to extend the witness registration window to wait for more witnesses to join. The original witness fee can also be upgraded to provide an incentive for more witnesses to join within a specified time window.

The smart contract is tested using the Kovan[8], which is one of the most famous Ethereum testnets in the community. To evaluate the cost of a real cloud auction, we create several accounts, deposit some tokens (Ethers) in advance, and then simulate a federated cloud auction scenario with four providers and six witnesses. Figure 4.8 reveals the detailed gas consumption of each interface, while

Table 4.3: Transaction fee of each auction participant in a specific auction event.

Participant	Gas Consumption	Transaction Fee	USD
Customer	3202593 Gas	0.0342 ETH	$91.00
Provider	464057 Gas	0.0048 ETH	$12.76
Witness	456688 Gas	0.0050 ETH	$13.25

[8] `https://kovan-testnet.github.io/website/`

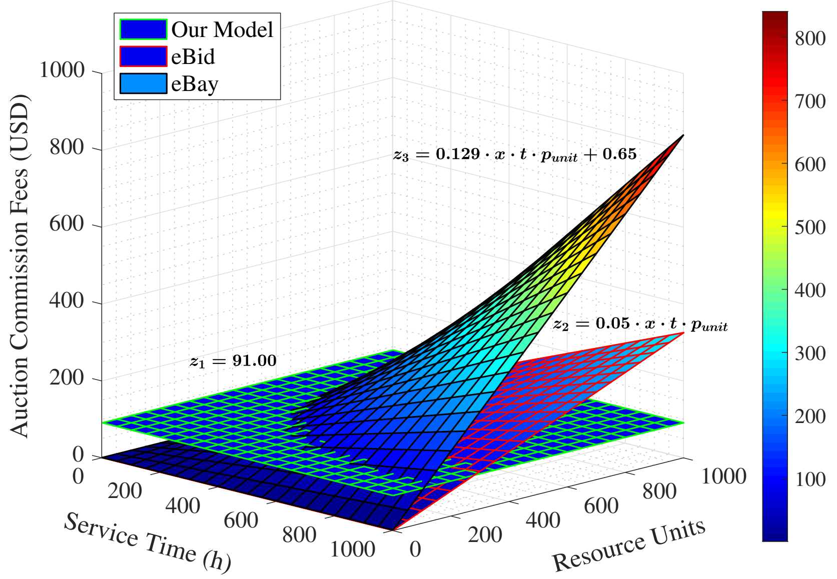


Figure 4.9: Theoretical comparison of auction commission fees. The unit price of resources in the figure is illustrated using the cost of the smallest and cheapest Amazon EC2 instance t2.nano ($0.0058/h).

Table 4.3 shows the total transaction fee of each auction participant (converted to US dollars).[9] In general, the customer needs to invoke the largest number of interfaces (more than 50%) and consume the largest amount of gas, which is in line with our expectations. The customer is the beneficiary and initiator of the service auction and therefore should bear more commission charges. In fact, the customer not only acts as the auction publisher in the model, but also assumes the tasks of deploying contracts (consuming the most gas) and triggering some functions related to auction management, e.g., “setupAuction", “placeBids", and “genSLAContract". These tasks are performed by auctioneers or auction platforms in traditional auctions. Besides, since all the auction rules are openly hard-coded in the smart contract, any specific third party can initiate the smart contract while ensuring the fairness and credibility of the auction. It can also be found from Table 4.3 that the transaction fees of each witness and provider are not expensive (around $13), which ensures both parties have sufficient motivation to participate in the auction.

We further compare the commission fees of our model with popular online auction platforms (eBay and eBid), as shown in Figure 4.9. The eBay auction fee is 12.9% of the total sale ($7500 in maximum) plus an insertion fee ($0.35)

[9] The exchange rate between ETH and USD is changing continuously. When collecting the data, the exchange rate is 1 Ether = $2662.08. Here the transaction fee is determined by both gas consumption and gas price. The difference in gas prices caused a larger transaction fee for witnesses than providers.

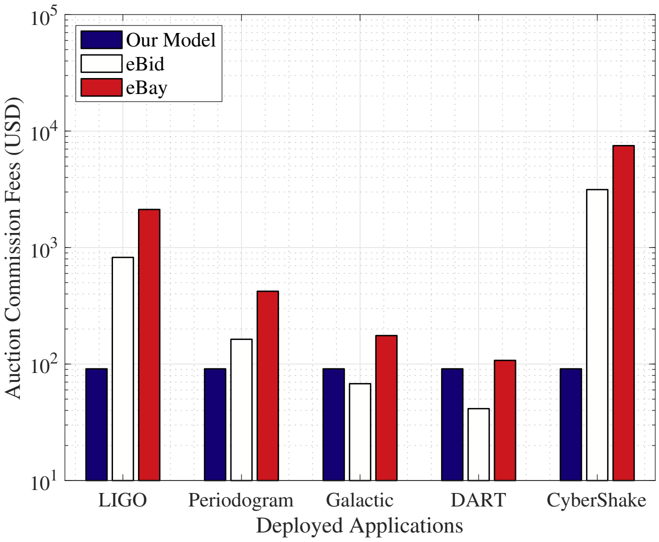


Figure 4.10: Experimental comparison of auction commission fees.

and an additional handling fee per order ($0.30) [50]. In contrast, eBid's base fee is 5% of total sales [51]. The difference between the two is that while eBid is cheaper, eBay is generally considered to have a stronger market share and buyer base. As can be seen from the plot, the fees for eBay and eBid increase exponential as the service time and resource units increases. This is based on the fact that their fee mechanism is determined in proportion to the final sale price of the auction. In contrast, the customer, as the initiator of the auction, only pays a fixed commission fee (i.e., $91.00 in this case) in our blockchain-based model. This fee is independent of the final price of the auctioned services.

Figure 4.10 shows the comparison of the fees when auctioning different applications and services. We selected five real scientific workflow applications provided by the Pegasus workflow gallery; their computational resources and running time can be found in [161]. As observed from the figure, our model has different price advantages compared to eBay and eBid when deploying LIGO, Periodogram, and CyberShake applications. In the case of CyberShake, due to the need for a massive computing cluster and service time, our commission fees are just 2.89% and 1.12% of eBay and eBid, respectively. Whereas in Galactic and DART applications, choosing eBid would be more economical because of the small final auction price. In fact, when the price of the auctioned cloud service is less than the commission fee, our model will not be applicable to public blockchains like Ethereum. At this time, a fee-free permissioned blockchain can be used as an alternative, and the whole proposed model is still valid.

Finally, it is important to note that the price of Ether is volatile, and there is a payment risk for users. We thus consider the fluctuations in the history of Ether to USD. Figure 4.11 shows the changing trend in fees for the three actors

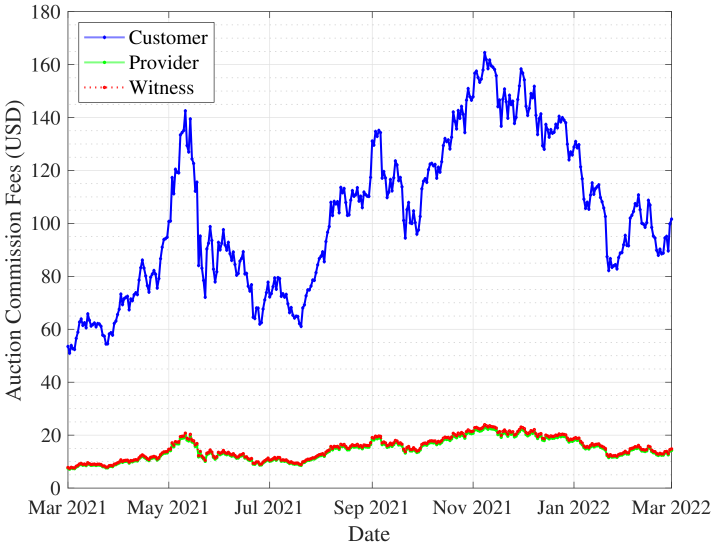


Figure 4.11: Historical changes in auction transaction fees on Ethereum.

in our model over the last year. It can be observed that the provider and witness only pay a small fee with small changes. In November 2021, the transaction fee reached its maximum, making it more expensive for customers to initiate an auction. However, the recent trend of a significant decrease indicates that our model is more cost-effective compared to other online auction platforms.

4.5 Conclusion

In this chapter, an incentivized AWESOME framework for federated cloud services is proposed. We leveraged Bayesian game theory to analyze the bidding of providers and the SLA monitoring of witnesses, with two unique BNEs generated for two respective groups. The first BNE enables the selection of cost-effective and suitable service providers to construct the federated cloud services, while the second BNE ensures that the witnesses can report the truth about service violations consistently, which further makes the auction enforcement trustworthy. We validated the equilibrium results of two BNEs and implemented the proposed model on the Ethereum blockchain. The analytical and experimental results demonstrated our model's feasibility, trustworthiness, and cost-effectiveness.

Chapter 5

Towards a Scalable AWESOME Framework: A Permissioned Blockchain Approach and Empirical Study

According to the design requirements, the core function of AWESOME is to leverage the blockchain as the underlying infrastructure to enhance the trust among the decentralized service providers and customers. Therefore, the blockchain, as the fundamental layer of the entire architecture, is the key part to support all the transactions. In the previous two chapters, we have validated the model feasibility using the Ethereum permissionless blockchain. To further demonstrate the feasibility and scalability of the AWESOME approach, in this chapter we conducted an empirical study of different permissioned blockchain platforms to analyze their performance. The results and discussion in this chapter will provide insights for choosing the right blockchain platform when building the AWESOME ecosystem.

This chapter is based on:

- **Zeshun Shi**, Huan Zhou, Yang Hu, Jayachander Surbiryala, Cees de Laat, and Zhiming Zhao. "Operating permissioned blockchain in clouds: A performance study of hyperledger sawtooth". *In 2019 18th IEEE International Symposium on Parallel and Distributed Computing (ISPDC)*, pp. 50-57. IEEE, 2019.
- **Zeshun Shi**, Huan Zhou, Jayachander Surbiryala, Yang Hu, Cees de Laat, and Zhiming Zhao. "An automated customization and performance profiling framework for permissioned blockchains in a virtualized environment". *In 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), workshop on resource brokering with blockchain (RBChain)*, pp. 404-410. IEEE, 2019.
- Huan Zhou, **Zeshun Shi**, Ouyang Xue, and Zhiming Zhao. "Building a

blockchain-based decentralized ecosystem for cloud and edge computing: an ALLSTAR approach and empirical study". *Peer-to-Peer Networking and Applications* (2021): 14(6), pp.3578-3594. (as co-first author)

5.1 Problem Statement

In the previous sections of the thesis, we have already discussed that the blockchain is the main component of the AWESOME framework to provide a decentralized cloud environment. The performance of the blockchain will directly affect the efficiency of the decentralized AWESOME framework when motivating resource providers to adopt our approach and join the ecosystem easily. It is, therefore, important to discuss the performance of AWESOME blockchain infrastructure.

To select the most suitable blockchain technologies for the AWESOME framework, we need to consider some basic questions. For example, does the blockchain have to provide a cryptocurrency to support auction payments? Is the auction designed to be implemented on a private or public network? In addition, some specific business requirements for the auction model need to be considered, such as user scenarios, security, privacy, and scalability. A permissionless Ethereum blockchain is focused on providing a universal platform for various transactions and applications. It has the advantage of being easy to use, secure, and having a wide user base. Therefore, it is suitable for open-outcry auctions and double auctions where a large number of bidders are required. However, its full decentralization and transparency come at the cost of performance and privacy. Therefore, it is more suitable for single-item auctions compared to multi-item auction models that require complex on-chain computation. On the other hand, due to privacy, regulatory, and scalability concerns, enterprises may prefer to use permissioned blockchains rather than permissionless ones to enable auctions. Hyperledger Fabric, for example, provides high throughput to help with on-chain winner determination calculations for some complex auctions (e.g., VCG auctions). However, the disadvantages of using it for auctions are also obvious; it is not equipped with a stable cryptocurrency. Besides, as a permissioned blockchain, it faces greater challenges in terms of data security and immutability. It should be noted that the choice of blockchain platform should be flexible for different auction scenarios. Most existing blockchain platforms are quite extensible and can be improved for different application requirements. For example, Ethereum has designed an alternative privacy deployment version to address the issues in permissionless deployment. Hyperledger Fabric could add an extra token component to solve the problem of not having native tokens, as the system is based on a highly modular design.

Due to the performance limitations of current permissionless blockchains, and the huge energy and money consumption when deploying and executing smart contracts, it is not always desirable to implement all the functions of AWESOME ecosystem on the permissionless blockchain platform. By contrast, the permissioned blockchains, which have exhibited better performance and demonstrated great potential to provide secure services in various industrial scenarios. In fact, permissioned blockchains provide an additional level of security compared to typical permissionless blockchain systems because they require an access control layer. In the AWESOME framework, we want to combine the advantages of permissioned blockchains to create a scalable and secure trading environment for different cloud resource users.

The rest of this chapter is organized as follows: Section 5.2 presents a comparative analysis of the performance of five popular permissioned blockchains. Section 5.3 shows the performance analysis of deploying a permissioned blockchain in a cloud environment. Finally, the chapter is concluded in Section 5.4.

5.2 Comparative Analysis of Five Popular Permissioned Blockchains

5.2.1 Preliminaries

To evaluate the features of different permissioned blockchain platforms, we have designed three experiments:

1. Performance Stability: Which blockchain platform offers the most stable performance at a fixed transaction workload?
2. Resource Consumption: Which blockchain platform consumes less computing resources (CPU and memory) on the host in the same blockchain transaction workload?
3. Performance Scalability: Which blockchain platform is more scalable given the continuously increasing blockchain transactions?

Before diving into the experimental part, we would like to clarify the basic assumptions and settings of our experiments.

- We would not discuss the performance of the permissionless blockchain platform (e.g., Ethereum), as this has been discussed in the previous sections on the implementation and validation of AWESOME. Instead, our evaluation mainly focuses on the discussion of performance comparison of different permissioned blockchain platforms under the same scenario, which is lacking in most current studies.
- Although different platforms implement their own consensus algorithms, the evaluation in this chapter focuses on the comparison of different blockchain

platforms and does not specifically discuss the comparison of consensus algorithms. In our previous research, we have demonstrated that there are some differences in the performance of different consensus algorithms under the same platform [182]. However, we believe that the impact of consensus algorithms on performance is limited by the framework itself.

- A basic smart contract was leveraged to model the decentralized cloud market and to benchmark different blockchain platforms. The functions of this smart contract include general operations in a decentralized cloud market, e.g., generating new transactions (write) and querying existing transactions (read). Since different permissioned blockchains use different smart contract programming environments, we leave the complex functional design of smart contracts to our future work.
- Regarding the benchmark metric, the performance of the blockchain is measured by the commonly adopted metric "throughput" in the permissioned blockchain community. It is defined as the rate at which write/read operations are committed to the blockchain (as shown in Equation (5.1)) and can reflect whether the underlying blockchain infrastructure can meet the requirements of industrial applications.

$$Throughput = \frac{Total\ committed\ transactions}{Total\ time\ in\ seconds} \tag{5.1}$$

- Five promising blockchain platforms were selected as our decentralized infrastructure to support the AWESOME framework, namely Hyperledger Sawtooth, Hyperledger Iroha, Hyperledger Fabric, Hyperledger Besu, and Ethereum.[1] These platforms have been involved in many successful commercial projects. It should be noted that Ethereum is not used as a permissionless blockchain but as a private deployment platform. A more detailed comparison of those five platforms is illustrated in Table 5.1
- We used a MacBook Pro laptop as the running machine, with 2.9 GHz Intel Core i5 CPU and 16 GB 1867 MHz DDR3 memory. We used docker container to deploy our blockchain network.

5.2.2 Experimental Results

Based on the above assumptions and settings, we designed experiments with three dimensions to discuss scalability, stability, the resource consumption (CPU, memory) of different blockchain platforms. More specifically, the experiments related to scalability is to explore the blockchain performance under different transaction input patterns. Performance stability experiments leverage the same transaction input rate to test the distribution and stability of throughput results.

[1] For simplicity, Sawtooth, Iroha, Fabric, Besu, and Ethereum are used in the following text.

Table 5.1: Comparison of five permissioned blockchain platforms.

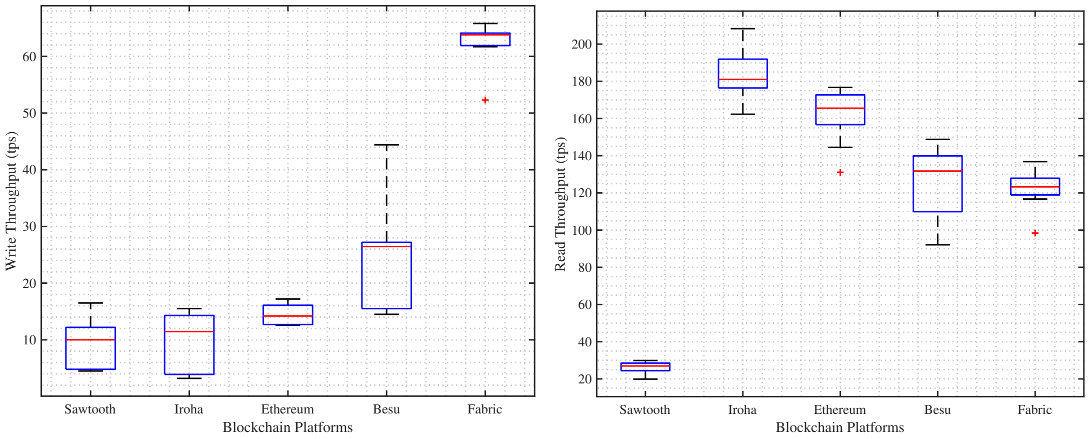
Blockchain	**Introduction**	**Consensus Algorithms**	**Smart Contract Protocol**	**Key Features**
Hyperledger Besu	An Ethereum client that can run on public networks, test networks, and private permissioned networks.	PoW and PoA	Smart Contract	- Support multiple types of Ethereum networks
Ethereum (private)	An open-source blockchain platform with smart contract functionality that handles transactions via the EVM.	PoA	Smart Contract	- Mature blockchain and DApp system - Customized for enterprise applications
Hyperledger Fabric	An open-source permissioned blockchain with a highly flexible and adaptable design for enterprise usage.	Raft and Kafka (deprecated in v2.x)	Chaincode	- Multi-ledger structure - Private data storage in channels
Hyperledger Iroha	A framework for incorporating blockchain into infrastructures or IoT projects that are easy and quick to implement.	YAC	Command (built in contracts)	- Easy to use - Built-in smart contract functionality
Hyperledger Sawtooth	A secure and modularity-based architecture for creating enterprise-level permissioned blockchains.	PoET, Raft, and PBFT	Transaction Processor	- Parallel transaction execution - Customizable transaction processors

Finally, the resource consumption experiment discussed the CPU and memory consumption of different blockchain platforms under different transaction models.

Stability Evaluation

In this experiment, we used a fixed transaction input rate with 100 tps and repeat it 10 times to observe the performance stability of the blockchain platform when facing the same transaction request multiple times. In fact, such an experimental setup can reveal the performance of the blockchain platform in the continuous processing of the same or similar transaction requests.

As can be seen from Figure 5.1a, in terms of write throughput, the performance of Fabric is the most stable one, and its throughput is much higher than other platforms. Although Ethereum's performance is relatively stable as well, the average throughput is not high, only about 14 tps. Iroha, Sawtooth, and Ethereum are at the same level and their stability is relatively poor. In terms of read throughput, although Sawtooth has the most stable performance and small fluctuations, its throughput is far less than the other four. In contrast, Fabric and Iroha have better stability whereas Besu has the worst one, which can be seen from Figure 5.1b.



(a) Variation in write throughput

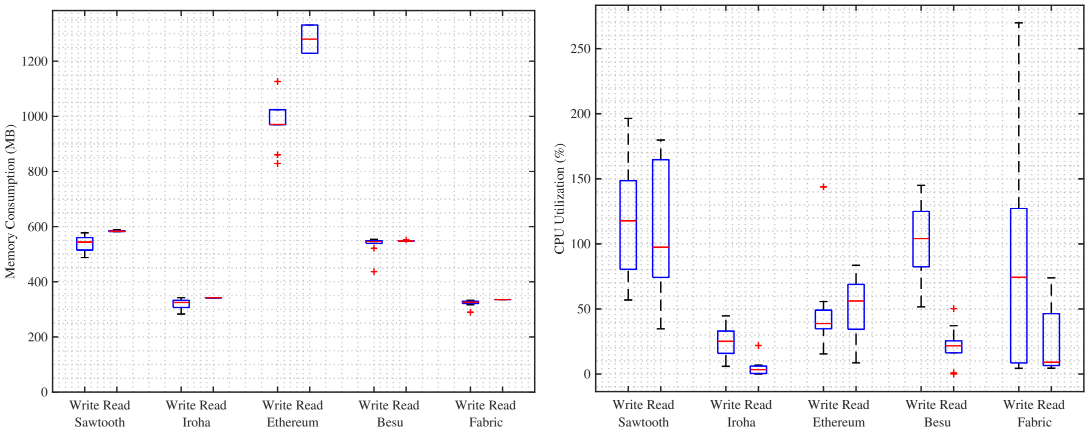
(b) Variation in read throughput

Figure 5.1: Variation in throughput of different blockchain platforms

Resource Consumption Evaluation

In this experiment, we use a linear transaction input rate to observe the CPU and memory resource consumption of the blockchain platform when facing transaction requests under dynamic transactions patterns.

As can be seen from Figure 5.2a, in general, the memory consumption of several blockchain platforms is relatively stable. More specifically, Ethereum is more



(a) Variation in memory consumption

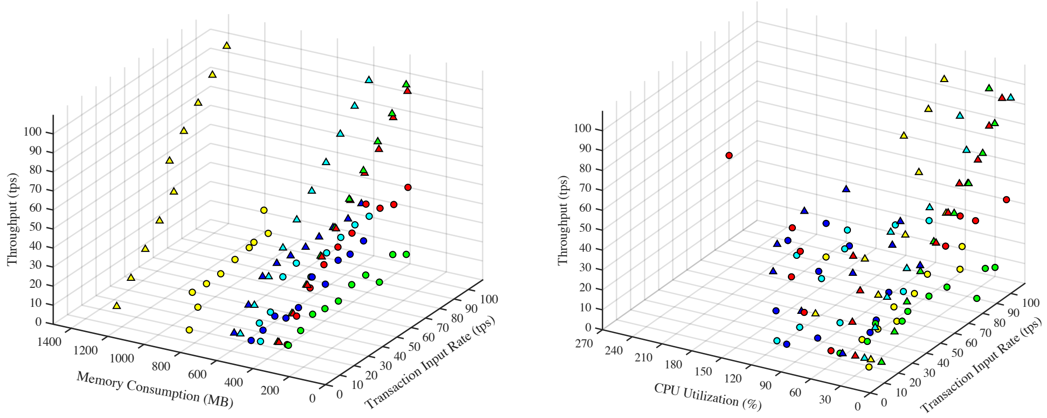
(b) Variation in CPU utilization

Figure 5.2: Variation in resource consumption of different blockchain platforms

sensitive to memory consumption in our transaction mode. Its average memory consumption is about 970 MB for write and 1280 MB for read. Sawtooth and Besu have similar memory consumption for write/read, both between 500-600 MB. Iroha and Fabric have similar memory consumption, both of which are around 330 MB. In addition, the memory consumption of read operation is more stable than write operation on most platforms. Ethereum is a special case. Although its overall memory consumption for write is more stable than read, there are three outliers in the write part.

Compared with the stable memory consumption, the CPU utilization of the different platform changes significantly and tends to be unstable, as shown in Figure 5.2b. Most platforms have higher CPU utilization when doing write operations, except for Ethereum. Among them, Sawtooth has the most CPU usage, and its average CPU utilization for write and read reached 119% and 107%, respectively. This proves that the current transaction input volume has reached the performance bottleneck. Iroha has the lowest CPU utilization, with write/read operations for only 25% and 5%.

There are also findings from the scatter plot shown in Figure 5.3a and Figure 5.3b. With the linear increase of transaction input rate from 0 to 100 tps, the improvement of the read performance of the platform is more obvious than the write performance. At the same time, the increase in transaction input rate is usually accompanied by an increase in memory consumption. Finally, when the input rate is increased, the changing trend of CPU utilization shows dynamic fluctuations and worse stability.



(a) Variation in memory consumption

(b) Variation in CPU utilization

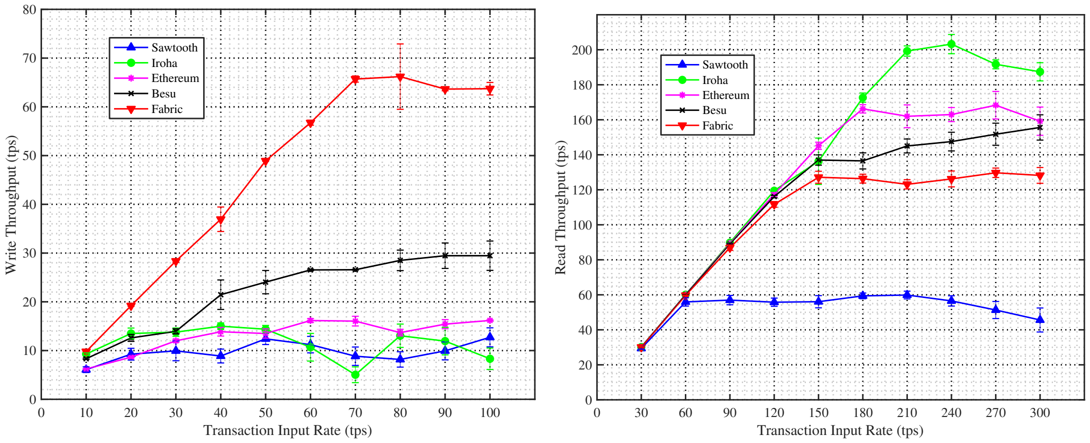
Figure 5.3: Scatter plot of blockchain resource consumption impact on performance. Blue, green, yellow, cyan, and red dots represent Sawtooth, Iroha, Ethereum, Besu, and Fabric, respectively. The triangle represents read operations, and the circles represent write operations.

Scalability Evaluation

In this experiment, we used the transaction input rates that increased linearly from 10 tps to 100 tps to observe the performance scalability of the blockchain platform when facing different levels of transaction requests. In practical scenarios, it often happens that blockchain transaction requests increase rapidly in a short period of time. Therefore, such a rate control strategy can simulate the performance change of the blockchain platform when processing transaction requests with different densities.

Figure 5.4a shows the changes in write throughput of different blockchain platforms at different transaction input rates. When the rate increases, the performance trend is to increase first and then stabilize at the bottleneck value. It can be seen that the performance of Fabric is significantly better than other blockchain platforms. Besu takes second place, and the remaining three are at the same level. For Fabric, the performance bottleneck is reached when the input rate is 70 tps. Whereas for Iroha, Sawtooth, and Ethereum, when the input rate reaches 20 tps, the performance already reached its limit. In general, the performance of Iroha, Sawtooth, and Ethereum shows some fluctuations, but the performance of Sawtooth tends to be the worst. It can also be seen that Ethereum outperforms Iroha at high transaction input rates (greater than 50 tps).

Similarly, Figure 5.4b shows the changes in read throughput of different blockchain platforms. In order to show trends and bottlenecks more clearly, we increased the maximum input rate to 300 tps. It can be seen that compared with

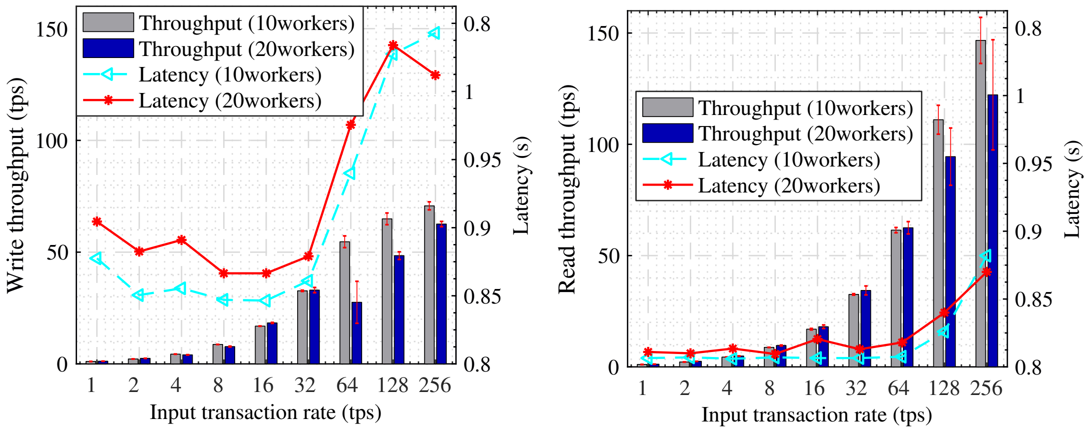


(a) Write throughput (b) Read throughput

Figure 5.4: Impact of transaction input rate on blockchain performance.

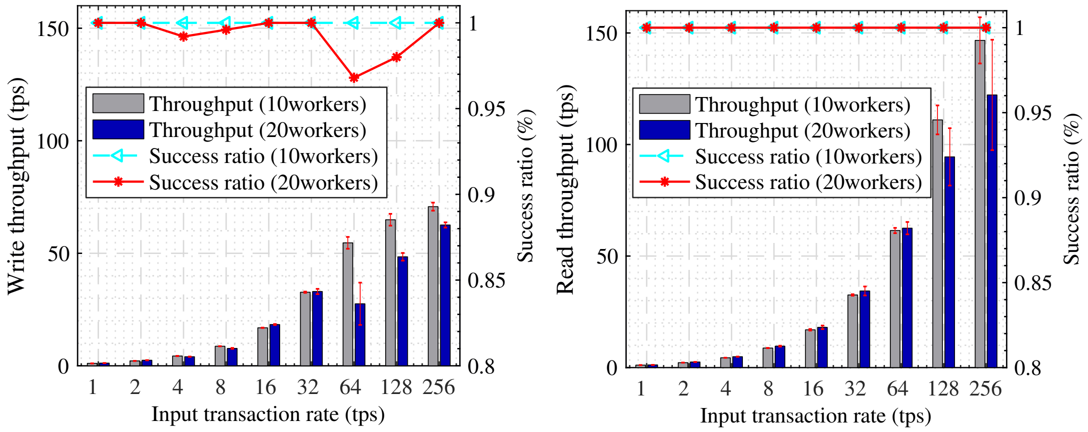
write performance, the read performance of major blockchain platforms has been greatly improved. Overall, Iroha performed the best, with a bottleneck of around 200 tps, followed by Ethereum, Besu, and Fabric. Not surprisingly, Sawtooth has the worst read performance, with a bottleneck of about 60 tps.

Through the above comparison experiment, we can conclude that although Fabric's read performance (throughput and latency) is not the best, it has the best performance for write operations and acceptable performance for read operations. In fact, write performance is particularly critical in our model because more transactions submitted to the blockchain imply a higher level of security for



(a) Write throughput (b) Read throughput

Figure 5.5: Performance (latency) of Fabric at different transaction workloads and worker numbers.



(a) Write throughput

(b) Read throughput

Figure 5.6: Performance (success ratio) of Fabric at different transaction workloads and worker numbers.

integrity verification. In contrast, read performance is not as critical since data auditing is not required at high frequency. Based on such comparative experiments, we conclude that Fabric is the blockchain platform that best meets our scalability requirement.

We further interprets the performance of the Fabric under different worker numbers. At this time, different numbers of workers (10 and 20) are leveraged to diversify clients connected to the blockchain network. In addition to throughput, the following two metrics are used as a supplement to the evaluation.

$$Success\ Ratio = \frac{Successful\ transactions}{Total\ submitted\ transactions} \tag{5.2}$$

$$Latency = Response\ time - Submission\ time \tag{5.3}$$

As can be seen from the plot, the highest write/read throughput can be found for both numbers of workers at an input transaction rate of 256 tps. When the number of workers is doubled from 10 to 20, the measurement's throughput and stability are slightly decreased. However, the overall performance bottleneck is maintained at a high level (write and read throughput are 60 tps and 120 tps for 20 workers). The performance is also compared with the transaction success ratio, as shown in Figure 5.6a and Figure 5.6b. It can be seen that when the number of workers is 10, all transactions can be submitted successfully. However, when worker numbers increase to 20, some of the write operations may fail when the input transaction rate is high. Nevertheless, the success ratio is maintained above 95% in all experiments, which demonstrates the good scalability of the model. It is also evident from the latency experiments in Figure 5.5a and Figure 5.5b that the latency increases in different degrees when both the number of workers and the input transaction rate increase. However, all the latency values are maintained within 1.5 seconds. This low latency of the Fabric blockchain ensures that the entire model can work efficiently.

In Table 5.2, we show the latency result of Fabric with different worker numbers. It can be seen that despite some fluctuations, the maximum, minimum, and average latency increases for both worker numbers when increasing the input transaction rate from 1 to 256 tps. If we take the example of 1 000 verification transactions, in the worst case (i.e., an input rate of 256), 10 and 20 workers can still complete the workload in an average of 16 and 18 minutes. Latency only increased by 9% when the number of workers doubled. Compared to write operations, read operations have very low latency, with 1 000 queries costing only about 30 seconds to complete. These experiments demonstrate that the latency satisfies the scalability requirements of our model.

Table 5.2: Execution latency of Fabric with different worker numbers.

Blockchain	**Input Rate (tps)**	**Write Operation**			**Read Operation**		
		Max Latency (s)	**Min Latency (s)**	**Avg Latency (s)**	**Max Latency (s)**	**Min Latency (s)**	**Avg Latency (s)**
Fabric (10 workers)	1	0.684 ± 0.390	0.124 ± 0.023	0.310 ± 0.057	0.074 ± 0.022	0.010 ± 0.000	0.026 ± 0.005
	2	0.342 ± 0.049	0.070 ± 0.000	0.202 ± 0.013	0.086 ± 0.030	0.012 ± 0.004	0.028 ± 0.008
	4	0.370 ± 0.025	0.082 ± 0.013	0.222 ± 0.026	0.056 ± 0.009	0.010 ± 0.000	0.024 ± 0.005
	8	0.364 ± 0.130	0.070 ± 0.007	0.188 ± 0.022	0.074 ± 0.018	0.010 ± 0.000	0.028 ± 0.004
	16	0.336 ± 0.044	0.076 ± 0.013	0.186 ± 0.009	0.062 ± 0.023	0.010 ± 0.000	0.026 ± 0.005
	32	0.558 ± 0.566	0.070 ± 0.010	0.244 ± 0.160	0.126 ± 0.128	0.012 ± 0.004	0.026 ± 0.013
	64	0.802 ± 0.130	0.216 ± 0.057	0.560 ± 0.079	0.108 ± 0.024	0.010 ± 0.000	0.030 ± 0.000
	128	1.206 ± 0.042	0.610 ± 0.046	0.910 ± 0.032	0.274 ± 0.032	0.012 ± 0.004	0.104 ± 0.015
	256	1.236 ± 0.009	0.678 ± 0.049	0.972 ± 0.018	0.548 ± 0.031	0.012 ± 0.004	0.328 ± 0.019
Fabric (20 workers)	1	0.892 ± 0.442	0.136 ± 0.061	0.418 ± 0.104	0.134 ± 0.054	0.012 ± 0.004	0.044 ± 0.026
	2	0.604 ± 0.090	0.120 ± 0.026	0.330 ± 0.033	0.144 ± 0.092	0.014 ± 0.005	0.040 ± 0.014
	4	1.430 ± 1.743	0.133 ± 0.036	0.333 ± 0.043	0.200 ± 0.175	0.012 ± 0.004	0.054 ± 0.043
	8	0.485 ± 0.062	0.108 ± 0.025	0.265 ± 0.026	0.100 ± 0.031	0.014 ± 0.005	0.038 ± 0.004
	16	0.444 ± 0.055	0.124 ± 0.034	0.266 ± 0.021	0.248 ± 0.224	0.016 ± 0.009	0.082 ± 0.105
	32	0.594 ± 0.203	0.120 ± 0.027	0.316 ± 0.086	0.124 ± 0.106	0.012 ± 0.004	0.052 ± 0.050
	64	0.860 ± 0.121	0.230 ± 0.092	0.547 ± 0.111	0.264 ± 0.328	0.012 ± 0.004	0.072 ± 0.069
	128	1.175 ± 0.114	0.683 ± 0.047	0.945 ± 0.068	0.320 ± 0.289	0.020 ± 0.007	0.160 ± 0.174
	256	1.390 ± 0.054	0.818 ± 0.061	1.060 ± 0.029	0.490 ± 0.047	0.018 ± 0.004	0.280 ± 0.029

5.2.3 Lesson Learned

Based on the observations above, we draw the conclusions as follows:

1. In terms of platform performance stability, Fabric has the most stable write transaction performance. Although Sawtooth has the most stable read performance, its average throughput is very low. In contrast, Iroha has both stable and the highest average read performance.
2. In terms of platform resource consumption, Ethereum needs more memory consumption under our transaction patterns. Its average memory consumption is about 970 MB for write and 1280 MB for read. Sawtooth and Besu have comparable memory consumption for write/read between 500-600 MB. The same situation comes to Iroha and Fabric, both of which are around 330

MB. In addition, when the transaction rate increases, it is usually accompanied by an increase in memory consumption. However, the CPU utilization is in a state of random dynamic fluctuations.

3. In terms of performance scalability, with the increase of input transaction volume, the write/read performance of different blockchain platforms will first increase and then reach its bottleneck. Specifically, Fabric has the highest write performance of 66 tps, and Iroha has the highest read performance bottleneck at around 203 tps. By contrast, Sawtooth has the worst overall performance, with write and read bottlenecks of about 13 tps and 60 tps, respectively.

The above experimental results give us a lot of insights when building the AWESOME framework. By analyzing the demand of AWESOME on the underlying blockchain infrastructure, we believe that write throughput is the most important metric for evaluating AWESOME in response to new service transactions submitted by large-scale cloud users. At the same time, the resource consumption of the AWESOME blockchain should be maintained at a low level to cope with the limited computing power of lightweight Edge devices.

In our experiments, Fabric has the optimal write throughput in terms of scalability and stability, while the read throughput remains at an acceptable level. It also has the lowest and most stable memory consumption. Therefore, we conclude that Fabric is the best choice to build the AWESOME framework at this moment. This is in line with our expectation that Fabric is currently the most successful permissioned blockchain available in the market. Our next plan is to choose Fabric as the main infrastructure to build the AWESOME framework. Nevertheless, each platform has its own features and there exist trade-offs between different platforms. For example, while Iroha is less impressive in terms of write throughput, it has the highest read throughput and the most stable and lowest CPU consumption, which is a perfect choice for lightweight devices.

5.3 Operating Permissioned Blockchains in Clouds: A Case Study of Hyperledger Sawtooth

Cloud environments provide elastic and cost-effective resources for data storage, processing, and computing. Nowadays, more and more enterprises are migrating their applications into clouds to save operation costs, deploy and operate services continuously, and increase the efficiency of IT infrastructure management [107]. The adoption of cloud services for operating the rapid growth blockchain-based systems has encouraged researchers to study the applicability of blockchain techniques in various business environments. However, the barriers to deploying a blockchain in clouds still exist, and the performance of the blockchain platform is often unstable due to too many influencing factors in a dynamic cloud environment. In fact, performance is a critical factor for the enterprise to utilize a blockchain-based solution, and it directly determines whether the platform is applicable. Evaluating the permissioned blockchain platform in cloud environments is highly important to provide insights into deploying the AWESOME framework.

5.3.1 Preliminaries

Hyperledger Sawtooth & PoET Consensus

Hyperledger Sawtooth is a permissioned blockchain platform for creating networks and distributed applications. The main design philosophy of Hyperledger Sawtooth is to simplify the development process of the blockchain application by separating the central system from the application layer. Enterprise users and application developers can use their own language to specify the business rules that are appropriate for their application without having to understand the underlying design of the core system [100]. A Sawtooth node participating in the system mainly consists of the following components: a validator, a REST API, some transaction processors, and clients. The validator is the core component of Hyperledger Sawtooth. Its main functions include receiving the transaction requests and forwarding them to the corresponding transaction processor. In addition, the validator needs to decide how to generate a new block based on the processing result of the transaction processor and how to echo the result to clients. Meanwhile, the validator also works with other validators to keep the global state of the Sawtooth network consistent. Transaction processors are used to encapsulate the application logic and business models. They work similarly to chaincodes in Hyperledger Fabric and smart contracts in Ethereum. Finally, A REST API is a bridge between the validator and clients.

PoET consensus algorithm is first developed by Intel in 2015 based on the Intel Software Guard Extensions (SGX) hardware, utilized as a Trusted Execution Environment (TEE). However, it is possible to use PoET in non-Intel-based systems using the PoET simulator with Hyperledger Sawtooth. PoET is a lottery based consensus algorithm. The use of TEE makes sure the selection process is carried out fairly. Intel merged this project with Hyperledger in 2016. After that, PoET becomes a trade mark of Hyperledger Sawtooth. Associated with the context of Sawtooth, PoET works as follows: 1) each validator requests for a waiting time from the trusted module (enclave); 2) the enclave randomly assigns a waiting time for each validator; 3) the validator with the shortest time becomes the leader; and 4) once the waiting time has elapsed, the validator can claim the leadership with the verification of the allocated waiting time. Apart from PoET, Hyperledger Sawtooth also supports pluggable consensus algorithms like PBFT and Raft [98].

Experimental Setup

To perform an in-depth study of the Hyperledger Sawtooth blockchain and benchmark the performance of Sawtooth with different cloud service providers and virtual machine (VM) instance configurations, we conducted several experiments with AWS (Amazon Web Services)[2] and ExoGENI[3] testbeds. AWS is a public cloud service platform that provides computing power, database storage, content delivery and other professional features to help businesses. On the other hand, ExoGENI is a community cloud service platform, which provides Networked Infrastructure-as-a-Service (NIaaS) for scientific experimentation. Through various experiments, we want to investigate following questions:

1. Performance Consistency: with the same Sawtooth transaction workload, will the platform perform consistently with the same cloud and VM configuration?
2. Performance Stability: with the same Sawtooth transaction workload, will the platform perform stable with different clouds or VM configurations?
3. Performance Scalability: with different Sawtooth transaction workloads, will the platform perform scalable with varying configuration parameters?

All the experiments of this section are conducted in two clouds. In AWS, we select data centers of Virginia, California, Frankfurt, Sydney, Sao Paulo, and Singapore. In ExoGENI testbeds, we use following data centers: Pittsburgh Supercomputing Center (PIS), Oakland Scientific Facility (OSF), RCI in Chapel Hill (RCI), West Virginia Net (WVN), University of Alaska (UAF), and UMass Amherst (UMASS). We use three different instance types from both cloud providers, i.e., "XOSmal", "XOMedium", "XOLarge" from ExoGENI and "t2.Small", "t2.Medium", "t2.Large" from AWS. Table 5.3 shows the detailed configurations of instance type from both clouds, all the VMs are installed with "Ubuntu 16.04" operation system.

For the blockchain setup, we adopted Hyperledger Sawtooth v1.1 as the permissioned blockchain platform, and PoET was used as the consensus algorithm. During our experiments, Intkey[4] was used as the benchmark application. Intkey allows to set, increase, and decrease the value of entries stored in a state dictionary. Hence, it can be used to generate comprehensive and stable transaction workloads. Finally, when deploying the Sawtooth blockchain in the cloud, we chose to deploy only one Sawtooth node to one cloud VM. This is because if we deploy multiple Sawtooth nodes on a single VM, there will be congestion among these nodes. In such a scenario, we cannot clarify whether the variation in performance is caused by conflict within the nodes or by some other parameters. It is also the reason that we choose VMs instead of containers. Because VMs can provide much better performance isolation than containers. In addition to VMs that

[2] `https://aws.amazon.com/`

[3] `http://www.ExoGENI.net/`

[4] `https://sawtooth.hyperledger.org/docs/core/releases/latest/cli/intkey.html`

Table 5.3: Resource type offered by ExoGENI and AWS.

Cloud Provider	Resource Name	CPU Cores	Memory	DISK Size
ExoGENI	XOSmall	1	1G	10G
ExoGENI	XOMedium	1	3G	25G
ExoGENI	XOLarge	2	6G	50G
AWS	t2.Small	1	2G	8G
AWS	t2.Medium	2	4G	8G
AWS	t2.Large	2	8G	8G

have deployed the Sawtooth blockchain (by default in this section, the Sawtooth platform consists of 5 nodes), we also deployed a monitor node to collect the real-time performance log data using InfluxDB[5]. Moreover, the entire process of provisioning, deploying, and executing is automated by CloudsStorm[6], from which we can prototype an experiment by assembling available infrastructures and services. Finally, the performance of the blockchain is measured by the commonly adopted metric "throughput", defined in equation Equation (5.1), which is the rate at which transactions are committed to the blockchain platform [97].

5.3.2 Evaluation Results

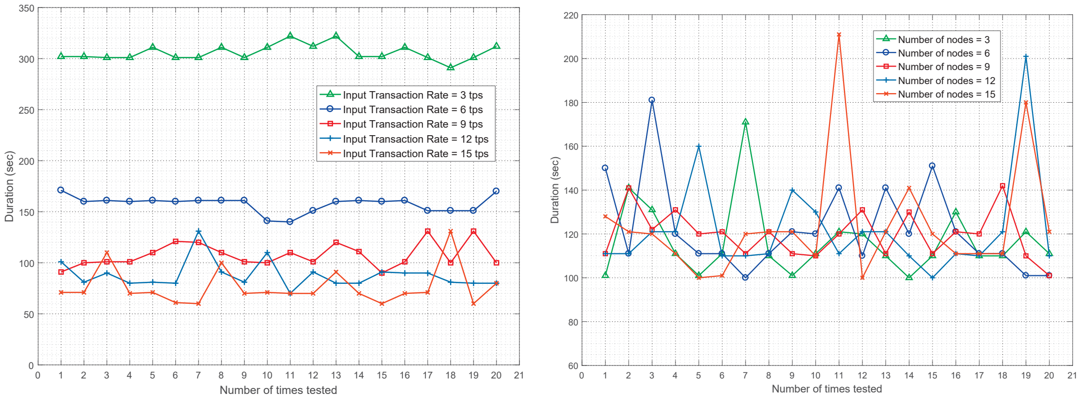
Performance Consistency

In this section, we investigated the performance consistency of the Sawtooth blockchain with a single cloud service provider. All of the experiments were executed on AWS. By repeatedly executing the same Sawtooth benchmark workload multiple times, we can see the variation of the performance. Figure 5.7a shows the results of benchmarking different input transaction rate with the same workload for 20 times. The x-axis is the number of times for testing, and the y-axis represents workload execution duration. Our results show that when the input transaction rate is low, the Sawtooth platform performs more consistently. But at the same time, it should be noted that the workload completion time is longer at lower rate, e.g., 3 tps (transactions per second), which means the input rate of the workload has not reached the performance bottleneck of the platform. When we increased the input transaction rate from 3 tps to 15 tps, the average throughput and variance of duration time increased a lot, as shown in Table 5.4.

Another observation is that as the transaction input rate increases, the overall duration time of the Sawtooth workload decreases. When the transaction input rate is around 12 tps, the current Sawtooth platform processing bottleneck is

[5] `https://www.influxdata.com/`

[6] `https://cloudsstorm.github.io/` [230]



(a) Different input transaction rates.

(b) Different number of VMs.

Figure 5.7: Variation in resource consumption of different blockchain platforms

Table 5.4: Performance variation of different input transaction rates.

Input Rate	Average Throughput	Average Duration	Variance
3 tps	2.93 tps	305.90 sec	57.39
6 tps	5.67 tps	157.65 sec	60.13
9 tps	8.36 tps	107.50 sec	132.25
12 tps	10.24 tps	87.95 sec	172.25
15 tps	12.03 tps	76.40 sec	316.44

reached. As the rate continues to increase from 12 tps to 15 tps, the results of the two transaction rates show a random pattern with a few overlaps.

We also noticed that when the input transaction rate or workload is set to a high value, the transactions from different nodes can be easily forked and lead to rejections. This threshold value is determined by the consensus algorithms, blockchain configuration settings, and network conditions of cloud providers.

Table 5.5: Performance variation of different number of VMs.

VM Numbers	**Average Throughput**	**Average Duration**	**Variance**
3	7.75 tps	116.60 sec	16.34
6	7.43 tps	122.20 sec	20.10
9	7.47 tps	119.80 sec	10.58
12	7.46 tps	122.05 sec	22.23
15	7.40 tps	124.00 sec	26.13

Figure 5.7b and Table 5.5 shows no obvious impact on performance consistency of the Sawtooth blockchain with different number of VM nodes. We used the same

workload and input transaction rate (9 tps) to benchmark the Sawtooth blockchain platform performance. When the number of VM instances changed from 3 to 15, the variance of the platform performance did not change significantly. Actually, the PoET consensus mechanism is especially designed for large networks [99] and it may not be relatively efficient for small networks, comparing to other algorithms, such as Raft or PBFT. Those algorithms may achieve better performance in small networks, but cannot maintain the performance consistency when scaling out the participant nodes. On the contrary, Table 5.5 indicates a trend that PoET is able to keep the performance consistency to fit a large-scale network.

Performance Stability

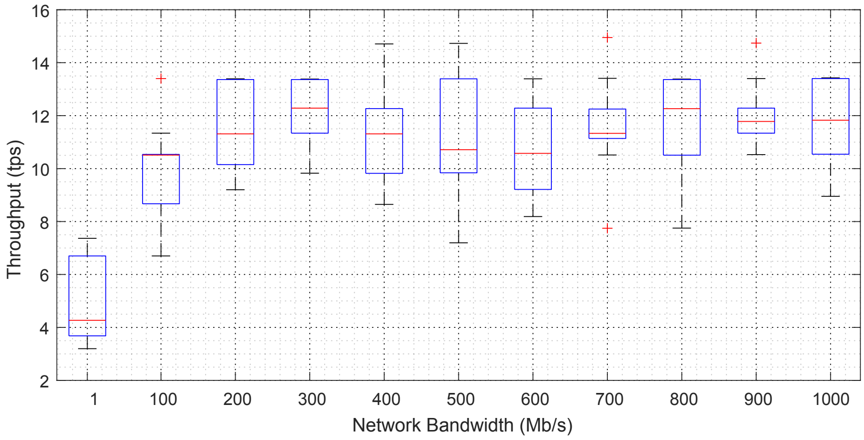


Figure 5.8: Variation in performance of different network bandwidth.

In this experiment, we studied the stability of the Sawtooth blockchain with the same transaction workload in different cloud providers, data centers, and network bandwidth between the VMs. In order to show the performance variation in different clouds, we deployed Sawtooth blockchains in both AWS and ExoGENI testbeds. For each cloud provider, three types of VM specifications and six data centers that are located geographically differently were used. Besides, the Linux Traffic Control (TC) tool was used to configure the bandwidth between the VMs.

Figure 5.8 shows the throughput variation of Sawtooth with different bandwidths, here we use the same workload and input transaction rate (15 tps). In general, when the network bandwidth is greater than 100 Mb/s, the median of platform throughput is around 12 tps. This indicates that platform performance is stable irrespective of bandwidth, and there are only a few cases where 1 or 2 outliers across different bandwidths. When the bandwidth is 100 Mb/s, platform performance begins to show a dropping trend. As the bandwidth drops extremely to 1 Mb/s, the platform performance drops to around one-third (4 tps). From the above results, we can conclude that the bandwidth has a certain impact on the performance of the Sawtooth blockchain, but the Sawtooth blockchain is not sensitive to bandwidth if the bandwidth is beyond a certain threshold. In this case, the platform performance will be influenced, if the bandwidth is below 100 Mb/s.

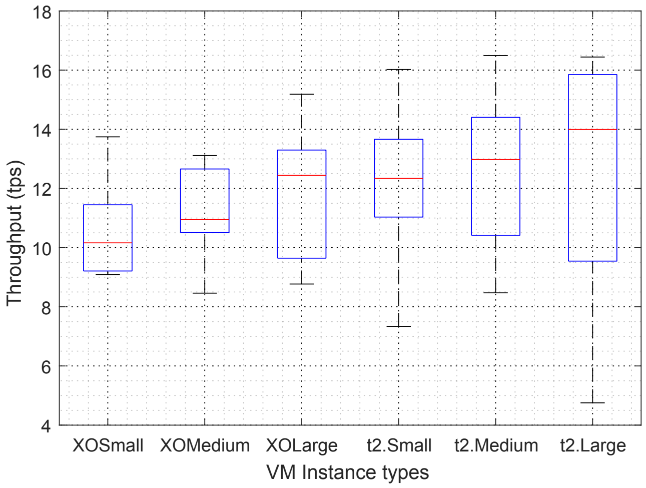


Figure 5.9: Variation in performance of different VM instance types.

Figure 5.9 shows the performance analysis result of Sawtooth with different VM instance configurations. Actually, when the Sawtooth workload is small, the performance variance between different VM instance is not obvious because the number of workload does not reach the limit of the platform. So here we tested a large workload to see the difference between instance types. Overall, AWS outperforms ExoGENI because it has a better VM specification (CPU, Memory, and DISK) in a similar instance type. One observation is that as the VM instance type changes from small to medium and from medium to large, the average performance (throughput median) of Sawtooth has a significant improvement, which can be observed on both ExoGENI and AWS. Further, as the specifications of the VM increase, the performance of the platform is less concentrated. However, it should be noted that this does not mean that the platform is more unstable. In fact, when we input a huge workload, the blockchain deployed in the cluster with small or medium VM configurations often have forks, causing the platform fails to reach a consensus. In this case, we can only restart the blockchain platform and re-run the workload. But in a cluster with large VM configurations, although the throughput is sometimes low (small and medium configurations may have failed in this situation), it still can finally achieve consensuses and work normally, which proves that the configuration with larger VMs improves the platform's resilience.

Finally, Figure 5.10 shows the performance of Sawtooth in two different data centers. Here, we used the same instance type ("XOSmall" for ExoGENI and "t2.Small" for AWS). The six ExoGENI data centers are located in the United States. Among them, the performance of UAF and PIS is more stable than

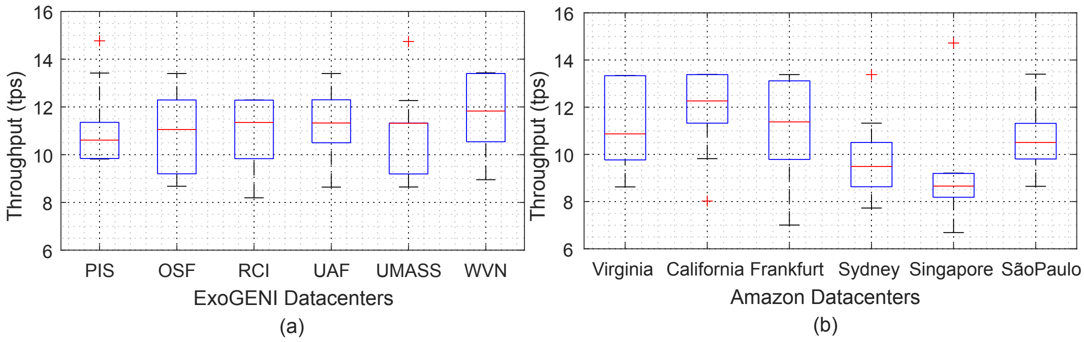
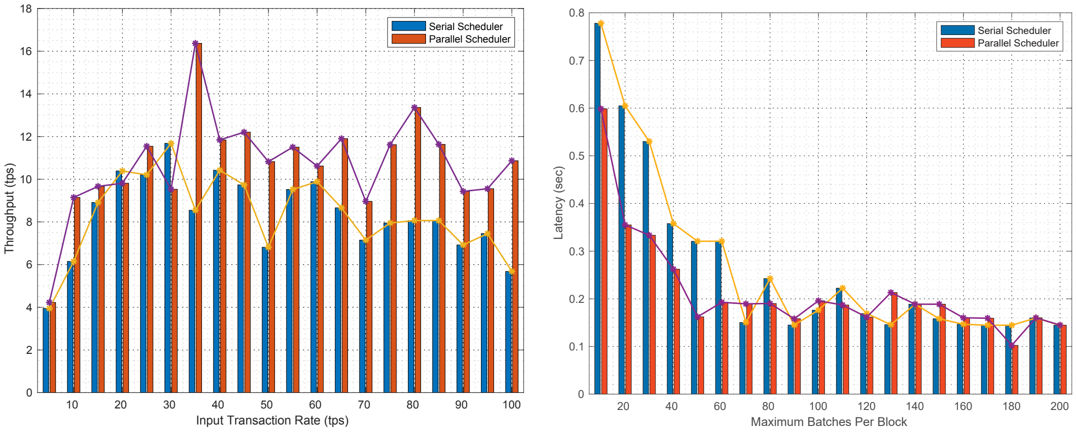


Figure 5.10: Variation in performance of different cloud providers across various data centers.

others, and WVN has the highest average throughput. We checked the resource utilization of different data centers and found this could be an important reason for the above phenomenon. When many customers use the same cloud resource at the same time, the data center becomes busy and the network condition gets worse (e.g., UMASS). On the contrary, when the data center is relatively idle, the performance of Sawtooth platform is better (e.g., WVN). In the AWS section, the California rack has the highest and most stable performance with an average throughput of around 12 tps, followed by the data centers of Frankfurt and Virginia. In contrast, the throughput of Australia and Singapore are lower. An interesting observation is that although Singapore has the lowest point of all test performance, its throughput is quite stable and maintained between 8 and 10. We also noticed that when Sawtooth platform was first built, the performance was extremely high and it went steady with several runs. This is the reason why some high-performance outliers occur.

Performance Scalability

In this section, we investigated the impact of different workloads on platform performance with the same cloud virtual infrastructure configuration. We also tested the platform performance variation when some parameter settings of Sawtooth changed. Here, we changed two parameters provided by Sawtooth: Maximum Batches Per Block (MBPB) and the scheduler type. Batch is the atomic unit of the state change of Sawtooth. In Sawtooth, transactions are carried out in batches. A batch contains a number of transactions, and when a particular transaction fails in the batch, all subsequent transactions fail. We can customize the MBPB parameter in Sawtooth blockchains to meet application requirements. In addition, transactions can be scheduled in the model of serial or parallel, which both produce deterministic results and are completely interchangeable. The running scheduler can schedule the next transaction based on the dependency graph of current transactions. If there is no dependency among the transactions, the parallel scheduler can deliver transactions to multiple transaction processors. On the contrary, the scheduler in serial model always delivers the transactions one by one to the transaction processors. In this experiment, 2 transaction processors are leveraged to perform the comparison.



(a) Different workloads and scheduler types. (b) Different MBPB values and scheduler types.

Figure 5.11: Variation in different parameters for blockchain performance.

Figure 5.11a shows the impact on the platform performance with different workload input transaction rates and scheduler models. In this experiment, we used the same instance type and fixed execution time to observe changes in platform performance for different input transaction rates. As shown from the figure, parallel scheduling model has overall better throughput and platform performance than serial scheduling model. In fact, when changing from serial scheduler to parallel scheduler, the maximum value of throughput has increased from 11.68 tps to 16.37 tps, and overall throughput increased by almost 30%. It can be observed that when the input rate is increased from 10 tps to 100 tps, the platform throughput first reaches the highest point and then falls. Afterward, as the rate continues to increase, platform performance begins to stabilize gradually. We also notice that the performance of the serial scheduler shows a significant downward trend when dealing with the larger transaction input rate. The reason here is that the increased input rate has reached the upper limit of the platform performance, and extra transactions have to be pended or directly rejected. However, the parallel model can achieve better performance at the same input rate, so it is more stable to handle the workload with the large input rate. Therefore, when some transactions are in a non-uniform duration (e.g., realistic complex workloads), the performance advantage in parallel model is greater than that in serial model.

Figure 5.11b shows the impact of the MBPB on platform performance. Here, the latency means the average execution duration for each transaction. As the

MBPB value increases from 10 to 200, the latency first decreases rapidly and then reaches its threshold and tends to be stable. The result also shows that when the MBPB value is less than a certain threshold, the parallel model is significantly better than the serial model. After that, the two models show a random interlaced state because the performance bottleneck has been reached.

5.3.3 Lesson Learned

The blockchain platform is the main component of the AWESOME framework and can influence the service smart contract enforcement in the transaction network. Therefore, the blockchain infrastructure directly determines the efficiency of the AWESOME framework when performing service auction, monitoring, and SLA violation detection.

Considering the trend of leveraging clouds to operate a permissioned blockchain service, we conducted the performance analysis with Sawtooth to simulate different scenarios according to our AWESOME framework. The main findings are the following:

- The number of VMs involved in the blockchain network affects the performance for consensus reasons. We demonstrate that the PoET consensus algorithm guarantees a good scalability level.
- The bandwidth among the nodes of the blockchain affects performance within a certain threshold. Hence, the bandwidth should be properly customized. Very high bandwidth values are of no help for further improving performance.
- The VM type with a bigger capacity can enhance performance. Thus, an appropriate VM type needs to be considered to achieve a balance between performance and costs.
- The data center where the AWESOME blockchain platform is deployed also impacts the platform performance. According to our experiments, an idle data center with low utilization can enable to reach a higher performance level. Nevertheless, a more practical solution is achieved by leveraging real-time profiling before selecting the data center to perform the deployment.

5.4 Conclusion

In this chapter, we conducted performance studies to provide insightful recommendations for deploying a more scalable AWESOME framework based on permissioned blockchains. We first provided a comparative performance analysis of five popular permissioned blockchain platforms. The result showed that Hyperledger Fabric is the best choice to build the AWESOME framework since it has the optimal write throughput and acceptable read throughput. Then, a permissioned blockchain platform (Hyperledger Sawtooth) was selected as an example for performance testing to demonstrate its performance in a dynamic cloud environment. The results showed that factors such as the number and type of VMs, bandwidth, and different data centers can affect the performance of a permissioned blockchain in a cloud environment.

Chapter 6

Conclusions

Industrial applications often require cloud services from multiple providers to improve reliability and flexibility. Traditional selection methods through auctions rely on a centralized auctioneer to coordinate the auction procedure. Blockchain and smart contracts provide a decentralized mechanism to automate the cloud auction process; however, existing solutions fail in the selection of the most suitable providers and the violation detection of the signed auction agreements, which are also known as service-level agreements (SLAs). To tackle these problems, we propose a novel framework called AWESOME based on decentralized auction and witness mechanisms in this thesis. To be specific:

- We first conduct an extensive literature review of blockchain-based models for decentralized auctions and marketplaces in Chapter 2. This chapter lays the theoretical foundation for the blockchain-based cloud marketplace and SLA management solution.
- The AWESOME framework is then introduced in Chapter 3. It contains four subsystems: a customizable graphical user interface, an auction-based service selection model, a witness committee management mechanism, and a smart contract factory orchestration. We developed a prototype AWESOME decentralized application (DApp) based on the Ethereum blockchain. Extensive experiments are designed to demonstrate the feasibility of the proposed model and DApp.
- Next, an incentivized AWESOME framework for federated cloud services is proposed using Bayesian game and Bayesian Nash Equilibriums (BNEs) in Chapter 4. The first BNE enables the selection of cost-effective providers to construct the federated cloud services, while the second BNE ensures consistent and trustworthy monitoring of federated SLAs. Moreover, a timed message submission (TMS) algorithm is proposed to protect the auction privacy during the message submission phase.

- Finally, to provide a scalable AWESOME approach, we conducted an empirical study of different permissioned blockchain platforms to analyze their performance in Chapter 5. The results and discussions in this chapter provide insights for choosing the right blockchain platform when building the AWESOME ecosystem.

In this chapter, we first conclude the thesis by answering the research questions proposed in Section 1.1. Then, we provide the lessons learned while designing and developing the AWESOME framework.

6.1 Conclusions of Outcomes

This thesis aims to use blockchain technology to enhance the cloud marketplace and SLA management lifecycle. An illustration of the thesis topic is shown in Figure 6.1. As can be seen from the figure, the focus of this thesis is on two layers, which correspond to the two major challenges faced in the current cloud services market: effective SLA construction and trustworthy SLA enforcement. The inner circle is a blockchain-based decentralized cloud auction model where cloud providers/customers can conduct decentralized P2P auctions and generate SLAs using smart contracts. The outer circle focuses on a witness model in which a new role called auction witnesses is involved in the entire cloud service trading process. In our model, decentralized blockchain users can work as witnesses and join SLA monitoring through an incentive mechanism that motivates them to make truthful judgments. These two circles/layers form an organic system; the decentralized witness mechanism in the outer layer provides support for the decentralized auction in the inner layer, and their common goal is to provide an effective solution for the construction and trustworthy enforcement of cloud SLAs.

In summary, AWESOME improves the existing cloud service trading environment from two aspects: 1) optimize the selection of service provider/customer pairs and the establishment of SLAs; and 2) ensure trustworthy penalties and compensation in the cases of SLA violations. It is designed to serve the current cloud services marketplace and meet the growing demand for trusted enforcement of cloud SLAs involving multiple service providers/customers. AWESOME aims to integrate the decentralized cloud marketplace technology into a wide range of industrial use cases. The validity of the model will be further validated in several ongoing EU projects, e.g., EU ARTICONF and CLARIFY.

As mentioned in Chapter 1, the main research question of the thesis was defined as:

RQ: How to enhance the efficiency and trustworthiness of the cloud SLAs using decentralized auctions and witnesses?

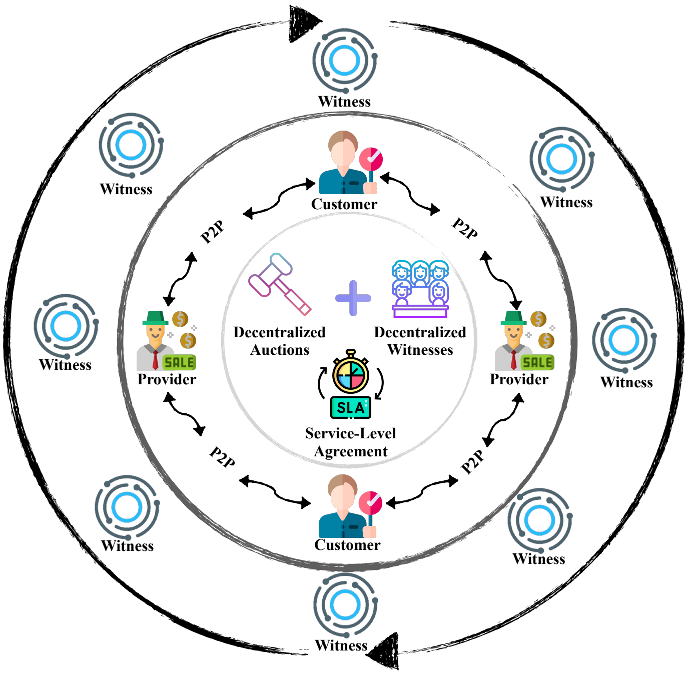


Figure 6.1: An illustration of the thesis topic: Enhancing SLAs using decentralized auctions and witnesses.

We proposed to design a blockchain-based decentralized SLA management framework, and design effective incentive mechanisms to enhance trust relationships and increase user participation. To answer this main research question, we further defined the following sub-questions:

RQ1: What are the state-of-the-art technologies and open challenges for building a decentralized service auction framework?

To answer this question, we conducted a comprehensive state-of-the-art survey on this research topic. We believe that blockchain technology can be used as a trustworthy infrastructure to build a decentralized framework for service auctions. On this basis, we summarized the trade-offs of different blockchain technologies and the challenges posed by blockchain-based auction models, e.g., auction enforcement, cost-effectiveness, privacy protection, performance & scalability, transaction ordering & fairness, front-end decentralization, cryptocurrency payment, and regulations & standards. These reviews and summaries lay the theoretical foundation for our follow-up construction of the AWESOME framework.

RQ2: How to automate the decentralized service auction and quality monitoring process in an SLA model?

To answer this question, we designed a novel AWESOME framework, which contains several advanced smart contract protocols that help automate the auction and SLA monitoring process. In this way, all participants can get the results immediately according to the established contract rules, without any intermediary involvement or time loss. We also developed a prototype AWESOME decentralized application (DApp) based on the Ethereum blockchain to facilitate user interaction with smart contracts. Finally, we presented extensive experiments to evaluate the proposed smart contracts and DApp.

RQ3: How to improve the efficiency of service auctions for managing federated clouds?

To answer this question, we designed an incentivized and cost-effective federated cloud auction model. Specifically, we first model the partition of federated cloud services as a graph partition problem. Then, the service selection is modeled as a decentralized auction based on Bayesian game theory. The derived Bayesian Nash Equilibrium (BNE) enables the selection of cost-effective providers to construct the federated cloud SLAs. Moreover, a timed message submission (TMS) algorithm is proposed to protect auction privacy on the blockchain.

RQ4: How to enhance the trustworthiness of federated SLAs in a decentralized service environment?

To answer this question, we designed an incentive mechanism for decentralized witnesses to monitor service quality. Especially, the majority decides whether the SLA is violated, and all the witnesses are motivated to participate and be honest. The monitoring process is also modeled as a Bayesian game. The derived BNE ensures consistent and trustworthy monitoring of federated SLAs. We validated the equilibrium situations of the BNE and implemented the proposed mechanism on the Ethereum blockchain.

RQ5: How to operate blockchain services to meet the scalability requirements of the AWESOME framework?

To answer this question, we conducted empirical studies to evaluate the scalability of different blockchain platforms. We first compare five popular permissioned blockchain platforms, including their scalability, stability, and resource consumption. The result shows that Hyperledger Fabric is the best choice to build the AWESOME framework since it has the optimal write throughput and acceptable read throughput. Then, we tested the performance of operating a permissioned blockchain in clouds. The results show that factors such as the VM number, VM type, bandwidth, and data centers can affect the performance of a permissioned blockchain when deployed in a cloud environment.

6.2 Lesson Learned

In this section, we will discuss some of the lessons learned while designing and developing the AWESOME framework.

6.2.1 Smart Contract Protocols

We do not let users deploy AWESOME smart contracts directly in the current model. Instead, users need to invoke the contract factory to generate auction, witness, and SLA contracts automatically. This design provides users an easy way to customize and deploy contracts while helping the DApp operator keep track of all deployed contracts. We believe such a design is reasonable and effective. Besides, the gas consumption of contracts is a huge challenge. In Section 3.4.2 we show that although the contract code has been optimized and the overall cost of the AWESOME smart contracts is economical, there are some functional interfaces such as *Place Bids* and *Calculate Witness Fee* may invoke a large amount of gas consumption. This will, to some extent, hinder the widespread utilization of the AWESOME DApp. To handle this issue, some off-chain solutions (e.g., state channels and trusted execution environments) can be leveraged to offload the on-chain computation tasks to off-chain networks.

6.2.2 Practical Issues

In Chapter 4, Bayesian games are leveraged to build the incentivized AWESOME framework and solve a practical problem (i.e., federated cloud auction). Therefore, our model is also subject to the limitations of game theory. For example, there is a basic assumption that players within the game will instinctively strive to maximize their payoffs. Bayesian games also assume that players have incomplete information about other players. However, these assumptions may be difficult to satisfy in real life since the player's decision is affected by complex factors, e.g., personal relationships and experiences. Therefore, our model may still have gaps when fully applied to reality. However, examples like the FCC auction model [155] have proved that game theory can indeed play a key role in guiding auction practices. We believe that our model could offer some new ideas for the current cloud auction research and industrial practices.

6.2.3 Security & Privacy Concerns

Blockchain technology is generally considered to be highly secure, but it may also suffer from some attacks, e.g., 51% attack and Sybil attack. Our blockchain-based AWESOME framework suffers from those attacks as well. Taking the Sybil attack as an example, any parties of the auction (provider or customer) may try to control the auction/monitoring result by registering a large number of fake bidder/witness users. In response to this issue, we set a registration threshold (e.g., a non-refundable registration fee and a minimum reputation value) in the smart contract to limit arbitrary blockchain users from joining the auction. This mechanism partly guarantees that no party is able to register many malicious accounts because such an activity requires a large amount of money. Besides, the unbiased sortition algorithm proposed in [231] can be used to select bidders/witnesses in a random and independent way, and to avoid possible unfairness or collusion.

The data stored on the blockchain must be public to all peer nodes to ensure traceability, verifiability, and immutability. This conflicts with the privacy requirements of auction users, especially for those applications with critical business secrets [163]. Our AWESOME framework will not be widely used if privacy and security are not adequately safeguarded. In general, blockchain-based auction models have two privacy concerns: identity privacy and transaction privacy (this has been discussed in Section 2.4.3). In this context, several privacy protection solutions can be used. AWESOME does not restrict users from choosing auction models and application scenarios. Therefore, users could select and implement appropriate privacy protection solutions for their specific needs in practice.

6.2.4 Blockchain Technologies

The last consideration is the trade-off between choosing permissionless and permissioned blockchain technologies to build our framework. Permissioned blockchains can address the huge operational cost and low-scalability issues of permissionless blockchains, but this is often considered at the cost of transaction security, especially in a low trust environment [15]. In Chapters 3 and 4, we used a static sealed-bid auction model to demonstrate and evaluate our AWESOME framework on the permissionless blockchain. Such an auction scenario does not require the high performance and scalability of the blockchain since each bidder only needs to submit a bid once. Besides, Ethereum's widely recognized token Ether can be commonly regarded as fiat money, which primarily motivated us to use the Ethereum blockchain to develop the current model and DApp. However, in a high-frequency and large-scale dynamic auction, the blockchain's performance is a key factor; therefore, we believe that a permissioned blockchain should be a better choice.

Chapter 7

Future Directions

In this chapter, we discuss directions that need to be explored in the future.

- *Optimizing smart contract protocols.* In order to support dynamic business requirements, we designed our AWESOME smart contracts to allow both forward and reverse auctions. In addition to the eight auction models mentioned in Section 3.3.1, many other popular auction models, e.g., double auction, combinatorial auction, and VCG auction, have demonstrated their great potential for integration with blockchain and cloud computing [180]. Therefore, we leave the implementation of these auctions in the AWESOME framework as our future work. In addition, we will continue to extend more functions/algorithms and investigate solutions that can effectively reduce contract execution costs.
- *Validating incentives in real-world use cases.* Regarding the incentive mechanism design, more game theory models, such as dynamic games and Perfect Bayesian Equilibriums (PBEs), will be considered for integration with the current model. Moreover, since game theory is a theoretical model, its effectiveness in practice needs to be further validated. In the future, we will continue to test our AWESOME framework and validate its incentives in two ongoing industrial projects, i.e., EU ARTICONF and CLARIFY.
- *Integrating security & privacy techniques.* Our blockchain-based system is inevitably subject to a number of security attacks. In the future, we plan to investigate the attack models of the AWESOME framework and the possible defense solutions that can be employed. Regarding privacy protection, we plan to integrate advanced cryptographic techniques to protect the user's identity and transaction privacy in different auction scenarios.
- *Implementing on permissioned blockchains.* We analyzed the performance of different permissioned blockchain platforms and the influencing factors in a dynamic cloud environment. It is also necessary to compare more consensus algorithms, cloud platforms, and cross-cloud deployments in the

future. In addition, since different blockchain platforms offer different contract programming protocols and languages, we have not fully implemented the AWESOME framework on those platforms. Therefore, we leave the implementation of the AWESOME framework on permissioned blockchains as our future work.

Appendix A

Literature Review of Blockchain-Based Auction Applications

This appendix provides a detailed literature review on blockchain-based auction applications.

A.1 Related Works

During the past years, auction-based theories and models have attracted extensive attention from many researchers. Most surveys on auction-related topics we can find were published before 2017 in the field of economics. Those surveys mainly concern the introduction and comparison of different auction models [117, 104, 196], market design [146], as well as the application of auctions in specialized areas such as wireless systems [226, 76] and crowdsensing [224]. The investigation efforts of blockchain, on the other hand, are relatively new. Despite the fact that blockchain is a newly emerging technology, almost every aspect of blockchain has been extensively studied in the literature. These surveys cover topics including blockchain overview [229, 26, 119], security & privacy [82, 38, 222, 127, 162], smart contract [93], consensus mechanism [202], models & tools [94], and various blockchain-based applications [27] such as healthcare [41], smart city [209], Internet of Things (IoT) [60, 125], cloud/edge computing [64, 179, 214], big data [45], and cryptocurrency [77]. Overall, both the publication number and the research diversity have increased significantly in the last few years, as shown in Figure A.1.

A Blockchain can provide a decentralized environment to support auction activities, thereby improving the security and trustworthiness of auctions. However, although there are so many studies on blockchain and auction models respectively, the combination of the two has rarely been addressed in previous survey works. The studies most relevant to our research are three survey papers working on blockchain-based energy trading solutions, where auction models are partially discussed [200, 158, 83]. The authors in these studies only focus on one specific

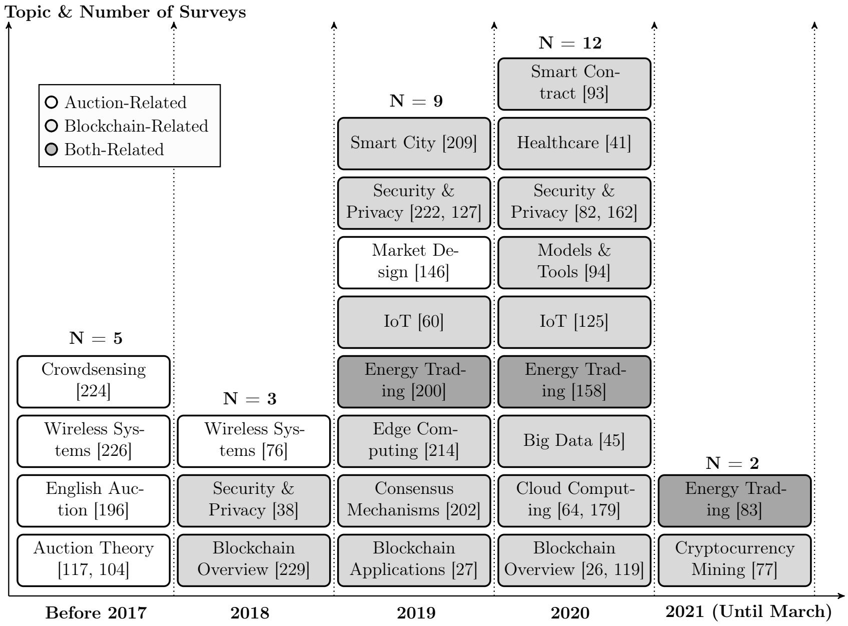


Figure A.1: Summary of existing related survey studies, categorized according to the year of publication and their focus. Here the value of N represents the number of surveys in each time interval.

application field and do not offer the comprehensiveness of this work.

In summary, most of the existing surveys discussed the two topics separately. There is no general survey on the current landscape of blockchain-based auction models. Therefore, the purpose of this survey is to summarize previous publications, and to complement existing research on blockchain-based auction models. To the best of our knowledge, this chapter is the first comprehensive survey to fill this gap.

A.2 Blockchain-Based Auction Applications

Existing surveys have indicated the huge potential of blockchain-based auction models in application fields like energy trading [83]. However, a systematic classification to categorize these applications is still lacking [27]. In this section, we propose an application-oriented taxonomy for blockchain-based auction applications, which is shown in Figure A.2. We identified and reviewed several key application fields, namely energy trading, wireless communication, service allocation, and others. Our classification method is based on a statistical analysis of existing literature and is therefore suitable to analyze current development efforts and illustrate future trends. Table A.1 further summarizes the auction models and blockchain technologies used in different studies.

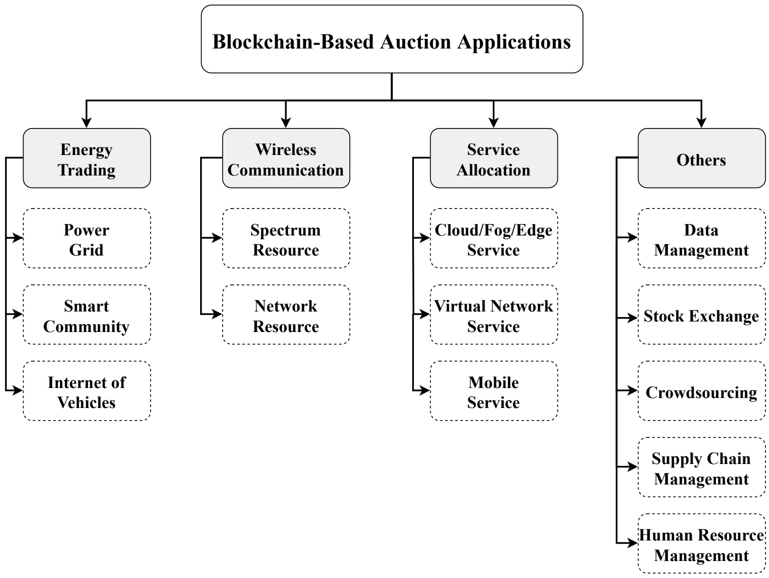


Figure A.2: Taxonomy of blockchain-based auction applications.

A.2.1 Energy Trading

Traditional centralized energy transaction models have many shortcomings, including high operating costs, low transparency, and latent risks of transaction data modification [198]. Integrating blockchain with energy trading is a new paradigm that has recently emerged. As an incentive and pricing mechanism, an auction plays a vital role in ensuring fairness and improving transaction efficiency in energy exchange. However, there are many challenges in integrating traditional energy auctions into blockchain technology. Researchers have proposed different blockchain-based auction models to address those challenges in the energy market [53]. Thematically, the relevant literature can be roughly classified into three categories: power grid, smart community, and Internet of Vehicles (IoV).

Power Grid

In traditional centralized power stations (e.g., thermal power, natural gas, and nuclear stations), consumers typically trade indirectly with energy suppliers through retailers in the market. The situation has been improved by a system named microgrid. It is a small-scale power generation and distribution system that comprises distributed power sources, electric loads, distribution facilities, and monitoring devices [140]. By promoting decentralized transactions between distributed generations (DGs) and consumers in a microgrid (instead of letting retailers act as intermediaries), the interests of both parties are increased. With the development of microgrids, transactive energy paradigms have been proposed to support the development of next-generation energy distribution systems. In this paradigm, customers might also act as suppliers rather than the one-way configuration of suppliers and consumers. Microgrid systems, in this way, allow customers to store electricity resources, sell them on-demand, and buy them from other customers [149].

In this regard, the fusion of blockchain and auction models can provide a transparent and credible trading environment for P2P microgrid energy transactions. The relevant literature has demonstrated that double auctions are more suitable for multi-seller and multi-buyer models in grid transactions. In particular, the energy distribution mechanism using double auctions eliminates the need for centralized control, which matches perfectly with the decentralized nature of blockchain. For example, Wang *et al.* [198] suggested a model for direct electricity trading between DGs and consumers in microgrids based on blockchain technology and continuous double auctions. The model aims to address the potential issues of centralized microgrid trading management, e.g., high operating costs of trading centers, trust issues between trading centers and traders, and huge information security risks. To allow dynamic adjustment of the auction bids, their model adopts an adaptive aggressiveness bidding strategy. Besides, DGs and consumers can exchange digital certificates on the blockchain to settle the auction and guarantee auction security. Yan *et al.* [213] used a similar pricing strategy, but they paid more attention to the generation right trade market. They focused on the problem of how to allocate available generation rights to integrate clean energy and reduce thermal power emissions. It should be noted that the energy payments in both of the above-mentioned studies are based on the Bitcoin cryptocurrency protocol. In addition, Thakur *et al.* [191] proposed that the information about energy surplus or deficit can be encoded as blockchain transactions and stored in an optimized Bitcoin data structure to support double auctions. They argued that blockchain performs a distributed calculation of the winner determination problem, which is more conducive to local energy trading among peers than centralized double auctions. Their simulation experiments showed that distributed double auctions facilitate energy transfer better than centralized double auctions. Stübs *et al.* [187] argued that in a smart grid network, there are multiple data communications between smart devices, edge servers and cloud servers. So a hierarchical double auction model is proposed for full on-chain implementation of energy transactions. AlAshery *et al.* [7] proposed a double auction model with an optimized VCG pricing mechanism for P2P energy trading in power grids on the blockchain. Zhao *et al.* [227] proposed a bandit learning-based double auction model that can provide participants with more auction revenues by learning the transaction history. Their simulation results showed that the bandit learning approach in a blockchain framework can provide market participants with more revenue than the way energy is traded with centralized entities.

Some traditional single-sided auction models are also presented for microgrid energy trading. Seven *et al.* [177] proposed a novel P2P energy trading scheme that uses smart contracts for virtual power plants (VPPs). In particular, the authors used an English auction-based workflow to achieve P2P transactions in a VPP. The platform is based on a public Ethereum blockchain so that it can be adapted to communications and power distributions on different networks. Hahn *et al.* [78] demonstrated how to implement Vickrey auctions on smart contracts and use them for a trading market, where multiple consumers bid for power resources from photovoltaic arrays. Energy consumers may question the fairness, trustworthiness and cyberattack resistance of centralized energy models. Therefore, the authors in [46] leveraged both Dutch and Vickrey auction models for user negotiation and power distribution. In addition, a wallet-based cryptocurrency called GreenCoin is created to support energy payments.

Blockchain-based decentralized systems bring new privacy challenges like the possible leakage of energy usage patterns [126]. So permissioned blockchains with better scalability and identity permission mechanisms are widely discussed in power grids. In this context, Zhang *et al.* [223] proposed a privacy-preserving scheme for direct power transactions in microgrids, in which a continuous double auction is combined with a permissioned blockchain to reduce costs and improve transaction privacy and efficiency. Hassan *et al.* [81] adopted a permissioned blockchain for the computation of complex on-chain transactions. They argued that the shortcomings of centralized auctioneers in terms of the trust, security, and privacy leakage are more exposed when using VCG auctions. Additionally, they leveraged the differential privacy technology to protect auction privacy. The authors in [126] proposed that transactions and bids can be de-anonymized based on network identifiers (e.g., IP addresses). Therefore, anonymity of the blockchain communication layer is crucial. This can be achieved by anonymous communication techniques such as onion routing.

Smart Community

The smart community is another blockchain-based energy auction application field that has attracted much public attention [84, 8]. In general, a community microgrid is a self-sufficient energy system designed to meet local energy needs (e.g., electricity, heating, and cooling) for communities, villages, towns, and cities. Some households may have extra renewable energy in their community microgrid and can therefore meet the needs of their neighbors. The community can flexibly absorb the peak hours of individual consumers; in this way, the energy demand of the community can be stabilized, and energy resources can be better planned. The success of a smart community heavily depends on the function of its auction economic backbone [84]. In [8], the authors proposed an auction model for energy and water resources between smart communities and smart homes, thus encouraging communities to optimize global consumption. In particular, users can use a Vickrey auction model on the blockchain network during the resource negotiation stage. Guo *et al.* [74] considered the issue of energy trading in combined cooling, heating, and power (CCHP) systems and developed a non-cooperative Stackelberg game between power grid agents and the system to model energy transactions. Their system consists of an Internet of Energy (IoE) subsystem and a blockchain subsystem, where P2P communication and energy transactions between power agents and CCHP systems can be performed efficiently and securely.

Other studies focus on improving the scalability of the blockchain to improve the performance of community energy auctions. Saxena *et al.* [173] presented a permissioned blockchain implementation of a P2P energy trading system for residential communities. In this system, a single house owner can place his/her energy bid in the district within discrete time intervals on the blockchain. A more scalable local grid system for smart communities is enerDAG [70], in which a blockchain with tangled data structures is leveraged to overcome issues such as expensive transaction fees and limited throughput. Their decentralized local energy trading platform achieves higher reliability; only a massive disruption of the communication network would cause a system collapse. However, there are still many debates regarding this blockchain since it deviates from the traditional blockchain's "chained block" data structure.

Quartierstrom [25] is a blockchain-based project for community energy trading. It is designed to manage the exchange and payment of electricity resources between consumers, producers, and local grid suppliers without any intermediaries. In Quartierstrom, a real-world prototype system has been implemented and tested in the town of Wallenstadt in Switzerland (a community with 37 families involved). The pricing mechanism of the Quartierstrom market is a double auction with discriminative pricing, while Tendermint serves as the underlying blockchain [2]. Tendermint is highly flexible and customizable to accommodate specific application requirements. It offers reduced communication, empty block creation, and customized time delays between blocks.

Internet of Vehicles

Vehicle-to-vehicle (V2V) describes a trading model in which plug-in electric vehicles (EVs) communicate with each other to exchange electricity energy. It can enhance the cooperation between vehicles, extend the driving endurance, and avoid the grid overload problem [208, 197]. However, conducting non-transparent energy transactions in IoV without trust is risky. Most existing IoV energy trading platforms and facilities are centralized, and they rely on TTPs to manage power dispatch, transaction payments, and security issues; nevertheless, these third parties are costly and can be corrupted [188]. In a blockchain-enabled decentralized IoV network, Xia *et al.* [208] argued that Bayesian games with incomplete information have significant advantages over complete information games in terms of communication overhead. Therefore, they presented a V2V electricity trading strategy using Bayesian game-based bidding and pricing. Sun *et al.* [188] further considered transaction privacy and efficiency issues. They proposed that

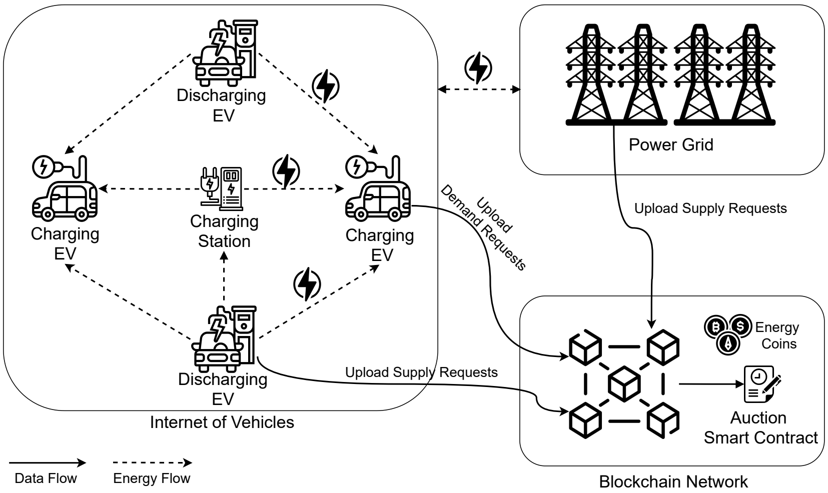


Figure A.3: An illustration of a blockchain-based energy trading model for IoV. Charging/discharging EVs and power grids upload the demand and supply requests as well as the bids/offers to the blockchain. After transactions are confirmed on the blockchain, the energy resources are traded between different entities and paid in cryptocurrencies.

centralized IoV energy trading platforms suffer from a single point of failure and lack privacy protection. In addition, power centers are inefficient in controlling large-scale and geographically distributed EVs, especially in social hotspots far from charging stations. They adopted a permissioned blockchain in the designed V2V energy trading architecture. Additionally, a novel DPoS consensus mechanism is utilized to boost trade efficiency. In [9] and [73], the authors argued that the high computational cost required in the classic permissionless blockchain is not suitable for IoV. Therefore, they adopted a blockchain with a DAG data structure for charging scheduling among EVs. Furthermore, Choubey *et al.* [37] introduced a new cryptocurrency called ETcoin to facilitate energy transactions among EVs on the permissioned blockchain.

Another related topic is vehicle-to-grid (V2G), which describes a system in which plug-in EVs communicate with the grid by returning electricity or limiting their charging rate to sell demand response services. Hassija *et al.* [85] proposed a scheme utilizing the IOTA blockchain for data sharing and energy trading in V2G networks. The scheme implements an auction-based game-theoretic approach for the price competition between EVs and grid users. Similarly, Liu *et al.* [132] developed a reverse auction-based dynamic pricing model for V2G networks in order to improve social welfare and transaction efficiency. In their model, unfilled charging EVs are powered by the smart grid, while charging and discharging transactions are executed on the smart contract. Pustišek *et al.* [165] presented a model that allows independent selection/dispatch of the most convenient charging stations for EVs in V2G networks via blockchain. Compared to traditional centralized approaches, such a solution does not require any central entity and can be fully automated, including the payment of energy. The model is implemented using the Ethereum blockchain and an FPSB auction model. To summarize, a general blockchain-based energy trading model for IoV is illustrated in Figure A.3.

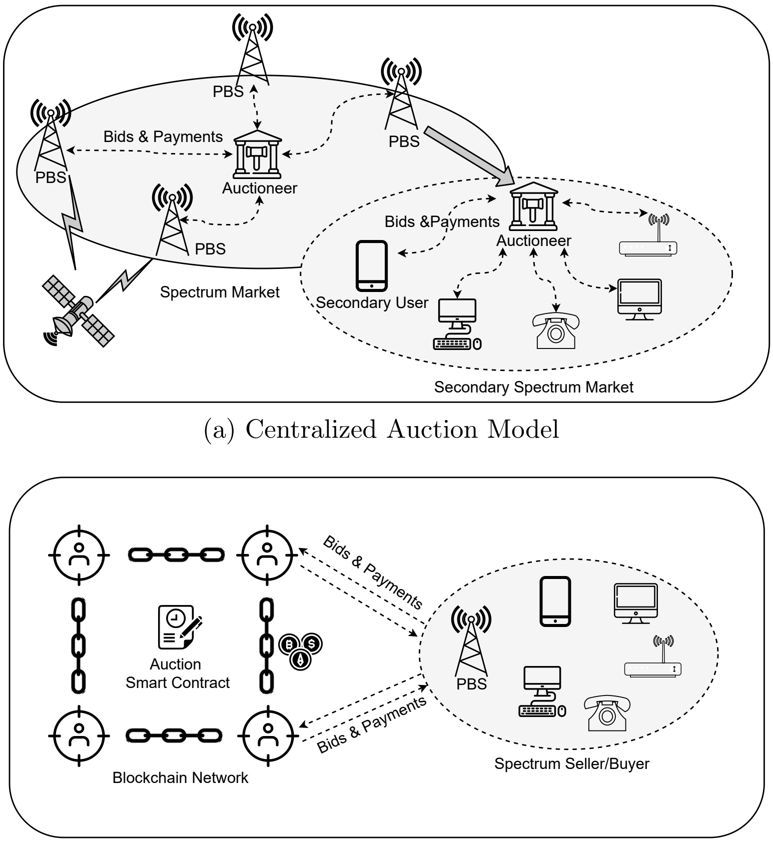
A.2.2 Wireless Communication

As wireless systems develop with new mobile communication technologies, they become increasingly complex in terms of architecture and management. Auctions have been proposed as practical mechanisms for assigning a wide range of wireless resources (e.g., spectra, subchannels, time slots, and transmit power levels). By designing and employing various auction procedures, wireless resources can be efficiently allocated between consumers and resource providers [225].

Spectrum Resource

With the rapid development of communication technology, users' demand for spectrum resources continues to increase, making spectrum a scarce resource in the trading market. However, the traditional government-led static spectrum allocation approach has failed to fully utilize the limited spectrum resources. According to the report from FCC, the utilization of the licensed spectrum can only be maintained between 15% to 85% with static spectrum allocation solutions [120]. As a result, market-driven spectrum auctions have emerged as promising solutions for spectrum allocation [199]. A spectrum auction can be centralized or decentralized, and Figure A.4 shows a comparison of the two approaches.

In this context, Fan and Huo [59] suggested a blockchain-based framework for license-free spectrum resource management in cyber-physical-social systems (CPSS). In particular, two ways of obtaining a spectrum access license (i.e., mining and auction) are designed. A new virtual currency, called Xcoin, is also introduced in this process to enhance spectrum trading. Yu *et al.* [216] focused on the space communication field and presented a spectrum auction model for heterogeneous spacecraft networks based on blockchains. They argued that the communication between different organizations in a heterogeneous spacecraft network is multi-hop compared to traditional space communication networks, which makes coordination difficult. Recent studies have further highlighted the security and privacy challenges [228]. For example, Tu *et al.* [193] designed a privacy-preserving double auction mechanism for blockchain-enabled spectrum sharing using the differential privacy technology. Wang *et al.* [199] designed a secure spectrum auction protocol that utilizes Intel Software Guard Extensions (SGX) technology and the Paillier cryptosystem. In their system, each bidder can use remote authentication to establish a secure communication channel with the SGX enclave thereby enabling the transmission and computation of sensitive data.



(a) Centralized Auction Model

(b) Decentralized Auction Model

Figure A.4: A comparison of centralized and decentralized auction models for spectrum resources. A primary base station (PBS) obtains or transfers the spectrum ownership through a centralized auction managed by an auctioneer. While in a decentralized auction, spectrum users can conduct P2P spectrum transactions on the blockchain without the need for a third-party auctioneer.

It should be noted that spectrum auctions are different from traditional auctions due to the reusable nature of spectrum resources. In most traditional auctions, the same items (e.g., artworks, antiques, and estates) can only be auctioned to a specific buyer. Spectrum auctions, by contrast, can allow the sharing of an auctioned channel as long as the buyers do not interfere with each other. In this context, dynamic spectrum management in cognitive radio (CR) networks can address the lack and underutilization of spectrum resources. CRs can be dynamically programmed and configured to use the best wireless channel nearby to avoid users interference and congestion. Based on the cognition and reconfiguration of CRs, the primary users can share their licensed spectrum with secondary users to improve spectrum utilization [228]. The authors in [122, 121] argued that the current centralized spectrum allocation is wasteful since license holders do not consistently utilize their allocated spectrum resources. They therefore introduced the idea of using blockchain as a decentralized database to verify spectrum sharing and auctions in CR networks. For secondary spectrum auctions in a CR network, an automatic pricing strategy based on a blockchain token called "spectrum dollars" is introduced in [115].

Network Resource

Network resources in wireless networks are another example. SAFE [33] is a framework designed for users to customize auction formats and allocate general wireless network resources, e.g., spectrum channels, femtocell access permissions, and resource blocks of device-to-device connections. Numerous experimental results have shown that the communication cost of SAFE is low enough so it is practical in real-life network environments. Afraz *et al.* [4] proposed a distributed resource market mechanism for future telecommunications networks, in which a double auction model and a permissioned blockchain are combined to enhance the scenarios of bilateral trading markets that exist in the telecommunications industry such as resource allocation in network functions virtualization, mobile crowd sensing, and femtocell access. Besides, cooperative relaying can be an effective way to improve the capacity, reliability, and security of wireless networks. It either helps establish communications between the source and destination or improves the established communications by adding diversity. In [113], relay operators are designed to be responsible for the relay/jammer selection and resource allocation. A double auction mechanism is used to simulate the interaction between transmitters and relay operators. Furthermore, User congestion in wireless networks is a severe problem to be solved. A Vickrey auction-based user offloading mechanism between macrocell base stations and small cell access points has been proposed in [32] to improve the capacity of heterogeneous wireless networks. Their blockchain-enabled decentralized auction solution avoids multiple malicious behaviors caused by auctioneers (third-party agents), sellers (macrocell base stations), and buyers (small cell access points).

An unmanned aerial vehicle (UAV), also known as a drone, is a newly emerging flying antenna system with a critical requirement for network resource allocation. Accordingly, a drone-mounted base station is primarily responsible for the communication between the UAV backhaul and access networks. In this field, Hassija *et al.* [87] introduced the idea of using dynamic auctions to allocate the bandwidth of drone-mounted base stations to different users to improve availability and reduce costs. They argued that communications between drone-mounted and regular base stations are vulnerable to wiretapping or man-in-the-middle attacks, so using blockchain to record the data exchange of wireless communication in a tamper-proof ledger would be a good choice. Khan *et al.* [112] proposed a multi-UAV network framework, which can: 1) outsource network coverage in specific areas based on the required service requirements; 2) enable each network entity to use the blockchain intelligently; and 3) provide an auction mechanism to make autonomous decisions. To model the interaction between UAV operators and business agents, a reputation-based truthful auction method is also presented.

A.2.3 Service Allocation

Recent developments in service computing allow the use of blockchain to allocate heterogeneous services, where blockchain can be used as decentralized auditing devices, and cryptocurrencies can secure money payments. However, most of the existing models do not provide incentives for matching service customers and providers; they often rely on manual and inefficient solutions [186]. Therefore, different auction models are proposed together with blockchain to provide secure, credible, and economical service allocation platforms.

Cloud/Fog/Edge Service

With the rapid growth of the cloud computing industry, more and more application operators are now using the cloud for service hosting, computing offloading, and data storage. Some large cloud service providers (e.g., AWS, Azure, and Google Cloud) have already supported spot instance pricing, allowing users to bid on unused capacity in cloud data centers. In this way, some users can even save up to 90% of the cost compared with the traditional on-demand instance pricing [12]. However, since cloud service providers usually sell services in a centralized and opaque manner, the fairness of the auction is challenging to guarantee in reality. A trustworthy transaction and payment mechanism is urgently needed to motivate service providers/customers and improve service utilization. AStERISK [186] is a framework designed to fill this gap; it automatically determines the best price for cloud services and assigns customers to the most appropriate providers by implementing sealed-bid auctions on the blockchain. Similarly, Chen *et al.* [34] introduced a blockchain-based auction and trading model for cloud virtual machine allocation. Their model can achieve fairness in auction transactions by implementing commitment-based state mechanisms, smart contracts, and cryptocurrency technologies. In [71, 72], the authors paid attention to the cloud storage problem and proposed VCG auction-based resource trading models for distributed cloud storage. This is based on the context that traditional storage resource trading systems typically operate in a centralized model, leading to high costs, vendor lock-in and single point of failure risks.

The paradigm of connecting things to the cloud to receive a centralized service is not always the best option, which leads to the context in which edge and fog computing are widely discussed. Basically, they both intend to distribute the computing capacity and assist the cloud server with additional resources located near the end users [233]. In this respect, DeCloud [219] is a secure and decentralized auction system specifically built for open edge computing infrastructures. It integrates a truthful double auction and a bidding language to match highly heterogeneous edge resources with different service requests. Compared to recently proposed decentralized cloud/fog solutions such as iExec, Sonm and Golem, DeCloud is more focused on designing an effective marketplace for decentralized open infrastructures. Debe *et al.* [44] demonstrated a blockchain-based reverse auction solution for public fog service allocation. Yu *et al.* [215] also leveraged the reverse auction model and presented a blockchain-based edge crowdsourcing service system. Specifically, a changeable auction algorithm is designed so that each request from the user will find a winner that can provide the appropriate edge service. CloudAgora [48] is a platform that enables low-cost cloud storage and computing access based on blockchain and auctions, where prices are determined through an auction game. ChainFaaS [67] is designed to run serverless tasks using the computing power of personal computers.

Virtual Network Service

Network functions virtualization (NFV) has come into view for its ability to provide multiple network functions at a low cost [80]. The traditional NFV marketplace relies on third-party companies for the provisioning, distribution, and execution of NFV resources. BRAIN [62] is a blockchain-based reverse auction solution with a focus on NFV scenarios. It is introduced to address the challenge of discovering and selecting infrastructures that can efficiently host NFV services based on specific user needs. Virtual network embedding (VNE) is one of the most important problems in network virtualization and is responsible for mapping virtual networks to underlying physical networks. Many auction methods have been presented in the literature to achieve efficient resource allocation in VNE. Rizk *et al.* [169] argued that although a centralized VNE approach demonstrates high efficiency in slice allocation, it suffers from scalability issues since everything depends on one virtual network provider. Therefore, they designed a decentralized VNE system that uses smart contracts and a Vickrey auction model for trustworthy virtual network partitioning and allocation.

Mobile Service

The number of mobile devices and compute-intensive mobile applications has exploded in recent decades. The focus of these mobile applications is to improve the quality of service (QoS) for end users; however, by improving the QoS, these applications generate a large amount of mobile traffic, thus posing a huge challenge to mobile network providers. One of the most promising ways to deal with this issue is mobile data offloading. For example, Hassija *et al.* [89] created a mobile data offloading model in which mobile devices and users can securely perform computation offloading services on the blockchain. The simulation results showed that their model achieves low communication costs and optimized scheduling performance compared to other offloading schemes. FlopCoin [29] is a virtual currency specially designed for compensating mobile devices when they execute device-to-device offloading services.

On the other hand, the widespread dissemination of programmable sensor-employed smartphones has facilitated mobile crowdsensing applications such as environmental monitoring, crowd journalism, and public safety. These applications require effective incentives to compensate and reward mobile users for their resource contributions. Chatzopoulos *et al.* [30] suggested the use of blockchain and smart contracts to manage spatial crowdsensing interactions between mobile service providers and customers. A truthful and cost-optimal auction model is also designed on the blockchain to reduce payments from crowdsensing providers to mobile users. Their experimental results showed that the time overhead of using blockchain in short-term crowdsourcing tasks is negligible compared to centralized server solutions.

A.2.4 Others

Data Management

The uncertainty of data value makes it difficult to make accurate estimates of the appropriate price for data. An auction is a powerful approach to protect the interests of both data sellers and buyers while maintaining the fundamental principles of the marketplace. To eliminate systemic risks caused by collusion in large-scale data auctions, the authors in [211] introduced a decentralized data auction system that uses an anti-collusion auction algorithm executed on the smart contract. The system ensures that buyers and sellers can engage in data auctions without relying on TTPs. An *et al.* [13] implemented a crowdsourcing data trading system using blockchain and reverse auctions. They used carefully designed smart contracts to replace third-party data brokers, thus providing a trustworthy environment for data sellers and consumers. Besides, a permissioned blockchain-based model is used in [31] to enable secure and efficient IoV data transactions. An iterative double auction model is also presented to optimize data pricing and improve data transaction volume.

Stock Exchange

A stock exchange is a marketplace where traders can buy and sell securities, e.g., stocks, bonds, options. Traditional stock markets are performed in a centralized manner. This structure ensures the authenticity and security of transactions, but is vulnerable to attacks and lack of transparency in the trading process. To address the single point of failure in centralized stock exchange platforms, Al-Shaibani *et al.* [178] introduced a permissioned blockchain-based decentralized stock exchange platform. Similarly, Pop *et al.* [164] suggested addressing the shortcomings of centralized stock trading to reduce transaction costs caused by brokers and central institutions. An Ethereum-based decentralized Bucharest stock exchange model is further proposed and validated. Their experimental results indicated that for partially filled order books, the blockchain-based solution has a significant price advantage compared to the centralized solution. Recently, dark pool trading, as an anonymous and decentralized stock trading approach, has become an increasingly important component of traditional stock exchanges. The decentralized and secure transaction properties of blockchain are well suited to provide support for anonymous dark pool transactions. AuditChain [195] is an auditing and record-keeping platform for financial markets using blockchain. In particular, a periodic double auction-based dark pool use case is used to demonstrate the platform's feasibility for stock trading. When a private corporation wants to raise capital by issuing new stocks, it can issue shares to the public by conducting an initial public offering (IPO). Purchasers usually acquire multiple shares from the seller at the same price in an IPO, which is a typical example of a uniform price auction. In [79], the authors introduced a uniform price auction model for IPOs on the permissioned blockchain. They designed an additional communication chaincode to provide applications with limited access to P2P APIs in the built-in communication layer. The model further leverages secure multi-party computation technology to protect the privacy of IPO transactions.

Crowdsourcing

Crowdsourcing is a specific business model for acquiring resources in which an individual or organization can leverage a large number of users to obtain desired services. Traditional centralized crowdsourcing platforms face many challenges, including motivating workers to share their truthful costs and guaranteeing trusted interactions among users and the platform. To cope with those challenges, ABCrowd [109] is a fully decentralized crowdsourcing framework that implements a repeated single-minded VCG auction mechanism on the blockchain. BitFund [88] is a platform designed to connect developers and investors in the global crowdfunding environment, where a novel ascending-price progressive auction algorithm is implemented for cost-effective task allocation.

Supply Chain Management

In a supply chain, decentralized auctions can be widely used to coordinate transactions between suppliers and consumers. BitCom [75] is a decentralized supply chain model built on the blockchain to provide a clean and efficient trading environment. Martins *et al.* [141] proposed a customer-driven supply chain marketplace on the blockchain, where customers post their proposals and suppliers strive to outbid each other in a reverse auction model. Similarly, Koirala *et al.* [118] introduced a solution to improve transparency and traceability in the carrier procurement process. Their solution considers multiple attributes of carriers in the supply chain during the reverse auction bidding process. The traditional English auction model has also been found in the literature. In [183], an online English auction system is implemented to sell and buy food products using the Ethereum blockchain.

Human Resource Management

Employment and labor industries become more and more important since the value of human resources is directly related to a company's profitability. However, employee background check remains a controversial field in HR operations, particularly in the cases of employment, education, and skills verification [135]. E^{2}C-Chain [135] [134] is a two-stage blockchain designed to assist the improvement of human resource management. In the first stage, the employees' background records can be stored in the blockchain in an immutable manner. After that, a VCG auction mechanism is leveraged to encourage verifiers to join in the skill verification of employees. Another application field is employee recognition program, where employers reward employees for their achievements, milestones, and anniversaries [176]. In such a context, Ward *et al.* [203] argued that employees could liquidate their unwanted gifts to others through auction mechanisms. Blockchain and smart contract technologies can be used in this process of matching individuals for exchanging gifts.

We also identified individual applications in blockchain-based auction models, e.g., federated learning (FL), IoT collaboration, and code ownership management. For instance, a centralized aggregator is usually needed to maintain and update the global state in a traditional FL model. BAFFLE [167] is a decentralized framework for non-aggregator FL. It uses smart contracts to coordinate FL tasks and a user scoring and bidding mechanism to reach the FL goal. For FL in edge computing, Fan *et al.* [58] proposed a resource trading system using a hybrid blockchain. Their main idea is to establish a transparent, decentralized, and high-performance trading platform that can encourage more edge nodes to join in the FL model training. Another interesting topic is collaborative IoT. As IoT projects become more and more complex, IoT managers, experts, and non-technical staff are expected to collaborate in the IoT development cycle [168]. In [35], a novel blockchain-based reverse auction model is proposed to prompt active cooperation among IoT participants. Besides, the current centralized code ownership management scheme is cumbersome and opaque. Therefore, a blockchain-based approach for managing code ownership is proposed in [175], where auctions are used for ubiquitous code allocation.

A.2.5 Key Observations

The key observations we obtained in this appendix are summarized as follows:

- Blockchain-based decentralized auctions offer great potential to optimize the traditional centralized auction model, which is particularly reflected by different application scenarios. Different researchers have used different auction models and blockchain technologies to handle auctions for specific application scenarios. These applications exist mainly in energy trading, wireless communication, and service allocation.

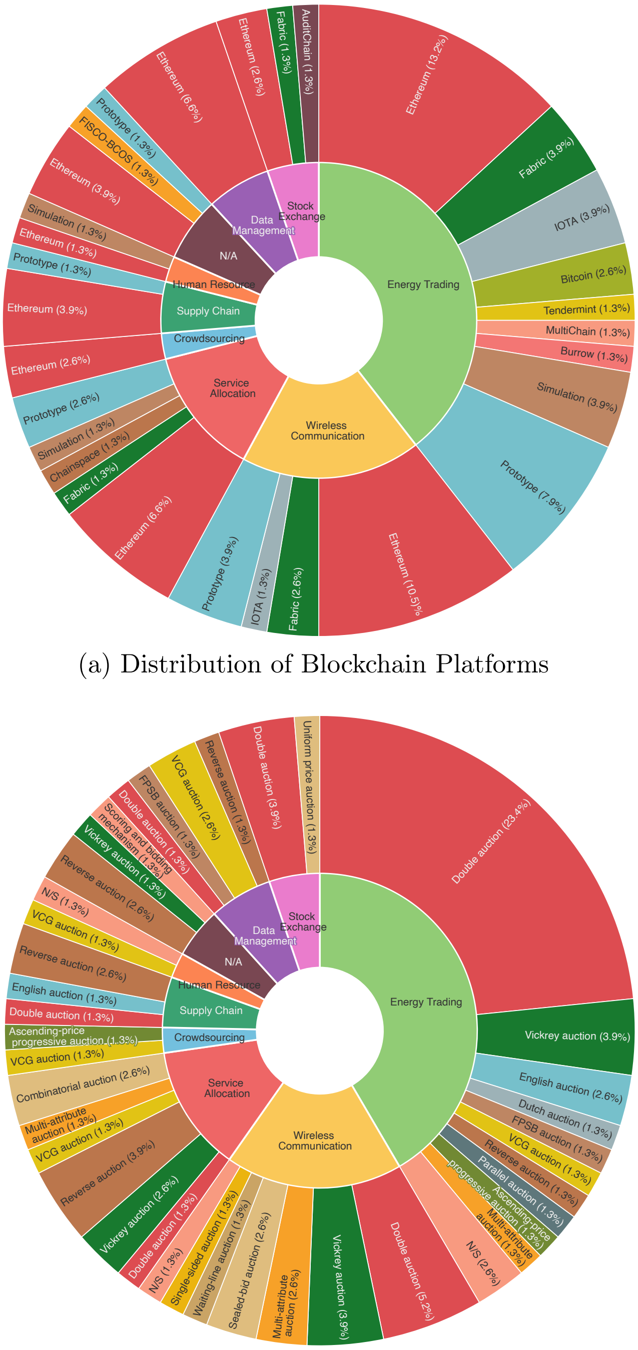
- The centralized auction model has long been the dominant trading model in energy trading. However, the development of traditional centralized energy markets has gradually encountered bottlenecks. For example, the performance of energy trading is highly dependent on the servers and networks of centralized third-party platforms in the traditional model. It is therefore vulnerable to single points of failure. In addition, centralized auction management leads to high operational costs, low transparency, and the potential risk of tampering with energy transaction data. Finally, centralized long-distance energy transmission makes the power supply vulnerable to disruptions [200]. In contrast, decentralized P2P energy trading is a more desirable solution in modern power systems to improve efficiency and stability.
- Efficient allocation of scarce network resources has always been a hot research topic in wireless communications. Although resource sharing architectures using both centralized and decentralized auctions can improve resource utilization, the security issue of conducting transactions between untrusted entities is severe in a centralized model. In addition, most traditional solutions can only maintain a single specific auction format and lack a common framework that can accommodate a variety of auction formats. Automation of business processes is becoming increasingly critical as it facilitates dynamic utilization of network resources.
- In terms of service allocation, the traditional centralized approach to service trading suffers from several weaknesses. For example, most existing cloud auction solutions have vendor lock-in issues, where the vendor acts as an auctioneer. In such cases, auction fairness is difficult to guarantee because large cloud providers can abuse their dominant market position, forcing users to trust their services and adapt to the rules and prices. In addition, some service providers and customers may collude with third-party auctioneers to learn about users' bids and use that knowledge to gain more profit or exit the market in time.
- All of the above issues are driving the application of blockchain-based decentralized auctions as a future trend. Overall, blockchain as an enabling technology in the transition from centralized to decentralized auctions provides the following advantages: 1) Decentralized trust management. Blockchain provides a decentralized, transparent, and trustworthy auction trading environment. Such a design does not require a centralized auctioneer and optimizes the design and operation of the trading platform; 2) Secure, private, and cost-effective transaction. Compared to traditional centralized

auctions, blockchain-based auctions can achieve the trust requirements of auction participants at a much lower cost; 3) Tokenized auction payment. Blockchain has a cryptocurrency market with a broad user base to support auctions, and some application-specific tokens are designed to be used for specific auctions; 4) Customizable auction format. With the powerful programmability provided by smart contracts, almost any auction format can be programmed to meet specific application and business requirements; and 5) Automated auction execution. Smart contracts can help automate the auction process so that all participants can immediately get results according to established rules without any intermediary involvement or loss of time.

To get an overview of how blockchain and auction models are integrated, we summarized the auction models and blockchain technologies used in different studies, as shown in Table A.1. In general, although different blockchain technologies have their trade-offs, researchers tend to have specific selection requirements and preferences when actually performing the model construction. We find that the largest share of studies (34.7%) adopt permissioned blockchain technologies. It is widely believed that the access control mechanism in the permissioned blockchain can protect business secrets better. In addition, the high throughput of permissioned blockchains can accommodate large-scale transactions, making them more suitable for real industrial applications. Slightly fewer studies (28.0%) use permissionless blockchains. In these studies, researchers argue that the fully decentralized nature of permissionless blockchains can make the auction platform more trustworthy, and the built-in cryptocurrency can directly support transactions within the blockchain platform. We find that only three studies choose the hybrid blockchain. Although cross-chain solutions have been proposed for several years, they have rarely been studied in auction applications. Nevertheless, we believe this could be a promising direction for future research. Finally, in one-third of the studies, no specific blockchain technology was determined. Those authors leave the choice of implementing blockchain technologies to users.

Figure A.5a further illustrates the distribution of blockchain platforms used in different auction application fields. Our finding is that more than half of the studies use Ethereum as the underlying blockchain infrastructure. Apart from energy trading, Ethereum is also the most popular blockchain platform in all application fields. Some researchers argue that microtransactions in P2P energy trading require high system throughput, so it is more favorable to implement a permissioned blockchain (e.g., Hyperledger Fabric) or DAG blockchain (e.g., IOTA) platform. In addition, we notice that a small number of authors do not choose established commercial blockchains; instead, they use simulation tools (e.g., Python or Matlab) to validate their models or frameworks. Other studies only present the conceptual proof of their blockchain-based auction models without on-chain implementations.

As shown in Figure A.5b, the most commonly used auction models are double



(a) Distribution of Blockchain Platforms

(b) Distribution of Auction Models

Figure A.5: The distribution of blockchain technologies and auction models in existing studies regarding different application fields. The results are obtained by quantitative statistics based on their number of appearances in the literature. Details of the auction model and blockchain technology used in each paper are listed in the Table A.1.

auction (36.4%), reverse auction (11.7%), Vickrey auction (11.7%), and VCG auction (7.8%). We notice that double auctions are most frequently used in energy trading and stock exchange. This is mainly because the energy trading and stock exchange markets with multiple sellers and multiple buyers are well suited to integrate with double auctions. Among other application fields, most researchers prefer traditional single-sided auctions (e.g., reverse, Vickrey, and VCG auctions). Another interesting finding is that reverse auctions are popular in service allocation and supply chain management. This is mainly because the reverse auction can bring substantial cost savings to buyers in those two application fields. A reverse auction also helps streamline the auction process; auction time is saved because buyers do not need to send requests to different sellers one by one.

Table A.1: Summary of Blockchain-Based Auction Applications

Application Field		**Ref.**	**Year**	**Addressed Issue**	**Auction Model**	**Blockchain Type**	**Blockchain Platform**
Energy Trading	**Power Grid**	[198]	2017	Microgrids energy trading	Continuous double auction	Permissionless	Bitcoin
		[213]	2018	Generation right trading	Continuous double auction	Permissioned	MultiChain
		[191]	2018	Microgrids energy trading	Double auction	Permissionless	Bitcoin
		[187]	2020	Smart grids energy trading	Hierarchical double auction	Permissionless	Ethereum
		[7]	2020	Smart grids energy trading	Double auction	Permissioned	Simulation
		[227]	2019	Transactive energy trading	Double auction with bandit learning	N/S	Ethereum
		[177]	2020	Energy trading in virtual power plants	English auction	Permissionless	Ethereum
		[78]	2017	Transactive energy trading	Vickrey auction	N/S	Ethereum
		[46]	2019	Smart power distribution	Dutch auction & Vickrey auction	N/S	Ethereum
		[126]	2017	Transactive energy trading	N/S	Permissionless	Prototype
		[223]	2019	Microgrids energy trading	Continuous double auction	Permissioned	Simulation
		[81]	2020	Microgrids energy trading	Modified VCG auction	Permissioned	Prototype
		[53]	2017	Decentralized energy trading market	Short-term parallel auction	Permissioned	Hyperledger Burrow
		[149]	2020	Microgrids energy trading	English auction & Continuous double auction	N/S	Ethereum
		[61]	2019	Decentralized energy trading market	Uniform-Price double auction	Permissioned	Ethereum
	Smart Community	[84]	2019	Smart communities energy trading	Double auction	Hybrid	Prototype
		[8]	2018	Smart communities energy trading	Vickrey auction	Permissioned	Ethereum
		[74]	2020	Energy trading in CCHP systems	N/S	Permissionless	Prototype
		[173]	2019	Residential communities energy trading	Periodic double auction	Permissioned	Hyperledger Fabric
		[70]	2020	Local energy trading market	Double auction	Permissionless	IOTA
		[25] [2]	2019	Local energy trading market	Double auction	Permissioned	Tendermint
	Internet of Vehicles	[208]	2020	V2V energy trading	Double auction	Permissioned	Hyperledger Fabric
		[197]	2020	EV group energy trading	Double auction	N/S	Prototype
		[188]	2020	V2V energy trading	Iterative double auction	Permissioned	Simulation
		[9]	2020	Energy trading in IoV	Multi-attribute auction	Permissionless	IOTA
		[73]	2020	EV charging scheduling	Constrained double auction	Permissionless	Prototype
		[37]	2019	V2V energy trading	Double auction	Permissioned	Hyperledger Fabric
		[85]	2020	V2G data sharing and energy trading	Ascending-price progressive auction	Permissionless	IOTA
		[132]	2019	V2G energy trading	Reverse sealed-bid auction	Permissioned	Ethereum
		[165]	2016	V2G charging scheduling	FPSB auction	N/S	Ethereum
Wireless Communication	**Radio Spectrum**	[199]	2020	Spectrum resource allocation	Single-sided auction	N/S	Ethereum
		[59]	2020	Spectrum resource management in CPSS	N/S	Permissioned	Prototype
		[228]	2020	Multiple-operators spectrum sharing	Double auction	Permissioned	Ethereum
		[193]	2020	Dynamic spectrum sharing	Double auction	Permissioned	Ethereum
		[122] [121]	2017 2018	Spectrum sharing in CR networks	Waiting-line auction	Permissionless	Prototype
		[115]	2020	Secondary spectrum trading market	Periodic sealed-bid auction	N/S	Prototype
		[216]	2019	Spectrum allocation in spacecraft networks	Generalized Vickrey auction	Permissionless	Ethereum
	Network Resource	[33]	2020	Wireless network resource allocation	General sealed-bid auction	Permissionless	Ethereum
		[4]	2019	Trade market for telecommunication networks	Double auction	Permissioned	Hyperledger Fabric

Continued on next page

Continued from previous page

Application Field		Ref.	Year	Addressed Issue	Auction Model	Blockchain Type	Blockchain Platform
		[113]	2019	Ccooperative relaying resource allocation	Double auction	Permissionless	Ethereum
		[32]	2020	User offloading in wireless networks	Vickrey auction	N/S	Ethereum
		[87]	2020	Bandwidth allocation for UAV base stations	Multi-attribute auction	Permissioned	Ethereum
		[112]	2020	UAV network resource allocation	Vickrey auction	Permissioned	Hyperledger Fabric
		[86]	2020	Bandwidth allocation between EVs and roadside units	Multi-attribute auction	Permissionless	IOTA
Service Allocation	**Cloud/ Fog/ Edge Service**	[186]	2019	Shared economy service allocation	Vickrey auction	Permissionless	Chainspace
		[34]	2020	Cloud VM allocation	Combinatorial auction	N/S	Ethereum
		[71]	2018	Cloud data storage resource trading	VCG auction	N/S	Ethereum
		[72]	2018	Distributed data storage	Reverse VCG auction	N/S	Ethereum
		[219]	2019	Edge/Cloud service trading	Double auction	N/S	Prototype
		[44]	2020	Fog service trading	Reverse auction	Permissionless	Ethereum
		[215]	2019	Edge service crowdsensing	Reverse auction	N/S	Prototype
		[129]	2020	Service allocation in fog-enabled IoV	VCG auction	Permissioned	Hyperledger Fabric
	Network Service	[62]	2019	Virtual network services in NFV markets	Reverse FPSB auction	Permissionless	Ethereum
		[169]	2018	Brokerless virtual network embedding	Vickrey auction	Permissioned	Ethereum
	Mobile Service	[89]	2020	Mobile data offloading	Multi-attribute auction	Permissionless	Simulation
		[30]	2018	Mobile service crowdsensing	Combinatorial auction	N/S	Ethereum
Others	**Data Management**	[211]	2020	Big data trading and auction	FPSB auction	N/S	Ethereum
		[13]	2019	Crowdsensed data trading	Reverse auction	N/S	Ethereum
		[31]	2019	Data trading in IoV	Iterative double auction	Permissioned	Ethereum
	Stock Exchange	[178]	2020	Decentralized stock exchange	Double auction	Permissioned	Ethereum
		[164]	2018	Decentralized stock exchange	Double auction	N/S	Ethereum
		[195]	2020	Financial trade auditing	Periodic double auction	Permissioned	AuditChain
		[79]	2019	Secure and efficient IPOs	Sealed-bid uniform price auction	Permissioned	Hyperledger Fabric
	Crowdsourcing	[109]	2020	Decentralized spatial crowdsourcing	Optimized VCG auction	N/S	Ethereum
		[88]	2020	Decentralized crowdfunding platform	Ascending-price progressive auction	N/S	Ethereum
	Supply Chain	[75]	2020	Decentralized supply chain management	Double auction	Hybrid	Prototype
		[141]	2020	Customer bargaining and e-procurement	Reverse auction	Permissionless	Ethereum
		[118]	2019	Multi-attribute carrier procurement	Reverse auction	N/S	Ethereum
		[183]	2021	Food supply chain management	English auction	N/S	Ethereum
	Human Resource	[135] [134]	2019	Education and employment verification	VCG auction	N/S	Simulation
		[203]	2018	Employee recognition programs reward	N/S	N/S	Ethereum
	N/A	[167]	2020	Decentralized federated learning	Scoring and bidding mechanism	N/S	Ethereum
		[58]	2020	Federated learning resource trading	Reverse auction	Hybrid	Ethereum & FISCO-BCOS
		[35]	2020	IoT collaboration	Reverse auction	Permissioned	Prototype
		[175]	2018	Code ownership management system	Vickrey auction	Permissionless	Ethereum

Bibliography

[1] Erik-Oliver Blass and Florian Kerschbaum. "BOREALIS: Building Block for Sealed Bid Auctions on Blockchains". In: *Proc. ACM ASIACCS*. Taipei, Oct. 2020, pages 558–571 (cited on page 29).

[2] Liliane Ableitner, Arne Meeuw, Sandro Schopfer, Verena Tiefenbeck, Felix Wortmann, and Anselma Wörner. *Quartierstrom – Implementation of a real world prosumer centric local energy market in Walenstadt, Switzerland*. arXiv:1905.07242. 2019 (cited on pages 126, 140).

[3] AERGO. *What is AERGO?* Accessed: Oct. 15, 2021. URL: `https://www.aergo.io/aergo/` (cited on page 22).

[4] N. Afraz and M. Ruffini. "A Distributed Bilateral Resource Market Mechanism for Future Telecommunications Networks". In: *Proc. IEEE GLOBECOM Workshop*. Hawaii, USA, Dec. 2019, pages 1–6 (cited on pages 130, 140).

[5] Alireza Afshari, Majid Mojahed, and Rosnah Mohd Yusuff. "Simple additive weighting approach to personnel selection problem". In: *Int. J. Innov. Technol. Manag* 1.5 (2010), page 511 (cited on page 67).

[6] Victor Ahlqvist, Pär Holmberg, and Thomas Tangerås. "A survey comparing centralized and decentralized electricity markets". In: *Energy Strateg. Rev.* 40 (2022), page 100812 (cited on page 23).

[7] Mohamed Kareem AlAshery, Zhehan Yi, Di Shi, Xiao Lu, Chunlei Xu, Zhiwei Wang, and Wei Qiao. "A Blockchain-Enabled Multi-Settlement Quasi-Ideal Peer-to-Peer Trading Framework". In: *IEEE Trans. Smart Grid* 12.1 (2020), pages 885–896 (cited on pages 124, 140).

[8] Ramon Alcarria, Borja Bordel, Tomás Robles, Diego Martín, and Miguel-Ángel Manso-Callejo. "A Blockchain-Based Authorization System for Trustworthy Resource Monitoring and Trading in Smart Communities". In: *Sensors* 18.10 (2018), page 3561 (cited on pages 125, 140).

[9] M. Ali, A. Anjum, A. Anjum, and M. A. Khan. "Efficient and Secure Energy Trading in Internet of Electric Vehicles Using IOTA Blockchain". In: *Proc. IEEE HONET*. Charlotte, USA, Dec. 2020, pages 87–91 (cited on pages 127, 140).

[10] Alibaba Open Source. *Alibaba Cluster Trace Program*. Accessed: Mar. 01, 2022. URL: `https://github.com/alibaba/clusterdata` (cited on page 79).

[11] Sarah Allen et al. *Design Choices for Central Bank Digital Currency: Policy and Technical Considerations*. Accessed: Aug. 28, 2021. URL: `https://www.nber.org/system/files/working_papers/w27634/w27634.pdf` (cited on page 28).

[12] Amazon Web Services Inc. *Amazon EC2 Spot Instances*. Accessed: Aug. 22, 2021. URL: `https://aws.amazon.com/ec2/spot/` (cited on page 131).

[13] Baoyi An, Mingjun Xiao, An Liu, Guoju Gao, and Hui Zhao. "Truthful Crowdsensed Data Trading Based on Reverse Auction and Blockchain". In: *Database Systems for Advanced Applications*. Springer, 2019, pages 292–309 (cited on pages 133, 141).

[14] M. Babaioff and N. Nisan. "Concurrent Auctions Across The Supply Chain". In: *J. Artif. Intell. Res.* 21 (2004), pages 595–629 (cited on page 15).

[15] Yannis Bakos and Hanna Halaburda. *Tradeoffs in Permissioned vs Permissionless Blockchains: Trust and Performance*. Available at SSRN 3789425. 2021 (cited on page 118).

[16] Imran Bashir. *Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained*. Packt Publishing Ltd, 2018 (cited on pages 2, 12).

[17] Michael R Baye, Dan Kovenock, and Casper G De Vries. "The all-pay auction with complete information". In: *Econ. Theory* 8.2 (1996), pages 291–305 (cited on page 16).

[18] F. Benhamouda, S. Halevi, and T. Halevi. "Supporting Private Data on Hyperledger Fabric with Secure Multiparty Computation". In: *IBM J. Res. Dev.* 63.2/3 (2019), 3:1–3:8 (cited on page 29).

[19] Matthias Berberich and Malgorzata Steiner. "Blockchain technology and the GDPR-how to reconcile privacy and distributed ledgers". In: *Eur. Data Prot. L. Rev.* 2 (2016), page 422 (cited on page 32).

[20] Shishir Bharathi, Ann Chervenak, Ewa Deelman, Gaurang Mehta, Mei-Hui Su, and Karan Vahi. "Characterization of scientific workflows". In: *Proc. Int. WORKS Workshop.* Austin, USA, Nov. 2008, pages 1–10 (cited on page 79).

[21] Martin Bichler. "An experimental analysis of multi-attribute auctions". In: *Decis. Support Syst.* 29.3 (2000), pages 249–268 (cited on page 16).

[22] Matthew Black, TingWei Liu, and Tony Cai. *Atomic Loans: Cryptocurrency Debt Instruments.* arXiv:1901.05117. 2019 (cited on page 32).

[23] Erik-Oliver Blass and Florian Kerschbaum. "Strain: A Secure Auction for Blockchains". In: *Proc. ESORICS.* Barcelona, Spain, Sept. 2018, pages 87–110 (cited on page 29).

[24] Chiara Braghin, Stelvio Cimato, Ernesto Damiani, and Michael Baronchelli. "Designing smart-contract based auctions". In: *Proc. Int. Conf. SICBS.* Guilin, China, Dec. 2018, pages 54–64 (cited on pages 3, 24, 27).

[25] Alain Brenzikofer and Noa Melchior. *Privacy-Preserving P2P Energy Market on the Blockchain.* arXiv:1905.07940. 2019 (cited on pages 126, 140).

[26] Bert-Jan Butijn, Damian A Tamburri, and Willem-Jan van den Heuvel. "Blockchains: A systematic multivocal literature review". In: *ACM Comput. Surv.* 53.3 (2020), pages 1–37 (cited on pages 121, 122).

[27] Fran Casino, Thomas K. Dasaklis, and Constantinos Patsakis. "A systematic literature review of blockchain-based applications: Current status, classification and open issues". In: *Telemat. Inform.* 36 (2019), pages 55–81 (cited on pages 3, 121, 122).

[28] Chainlink. *What Is the Blockchain Oracle Problem?* Accessed: Aug. 24, 2021. URL: `https://blog.chain.link/what-is-the-blockchain-oracle-problem/` (cited on page 25).

[29] D. Chatzopoulos, M. Ahmadi, S. Kosta, and P. Hui. "FlopCoin: A Cryptocurrency for Computation Offloading". In: *IEEE Trans. Mobile Comput.* 17.5 (2018), pages 1062–1075 (cited on page 132).

[30] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui. "Privacy Preserving and Cost Optimal Mobile Crowdsensing Using Smart Contracts on Blockchain". In: *Proc. IEEE MASS.* Chengdu, China, Oct. 2018, pages 442–450 (cited on pages 133, 141).

[31] C. Chen, J. Wu, H. Lin, W. Chen, and Z. Zheng. "A Secure and Efficient Blockchain-Based Data Trading Approach for Internet of Vehicles". In: *IEEE Trans. Veh. Technol.* 68.9 (2019), pages 9110–9121 (cited on pages 133, 141).

[32] T. Chen, A. S. Khan, G. Zheng, and S. Lambotharan. "Blockchain Secured Auction-Based User Offloading in Heterogeneous Wireless Networks". In: *IEEE Wireless Commun. Lett.* 9.8 (2020), pages 1141–1145 (cited on pages 130, 141).

[33] Y. Chen, X. Tian, Q. Wang, J. Jiang, M. Li, and Q. Zhang. "SAFE: A General Secure and Fair Auction Framework for Wireless Markets with Privacy Preservation". In: *IEEE Trans. Depend. Sec. Comput.* (2020). Early Access (cited on pages 130, 140).

[34] Zhili Chen, Wei Ding, Yan Xu, Miaomiao Tian, and Hong Zhong. *Fair Auction and Trade Framework for Cloud VM Allocation based on Blockchain.* arXiv:2001.00771. 2020 (cited on pages 36, 37, 131, 141).

[35] G. Cheng, S. Deng, Z. Xiang, Y. Chen, and J. Yin. "An Auction-Based Incentive Mechanism with Blockchain for IoT Collaboration". In: *Proc. IEEE ICWS.* Beijing, China, Oct. 2020, pages 17–26 (cited on pages 135, 141).

[36] Tony Chew. *The 8 Challenges to Overcome to Enable Cryptocurrency Payments.* Accessed: Sep. 02, 2021. URL: `https://medium.com/aditusnetwork/8-challenges-to-overcome-to-enable-cryptocurrency-payments-c7a49e379d61` (cited on page 32).

[37] A. Choubey, S. Behera, Y. S. Patel, K. Mahidhar, and R. Misra. "EnergyTradingRank Algorithm for Truthful Auctions among EVs via Blockchain Analytics of Large Scale Transaction Graphs". In: *Proc. Int. Conf. COMSNETS.* Bangalore, India, Jan. 2019, pages 1–6 (cited on pages 127, 140).

[38] Mauro Conti, E Sandeep Kumar, Chhagan Lal, and Sushmita Ruj. "A survey on security and privacy issues of Bitcoin". In: *IEEE Commun. Surveys Tuts.* 20.4 (2018), pages 3416–3452 (cited on pages 121, 122).

[39] Li Da Xu and Wattana Viriyasitavat. "Application of blockchain in collaborative Internet-of-Things services". In: *IEEE Trans. Comput. Social Syst.* 6.6 (2019), pages 1295–1305 (cited on page 12).

[40] Sankarshan Damle, Boi Faltings, and Sujit Gujar. *A Practical Solution to Yao's Millionaires' Problem and Its Application in Designing Secure Combinatorial Auction.* arXiv:1906.06567. 2019 (cited on page 29).

[41] Erikson Júlio De Aguiar, Bruno S Faiçal, Bhaskar Krishnamachari, and Jó Ueyama. "A survey of blockchain-based strategies for healthcare". In: *ACM Comput. Surv.* 53.2 (2020), pages 1–27 (cited on pages 121, 122).

[42] Stefano De Angelis, Leonardo Aniello, Roberto Baldoni, Federico Lombardi, Andrea Margheri, and Vladimiro Sassone. "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain". In: *Proc. ITASEC.* Milan, Italy, Feb. 2018, page 11 (cited on page 20).

[43] Sven De Vries and Rakesh V Vohra. "Combinatorial auctions: A survey". In: *INFORMS J. Comput.* 15.3 (2003), pages 284–309 (cited on page 14).

[44] M. Debe, K. Salah, M. H. U. Rehman, and D. Svetinovic. "Blockchain-Based Decentralized Reverse Bidding in Fog Computing". In: *IEEE Access* 8 (2020), pages 81686–81697 (cited on pages 36, 37, 132, 141).

[45] Natarajan Deepa, Quoc-Viet Pham, Dinh C. Nguyen, Sweta Bhattacharya, B Prabadevi, Thippa Reddy Gadekallu, Praveen Kumar Reddy Maddikunta, Fang Fang, and Pubudu N. Pathirana. *A Survey on Blockchain for Big Data: Approaches, Opportunities, and Future Directions.* arXiv:2009.00858. 2020 (cited on pages 121, 122).

[46] S. Dekhane, K. Mhalgi, K. Vishwanath, S. Singh, and N. Giri. "Greencoin: Empowering Smart Cities Using Blockchain 2.0". In: *Proc. ICNTE.* Navi Mumbai, India, Jan. 2019, pages 1–5 (cited on pages 32, 125, 140).

[47] H. Desai, M. Kantarcioglu, and L. Kagal. "A Hybrid Blockchain Architecture for Privacy-Enabled and Accountable Auctions". In: *Proc. IEEE Blockchain.* Atlanta, USA, July 2019, pages 34–43 (cited on page 29).

[48] Katerina Doka, Tasos Bakogiannis, Ioannis Mytilinis, and Georgios Goumas. "CloudAgora: Democratizing the cloud". In: *Proc. IEEE Blockchain.* Atlanta, USA, July 2019, pages 142–156 (cited on pages 36, 37, 132).

[49] David Easley, Jon Kleinberg, et al. *Networks, Crowds, and Markets.* Cambridge University Press, 2010 (cited on pages 13, 14).

[50] eBay. *Selling fees.* Accessed: Mar. 01, 2022. URL: `https://www.ebay.com/help/selling/fees-credits-invoices/selling-fees?id=4822` (cited on page 89).

[51] eBid. *Fees and charges.* Accessed: Mar. 01, 2022. URL: `https://www.ebid.net/us/help/fees-and-charges/` (cited on page 89).

[52] Mark Emem. *Andy Warhol's Multi-Million Dollar Painting Tokenized and Sold on Blockchain.* Accessed: Aug. 22, 2021. URL: `https://finance.yahoo.com/news/andy-warhol-multi-million-dollar-162928721.html` (cited on page 12).

[53] Ayman Esmat, Martijn de Vos, Yashar Ghiassi-Farrokhfal, Peter Palensky, and Dick Epema. "A novel decentralized platform for peer-to-peer energy trading market with blockchain technology". In: *Appl. Energy* 282 (2021), page 116123 (cited on pages 123, 140).

[54] Ethereum. *Proof-Of-Stake (POS).* Accessed: Jun. 18, 2022. URL: `https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/` (cited on page 20).

[55] Ethereum Community. *Ethereum Development Standards.* Accessed: Sep. 03, 2021. URL: `https://ethereum.org/en/developers/docs/standards/` (cited on page 33).

[56] Ethereum Community. *Gas and Fees.* Accessed: Aug. 26, 2021. URL: `https://ethereum.org/en/developers/docs/gas/` (cited on page 27).

[57] Ethereum Community. *Oracle Services.* Accessed: Aug. 26, 2021. URL: `https://ethereum.org/en/developers/docs/oracles/` (cited on pages 27, 36).

[58] Sizheng Fan, Hongbo Zhang, Yuchen Zeng, and Wei Cai. "Hybrid Blockchain-Based Resource Trading System for Federated Learning in Edge Computing". In: *IEEE Internet Things J.* 8.4 (2020), pages 2252–2264 (cited on pages 135, 141).

[59] X. Fan and Y. Huo. "Blockchain Based Dynamic Spectrum Access of Non-Real-Time Data in Cyber-Physical-Social Systems". In: *IEEE Access* 8 (2020), pages 64486–64498 (cited on pages 32, 128, 140).

[60] Mohamed Amine Ferrag, Makhlouf Derdour, Mithun Mukherjee, Abdelouahid Derhab, Leandros Maglaras, and Helge Janicke. "Blockchain technologies for the internet of things: Research issues and challenges". In: *IEEE Internet Things J.* 6.2 (2018), pages 2188–2204 (cited on pages 121, 122).

[61] Magda Foti and Manolis Vavalis. "Blockchain based uniform price double auctions for energy markets". In: *Appl. Energy* 254 (2019), page 113604 (cited on page 140).

[62] Muriel Figueredo Franco, Eder John Scheid, Lisandro Zambenedetti Granville, and Burkhard Stiller. "BRAIN: Blockchain-based reverse auction for infrastructure supply in virtual network functions-as-a-service". In: *Proc. IFIP Networking.* Warsaw, Poland, May 2019, pages 1–9 (cited on pages 30, 32, 132, 141).

[63] Daniel Friedman. "The double auction market institution: A survey". In: *The Double Auction Market Institutions, Theories, and Evidence.* Routledge, 2018, pages 3–26 (cited on page 15).

[64] Keke Gai, Jinnan Guo, Liehuang Zhu, and Shui Yu. "Blockchain meets cloud computing: A survey". In: *IEEE Commun. Surveys Tuts.* 22.3 (2020), pages 2009–2030 (cited on pages 121, 122).

[65] Hisham S Galal and Amr M Youssef. "Verifiable sealed-bid auction on the Ethereum blockchain". In: *Proc. Int. Conf. FC.* Nieuwpoort, Curaçao, Feb. 2018, pages 265–278 (cited on page 29).

[66] Hisham S Galal and Amr M Youssef. "Succinctly verifiable sealed-bid auction smart contract". In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology.* Springer, 2018, pages 3–19 (cited on page 29).

[67] Sara Ghaemi, Hamzeh Khazaei, and Petr Musilek. "ChainFaas: An open blockchain-based serverless platform". In: *IEEE Access* 8 (2020), pages 131760–131778 (cited on pages 36, 37, 132).

[68] Golem. *Golem: A Decentralized Cloud Computing Network.* Accessed: Apr. 01, 2022. 2022. URL: `https://www.golem.network/` (cited on pages 36, 37).

[69] Christian Gorenflo, Stephen Lee, Lukasz Golab, and Srinivasan Keshav. "FastFabric: Scaling hyperledger fabric to 20 000 transactions per second". In: *Int. J. Netw. Manag.* 30.5 (2020), e2099 (cited on page 30).

[70] C. Groß, M. Schwed, S. Mueller, and O. Bringmann. "enerDAG – Towards a DLT-Based Local Energy Trading Platform". In: *Proc. IEEE COINS.* Barcelona, Spain, Aug. 2020, pages 1–8 (cited on pages 126, 140).

[71] Yonggen Gu, Dingding Hou, and Xiaohong Wu. “A Cloud Storage Resource Transaction Mechanism Based on Smart Contract”. In: *Proc. ICCNS*. Qingdao, China, Nov. 2018, pages 134–138 (cited on pages 131, 141).

[72] Yonggen Gu, Dingding Hou, Xiaohong Wu, Jie Tao, and Yanqiong Zhang. “Decentralized Transaction Mechanism Based on Smart Contract in Distributed Data Storage”. In: *Information* 9.11 (2018), page 286 (cited on pages 36, 37, 131, 141).

[73] Jianxiong Guo, Xingjian Ding, and Weili Wu. *A Double Auction for Charging Scheduling among Vehicles Using DAG-Blockchains*. arXiv:2010.01436. 2020 (cited on pages 127, 140).

[74] Jianxiong Guo, Xingjian Ding, and Weili Wu. *Combined Cooling, Heating, and Power System in Blockchain-Enabled Energy Management*. arXiv:2003.13416. 2020 (cited on pages 125, 140).

[75] S. Gupta, H. Sharma, V. Hassija, and V. Saxena. “BitCom: A Commerce Model on Blockchain”. In: *Proc. IEEE ICSC*. San Diego, USA, Mar. 2020, pages 64–70 (cited on pages 134, 141).

[76] Ummy Habiba and Ekram Hossain. “Auction mechanisms for virtualization in 5G cellular networks: Basics, trends, and open challenges”. In: *IEEE Commun. Surveys Tuts.* 20.3 (2018), pages 2264–2293 (cited on pages 13, 121, 122).

[77] Umit Hacioglu, Dounia Chlyeh, Mustafa K Yilmaz, Ekrem Tatoglu, and Dursun Delen. “Crafting performance-based cryptocurrency mining strategies using a hybrid analytics approach”. In: *Decis. Support Syst.* 142 (2021), page 113473 (cited on pages 121, 122).

[78] A. Hahn, R. Singh, C. Liu, and S. Chen. “Smart Contract-Based Campus Demonstration of Decentralized Transactive Energy Auctions”. In: *Proc. IEEE ISGT*. Washington, USA, Apr. 2017, pages 1–5 (cited on pages 125, 140).

[79] Tzipora Halevi, Fabrice Benhamouda, Angelo De Caro, Shai Halevi, Charanjit Jutla, Yacov Manevich, and Qi Zhang. “Initial Public Offering (IPO) on Permissioned Blockchain Using Secure Multiparty Computation”. In: *Proc. IEEE Blockchain*. Atlanta, USA, July 2019, pages 91–98 (cited on pages 134, 141).

[80] Pengchao Han, Lei Guo, and Yejun Liu. “Virtual network embedding in SDN/NFV based fiber-wireless access network”. In: *Proc. ICSN*. Jeju Island, Republic of Korea, May 2016, pages 1–5 (cited on page 132).

[81] M. U. Hassan, M. H. Rehmani, and J. Chen. “DEAL: Differentially Private Auction for Blockchain-Based Microgrids Energy Trading”. In: *IEEE Trans. Serv. Comput.* 13.2 (2020), pages 263–275 (cited on pages 125, 140).

[82] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. “Differential privacy in blockchain technology: A futuristic approach”. In: *J. Parallel Distrib. Comput.* 145 (2020), pages 50–74 (cited on pages 121, 122).

[83] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. *Optimizing Blockchain Based Smart Grid Auctions: A Green Revolution*. arXiv:2102.02583. 2021 (cited on pages 6, 121, 122).

[84] V. Hassija, G. Bansal, V. Chamola, V. Saxena, and B. Sikdar. “BlockCom: A Blockchain Based Commerce Model for Smart Communities using Auction Mechanism”. In: *Proc. IEEE ICC Workshop*. Shanghai, China, May 2019, pages 1–6 (cited on pages 125, 140).

[85] V. Hassija, V. Chamola, S. Garg, D. N. G. Krishna, G. Kaddoum, and D. N. K. Jayakody. “A Blockchain-Based Framework for Lightweight Data Sharing and Energy Trading in V2G Network”. In: *IEEE Trans. Veh. Technol.* 69.6 (2020), pages 5799–5812 (cited on pages 127, 140).

[86] V. Hassija, V. Chamola, V. Gupta, and G. S. S. Chalapathi. “A Framework for Secure Vehicular Network using Advanced Blockchain”. In: *Proc. Int. Conf. IWCMC*. Limassol, Cyprus, June 2020, pages 1260–1265 (cited on page 141).

[87] V. Hassija, V. Saxena, and V. Chamola. “A Blockchain-based Framework for Drone-Mounted Base Stations in Tactile Internet Environment”. In: *Proc. IEEE INFOCOM Workshop*. Beijing, China, July 2020, pages 261–266 (cited on pages 130, 141).

[88] Vikas Hassija, Vinay Chamola, and Sherali Zeadally. “Bitfund: A Blockchain-Based Crowd Funding Platform for Future Smart and Connected Nation”. In: *Sust. Cities Soc.* 60 (2020), page 102145 (cited on pages 134, 141).

[89] Vikas Hassija, Vikas Saxena, and Vinay Chamola. “A mobile data offloading framework based on a combination of blockchain and virtual voting”. In: *Softw.-Pract. Exp.* (2020). Early Access (cited on pages 132, 141).

[90] Florian Hawlitschek, Benedikt Notheisen, and Timm Teubner. “The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy”. In: *Electron. Commer. Res. Appl.* 29 (2018), pages 50–63 (cited on page 24).

[91] Alexandra Henzinger, Alexander Noe, and Christian Schulz. “ILP-based local search for graph partitioning”. In: *J. Exp. Algorithmics* 25 (2020), pages 1–26 (cited on page 67).

[92] Heidi Howard and Richard Mortier. “Paxos vs Raft: Have we reached consensus on distributed consensus?” In: *Proc. PaPoC*. Apr. 2020, pages 1–9 (cited on page 21).

[93] Bin Hu, Zongyang Zhang, Jianwei Liu, Yizhong Liu, Jiayuan Yin, Rongxing Lu, and Xiaodong Lin. “A comprehensive survey on smart contract construction and execution: Paradigms, tools, and systems”. In: *Patterns* 2.2 (2021), page 100179 (cited on pages 121, 122).

[94] Huawei Huang, Wei Kong, Sicong Zhou, Zibin Zheng, and Song Guo. “A survey of state-of-the-art on blockchains: Theories, modelings, and tools”. In: *ACM Comput. Surv.* 54.2 (2021), pages 1–42 (cited on pages 121, 122).

[95] Hyperledger. *Hyperledger Architecture.* Accessed: Jun. 20, 2022. URL: `https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf` (cited on page 21).

[96] Hyperledger Fabric. *Chaincode Tutorials.* Accessed: Aug. 26, 2021. URL: `https://hyperledger-fabric.readthedocs.io/en/release-1.1/chaincode.html` (cited on page 19).

[97] Hyperledger Performance and Scale Working Group. *Hyperledger Blockchain Performance Metrics White Paper.* Accessed: Jul. 01, 2022. URL: `https://www.hyperledger.org/resources/publications/blockchain-performance-metrics` (cited on page 105).

[98] Hyperledger Sawtooth. *About Dynamic Consensus.* Accessed: Jul. 01, 2022. URL: `https://sawtooth.hyperledger.org/docs/1.2/sysadmin_guide/about_dynamic_consensus.html` (cited on page 103).

[99] Hyperledger Sawtooth. *Frequently-asked Questions.* Accessed: Jul. 01, 2022. URL: `https://sawtooth.hyperledger.org/faq/` (cited on page 107).

[100] Hyperledger Sawtooth. *Introduction.* Accessed: Jul. 01, 2022. URL: `https://sawtooth.hyperledger.org/docs/core/releases/latest/introduction.html` (cited on page 103).

[101] Hyperledger Sawtooth. *Transaction Processor: Creating a Transaction Handler.* Accessed: Aug. 30, 2021. URL: `https://sawtooth.hyperledger.org/docs/core/releases/1.1/_autogen/sdk_TP_tutorial_js.html` (cited on page 19).

[102] Shadi Ibrahim, Bingsheng He, and Hai Jin. “Towards pay-as-you-consume cloud computing”. In: *Proc. IEEE SCC.* Washington, USA, July 2011, pages 370–377 (cited on page 1).

[103] iExec. *iExec: Blockchain-Based Decentralized Cloud Computing.* Accessed: Apr. 01, 2022. 2022. URL: `https://iex.ec/` (cited on pages 36, 37).

[104] Anurag Jain and Riyaz Sikora. *A Classification of Auction Mechanism: Potentiality for Multi-Agent System (MAS) based Modeling.* Accessed: Oct. 01, 2021. URL: `http://www.swdsi.org/swdsi06/Proceedings06/Papers/MIS15.pdf` (cited on pages 121, 122).

[105] Bernard J Jansen and Tracy Mullen. “Sponsored search: an overview of the concept, history, and technology”. In: *Int. J. Electron. Bus.* 6.2 (2008), pages 114–131 (cited on page 16).

[106] Sandy D Jap. “The impact of online reverse auction design on buyer–supplier relationships”. In: *J. Mark.* 71.1 (2007), pages 146–159 (cited on page 15).

[107] Keith Jeferry, George Kousiouris, Dimosthenis Kyriazis, Jörn Altmann, Augusto Ciuffoletti, Ilias Maglogiannis, Paolo Nesi, Bojan Suzic, and Zhiming Zhao. “Challenges emerging from future cloud application scenarios”. In: *Procedia Comput. Sci.* 68 (2015), pages 227–237 (cited on page 102).

[108] Naman Kabra, Pronaya Bhattacharya, Sudeep Tanwar, and Sudhanshu Tyagi. “MudraChain: Blockchain-based framework for automated cheque clearance in financial institutions”. In: *Futur. Gener. Comp. Syst.* 102 (2020), pages 574–587 (cited on page 3).

[109] M. Kadadha, R. Mizouni, S. Singh, H. Otrok, and A. Ouali. “ABCrowd: An Auction Mechanism on Blockchain for Spatial Crowdsourcing”. In: *IEEE Access* 8 (2020), pages 12745–12757 (cited on pages 134, 141).

[110] Maha Kadadha, Hadi Otrok, Rabeb Mizouni, Shakti Singh, and Anis Ouali. “SenseChain: A blockchain-based crowdsensing framework for multiple requesters and multiple workers”. In: *Futur. Gener. Comp. Syst.* 105 (2020), pages 650–664 (cited on page 3).

[111] George Karypis and Vipin Kumar. *METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices.* Accessed: Feb. 01, 2022. URL: `http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/manual.pdf` (cited on page 79).

[112] A. S. Khan, G. Chen, Y. Rahulamathavan, G. Zheng, B. Assadhan, and S. Lambotharan. “Trusted UAV Network Coverage Using Blockchain, Machine Learning, and Auction Mechanisms”. In: *IEEE Access* 8 (2020), pages 118219–118234 (cited on pages 130, 141).

[113] A. S. Khan, Y. Rahulamathavan, B. Basutli, G. Zheng, B. Assadhan, and S. Lambotharan. “Blockchain-Based Distributive Auction for Relay-Assisted Secure Communications”. In: *IEEE Access* 7 (2019), pages 95555–95568 (cited on pages 130, 141).

[114] Kashif Mehboob Khan, Junaid Arshad, and Muhammad Mubashir Khan. “Investigating performance constraints for blockchain based secure e-voting system”. In: *Futur. Gener. Comp. Syst.* 105 (2020), pages 13–26 (cited on page 3).

[115] Mubbashar A. Khan, Mohsin M. Jamali, Taras Maksymyuk, and Juraj Gazda. “A Blockchain Token-Based Trading Model for Secondary Spectrum Markets in Future Generation Mobile Networks”. In: *Wirel. Commun. Mob. Comput.* 2020 (2020), pages 1–12 (cited on pages 130, 140).

[116] Paul Klemperer. “Auction theory: A guide to the literature”. In: *J. Econ. Surv.* 13.3 (1999), pages 227–286 (cited on pages 12, 14, 45).

[117] Paul Klemperer. *Auctions: Theory and Practice.* Princeton University Press, 2004 (cited on pages 1, 13, 121, 122).

[118] R. C. Koirala, K. Dahal, S. Matalonga, and R. Rijal. “A Supply Chain Model with Blockchain-Enabled Reverse Auction Bidding Process for Transparency and Efficiency”. In: *Proc. Int. Conf. SKIMA.* Island of Ulkulhas, Maldives, Aug. 2019, pages 1–6 (cited on pages 134, 141).

[119] John Kolb, Moustafa AbdelBaky, Randy H Katz, and David E Culler. "Core concepts, challenges, and future directions in blockchain: A centralized tutorial". In: *ACM Comput. Surv.* 53.1 (2020), pages 1–39 (cited on pages 19, 121, 122).

[120] Paul Kolodzy. *Spectrum Policy Task Force.* Accessed: Oct. 03, 2021. URL: `https://docs.fcc.gov/public/attachments/DOC-228542A1.pdf` (cited on page 128).

[121] K. Kotobi and S. G. Bilen. "Secure Blockchains for Dynamic Spectrum Access: A Decentralized Database in Moving Cognitive Radio Networks Enhances Security and User Access". In: *IEEE Veh. Technol. Mag.* 13.1 (2018), pages 32–39 (cited on pages 129, 140).

[122] K. Kotobi and S. G. Bilén. "Blockchain-Enabled Spectrum Access in Cognitive Radio Networks". In: *Proc. WTS.* Chicago, USA, Apr. 2017, pages 1–6 (cited on pages 129, 140).

[123] Vijay Krishna. *Auction Theory.* Academic Press, 2009 (cited on pages 12, 14).

[124] Pascal Lafourcade, Mike Nopère, and Daniela Pizzuti. *Auctionity Yellow Paper.* Accessed: Oct. 04, 2021. URL: `https://www.auctionity.com/wp-content/uploads/2018/09/Auctionity-Yellow-Paper.pdf` (cited on page 27).

[125] Laphou Lao, Zecheng Li, Songlin Hou, Bin Xiao, Songtao Guo, and Yuanyuan Yang. "A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling". In: *ACM Comput. Surv.* 53.1 (2020), pages 1–32 (cited on pages 121, 122).

[126] Aron Laszka, Abhishek Dubey, Michael Walker, and Doug Schmidt. "Providing Privacy, Safety, and Security in IOT-Based Transactive Energy Systems Using Distributed Ledgers". In: *Proc. Int. Conf. IoT.* Linz, Austria, Oct. 2017, pages 1–8 (cited on pages 125, 140).

[127] Jae Hyung Lee et al. "Systematic approach to analyzing security and vulnerabilities of blockchain systems". Master's thesis. Massachusetts Institute of Technology, 2019 (cited on pages 121, 122).

[128] Jung-San Lee, Chit-Jie Chew, Ying-Chin Chen, and Kuo-Jui Wei. "Preserving Liberty and Fairness in Combinatorial Double Auction Games Based on Blockchain". In: *IEEE Syst. J.* (2020). Early Access (cited on page 15).

[129] Yunseong Lee, Seohyeon Jeong, Arooj Masood, Laihyuk Park, Nhu-Ngoc Dao, and Sungrae Cho. "Trustful Resource Management for Service Allocation in Fog-Enabled Intelligent Transportation Systems". In: *IEEE Access* 8 (2020), pages 147313–147322 (cited on page 141).

[130] Ao Lei, Yue Cao, Shihan Bao, Dasen Li, Philip Asuquo, Haitham Cruickshank, and Zhili Sun. "A blockchain based certificate revocation scheme for vehicular communication systems". In: *Futur. Gener. Comp. Syst.* 110 (2020), pages 892–903 (cited on page 3).

[131] Ying Lin. *10 eBay Statistics You Need to Know.* Accessed: Aug. 22, 2021. URL: `https://www.oberlo.com/blog/ebay-statistics` (cited on page 29).

[132] H. Liu, Y. Zhang, S. Zheng, and Y. Li. "Electric Vehicle Power Trading Mechanism Based on Blockchain and Smart Contract in V2G Network". In: *IEEE Access* 7 (2019), pages 160546–160558 (cited on pages 127, 140).

[133] Lietong Liu, Mingxiao Du, and Xiaofeng Ma. "Blockchain-Based Fair and Secure Electronic Double Auction Protocol". In: *IEEE Intell. Syst.* 35.3 (2020), pages 31–40 (cited on page 29).

[134] Liyuan Liu, Meng Han, Yiyun Zhou, and Reza M. Parizi. "E^{2}C-Chain: A Two-Stage Incentive Education Employment and Skill Certification Blockchain". In: *Proc. IEEE Blockchain.* Atlanta, USA, July 2019, pages 140–147 (cited on pages 135, 141).

[135] Liyuan Liu, Meng Han, Yiyun Zhou, Reza M. Parizi, and Mohammed Korayem. "Blockchain-Based Certification for Education, Employment, and Skill with Incentive Mechanism". In: *Blockchain Cybersecurity, Trust and Privacy.* Springer, 2020, pages 269–290 (cited on pages 135, 141).

[136] Yue Liu, Qinghua Lu, Xiwei Xu, Liming Zhu, and Haonan Yao. "Applying design patterns in smart contracts". In: *Proc. IEEE Blockchain.* Halifax, Canada, Aug. 2018, pages 92–106 (cited on page 46).

[137] Jie Ma, Bin Qi, and Kewei Lv. "Fully Private Auctions for the Highest Bid". In: *Proc. ACM TURC.* Chengdu, China, May 2019, pages 1–6 (cited on page 29).

[138] Lodovica Marchesi, Michele Marchesi, and Roberto Tonelli. *ABCDE – Agile Block Chain Dapp Engineering.* Available at arXiv 1912.09074. 2019 (cited on page 40).

[139] MarketsandMarkets Inc. *Blockchain Market.* Accessed: Aug. 26, 2021. URL: `https://www.marketsandmarkets.com/Market-Reports/blockchain-technology-market-90100890.html` (cited on pages 12, 19).

[140] Chris Marnay, Spyros Chatzivasileiadis, Chad Abbey, Reza Iravani, Geza Joos, Pio Lombardi, Pierluigi Mancarella, and Jan von Appen. "Microgrid evolution roadmap". In: *Proc. Int. Symp. EDST.* Vienna, Austria, Sept. 2015, pages 139–144 (cited on page 123).

[141] 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". In: *IEEE Trans. Eng. Manag.* (2020). Early Access, pages 1–15 (cited on pages 134, 141).

[142] Jacob Maslow. *What Is a Crypto Payment Gateway?* Accessed: Oct. 21, 2021. URL: `https://www.influencive.com/what-is-a-crypto-payment-gateway/` (cited on page 32).

[143] Hitoshi Matsushima and Shunya Noda. *Mechanism Design with Blockchain Enforcement.* Accessed: Oct. 01, 2021. URL: `http://www.cirje.e.u-tokyo.ac.jp/research/dp/2020/2020cf1145.pdf` (cited on page 27).

[144] McAfee. *How Bad is the eBay Breach?* Accessed: Jun. 21, 2022. URL: `https://www.mcafee.com/blogs/enterprise/cloud-security/how-bad-is-the-ebay-breach-here-are-the-stats/` (cited on page 24).

[145] Alison McGuire. *Hybrid blockchain: The best of both chains.* Accessed: Oct. 16, 2021. URL: `https://irishtechnews.ie/hybrid-blockchain-the-best-of-both-chains/` (cited on page 22).

[146] Paul Milgrom. "Auction Market Design: Recent Innovations". In: *Annu. Rev. Econ.* 11.1 (2019), pages 383–405 (cited on pages 12, 121, 122).

[147] Paul Milgrom. "Putting auction theory to work: The simultaneous ascending auction". In: *J. Polit. Econ.* 108.2 (2000), pages 245–272 (cited on page 12).

[148] Roman Mühlberger, Stefan Bachhofner, Eduardo Castelló Ferrer, Claudio Di Ciccio, Ingo Weber, Maximilian Wöhrer, and Uwe Zdun. "Foundational Oracle Patterns: Connecting Blockchain to the Off-chain World". In: *Proc. Int. Conf. BPM.* Seville, Spain, Sept. 2020, pages 35–51 (cited on page 3).

[149] Sein Myung and Jong-Hyouk Lee. "Ethereum smart contract-based automated power trading algorithm in a microgrid environment". In: *J. Supercomput.* 76.7 (2020), pages 4904–4914 (cited on pages 124, 140).

[150] Satoshi Nakamoto. *Bitcoin: A Peer-To-Peer Electronic Cash System.* Accessed: Oct. 04, 2021. URL: `https://bitcoin.org/bitcoin.pdf` (cited on pages 2, 12).

[151] Henri Neuendorf. *Christie's Will Become the First Major Auction House to Use Blockchain in a Sale.* Accessed: Aug. 29, 2021. URL: `https://news.artnet.com/market/christies-artory-blockchain-pilot-1370788` (cited on page 12).

[152] Truc D. T. Nguyen and My T. Thai. *A Blockchain-based Iterative Double Auction Protocol using Multiparty State Channels.* arXiv:2007.08595. 2020 (cited on pages 12, 27).

[153] Chris Nickson. *Auction House Commissions.* Accessed: Aug. 26, 2021. URL: `http://www.exploreauctions.co.uk/AuctionHouseCommissions.html` (cited on page 24).

[154] Nobel Media AB. *The Prize in Economic Sciences 2020.* Accessed: Aug. 26, 2021. URL: `https://www.nobelprize.org/prizes/economic-sciences/2020/summary/` (cited on page 12).

[155] Nobel Media AB. *The quest for the perfect auction.* Accessed: Apr. 01, 2022. 2020. URL: `https://www.nobelprize.org/prizes/economic-sciences/2020/popular-information/` (cited on page 117).

[156] Clare O'Gara. *2 Waves of DDoS Attacks Stop Rare Spirits Auction.* Accessed: Jun. 21, 2022. URL: `https://www.secureworld.io/industry-news/2-waves-of-ddos-attacks-stop-rare-spirits-auction` (cited on page 31).

[157] Ilhaam A Omar, Haya R Hasan, Raja Jayaraman, Khaled Salah, and Mohammed Omar. "Implementing decentralized auctions using blockchain smart contracts". In: *Technol. Forecast. Soc. Chang.* 168 (2021), page 120786 (cited on page 27).

[158] Simona-Vasilica Oprea and Adela Bâra. "Local market mechanisms survey for peer-to-peer electricity trading on blockchain platform". In: *Sci. Bull. "Mircea cel Bătrân" Nav. Acad.* 23.1 (2020), 186A–191 (cited on pages 6, 121, 122).

[159] Ariel Orda and Ori Rottenstreich. "Enforcing fairness in blockchain transaction ordering". In: *Peer Peer Netw. Appl.* 14.6 (2021), pages 3660–3673 (cited on page 31).

[160] Pankesh Patel, Ajith H Ranabahu, and Amit P Sheth. *Service level agreement in cloud computing.* Accessed: Nov. 15, 2021. 2009. URL: `https://corescholar.libraries.wright.edu/knoesis/78/` (cited on page 2).

[161] Pegasus. *Workflow gallery.* Accessed: Mar. 01, 2022. URL: `https://pegasus.isi.edu/workflow_gallery/index.php` (cited on page 89).

[162] Li Peng, Wei Feng, Zheng Yan, Yafeng Li, Xiaokang Zhou, and Shohei Shimizu. "Privacy preservation in permissionless blockchain: A survey". In: *Digit. Commun. Netw.* (2020). Early Access (cited on pages 121, 122).

[163] Li Peng, Wei Feng, Zheng Yan, Yafeng Li, Xiaokang Zhou, and Shohei Shimizu. "Privacy preservation in permissionless blockchain: A survey". In: *Digit. Commun. Netw* 7.3 (2021), pages 295–307 (cited on page 118).

[164] C. Pop, C. Pop, A. Marcel, A. Vesa, T. Petrican, T. Cioara, I. Anghel, and I. Salomie. "Decentralizing the Stock Exchange Using Blockchain an Ethereum-Based Implementation of the Bucharest Stock Exchange". In: *Proc. IEEE ICCP.* Cluj-Napoca, Romania, Sept. 2018, pages 459–466 (cited on pages 133, 141).

[165] M. Pustišek, A. Kos, and U. Sedlar. "Blockchain Based Autonomous Selection of Electric Vehicle Charging Station". In: *Proc. Int. Conf. IIKI.* Beijing, China, Oct. 2016, pages 217–222 (cited on pages 127, 140).

[166] R3. *Consensus on Corda.* Accessed: Jun. 21, 2022. URL: `https://docs.r3.com/en/platform/corda/4.8/open-source/key-concepts-consensus.html` (cited on page 21).

[167] Paritosh Ramanan and Kiyoshi Nakayama. "Baffle: Blockchain Based Aggregator Free Federated Learning". In: *Proc. IEEE Blockchain.* Rhodes Island, Greece, Nov. 2020, pages 72–81 (cited on pages 135, 141).

[168] Record Evolution GmbH. *IoT Collaboration: The New Power in the Internet of Things.* Accessed: Aug. 25, 2021. URL: `https://www.record-evolution.de/en/iot-collaboration-the-collaborative-turn-in-the-internet-of-things/` (cited on page 135).

[169] Amr Rizk, Jordi Bisbal, Sonja Bergsträßer, and Ralf Steinmetz. "Brokerless Inter-Domain Virtual Network Embedding: A Blockchain-Based Approach". In: *it - Inform. Technol.* 60.5-6 (2018), pages 293–306 (cited on pages 132, 141).

[170] Zack Rutherford. *How Good Is eBid for Selling? An in-Depth Look.* Accessed: Aug. 21, 2021. URL: `https://www.salehoo.com/blog/is-ebid-a-viable-alternative-to-ebay` (cited on page 30).

[171] David Cerezo Sánchez. *Raziel: Private and Verifiable Smart Contracts on Blockchains.* Accessed: Oct. 02, 2021. URL: `https://eprint.iacr.org/2017/878.pdf` (cited on page 29).

[172] Aaliya Sarfaraz, Ripon K. Chakrabortty, and Daryl L. Essam. "A Tree Structure-Based Improved Blockchain Framework for a Secure Online Bidding System". In: *Comput. Secur.* 102 (2021), page 102147 (cited on page 29).

[173] S. Saxena, H. Farag, A. Brookson, H. Turesson, and H. Kim. "Design and Field Implementation of Blockchain Based Renewable Energy Trading in Residential Communities". In: *Proc. Int. Conf. SGRE.* Doha, Qatar, Nov. 2019, pages 1–6 (cited on pages 126, 140).

[174] Eder J Scheid, Bruno B Rodrigues, Lisandro Z Granville, and Burkhard Stiller. "Enabling dynamic sla compensation using blockchain-based smart contracts". In: *Proc. IEEE IM.* Washington, USA, Apr. 2019, pages 53–61 (cited on page 3).

[175] H. Seike, T. Hamada, T. Sumitomo, and N. Koshizuka. "Blockchain-Based Ubiquitous Code Ownership Management System without Hierarchical Structure". In: *IEEE SmartWorld/SCALCOM/UIC/ATC/CB-DCom/IOP/SCI.* Guangzhou, China, Oct. 2018, pages 271–276 (cited on pages 135, 141).

[176] Semos Cloud. *Employee Recognition Program Benefits and Ideas.* Accessed: Aug. 21, 2021. URL: `https://semoscloud.com/blog/employee-recognition-program-benefits-ideas/` (cited on page 135).

[177] 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". In: *IEEE Access* 8 (2020), pages 175713–175726 (cited on pages 124, 140).

[178] H. Al-Shaibani, N. Lasla, and M. Abdallah. "Consortium Blockchain-Based Decentralized Stock Exchange Platform". In: *IEEE Access* 8 (2020), pages 123711–123725 (cited on pages 133, 141).

[179] Pratima Sharma, Rajni Jindal, and Malaya Dutta Borah. "Blockchain technology for cloud storage: A systematic literature review". In: *ACM Comput. Surv.* 53.4 (2020), pages 1–32 (cited on pages 121, 122).

[180] Zeshun Shi, Cees de Laat, Paola Grosso, and Zhiming Zhao. *When Blockchain Meets Auction Models: A Survey, Some Applications, and Challenges.* Available at arXiv:2110.12534. 2021 (cited on page 119).

[181] Zeshun Shi, Huan Zhou, Cees de Laat, and Zhiming Zhao. "A Bayesian game-enhanced auction model for federated cloud services using blockchain". In: *Futur. Gener. Comp. Syst.* 136 (2022), pages 49–66 (cited on page 24).

[182] Zeshun Shi, Huan Zhou, Jayachander Surbiryala, Yang Hu, Cees de Laat, and Zhiming Zhao. "An Automated Customization and Performance Profiling Framework for Permissioned Blockchains in a Virtualized Environment". In: *Proc. IEEE CloudCom.* Sydney, Australia, July 2019, pages 404–410 (cited on page 94).

[183] AN Shwetha and CP Prabodh. "Auction System in Food Supply Chain Management Using Blockchain". In: *Proc. ICACECS.* Hyderabad, India, Aug. 2021, pages 31–40 (cited on pages 135, 141).

[184] Solidity Documentation. *Security Considerations.* Accessed 01 April 2022. 2022. URL: `https://docs.soliditylang.org/en/v0.4.24/security-considerations.html#security-considerations` (cited on page 86).

[185] Solidity Documentation. *Withdrawal from Contracts.* Accessed: Apr. 01, 2022. 2022. URL: `https://docs.soliditylang.org/en/v0.4.24/common-patterns.html#withdrawal-from-contracts` (cited on page 86).

[186] Alberto Sonnino, Michał Król, Argyrios G. Tasiopoulos, and Ioannis Psaras. *AStERISK: Auction-based Shared Economy ResolutIon System for blocKchain.* arXiv:1901.07824. 2019 (cited on pages 36, 37, 131, 141).

[187] M. Stübs, W. Posdorfer, and S. Momeni. "Blockchain-Based Multi-Tier Double Auctions for Smart Energy Distribution Grids". In: *Proc. IEEE ICC Workshop.* Dublin, Ireland, June 2020, pages 1–6 (cited on pages 124, 140).

[188] G. Sun, M. Dai, F. Zhang, H. Yu, X. Du, and M. Guizani. "Blockchain-Enhanced High-Confidence Energy Sharing in Internet of Electric Vehicles". In: *IEEE Internet Things J.* 7.9 (2020), pages 7868–7882 (cited on pages 126, 140).

[189] Steven Tadelis. *Game theory: an introduction.* Princeton University Press, 2013 (cited on page 75).

[190] Nachiket Tapas, Francesco Longo, Giovanni Merlino, and Antonio Puliafito. "Experimenting with smart contracts for access control and delegation in IoT". In: *Futur. Gener. Comp. Syst.* 111 (2020), pages 324–338 (cited on pages 54, 84).

[191] Subhasis Thakur, Barry P. Hayes, and John G. Breslin. "Distributed Double Auction for Peer to Peer Energy Trade Using Blockchains". In: *Proc. Int. Symp. EFEA.* Rome, Italy, Sept. 2018, pages 1–8 (cited on pages 124, 140).

[192] The Linux Foundation. *What Is Hyperledger?* Accessed: Aug. 31, 2021. URL: `https://www.hyperledger.org/` (cited on page 22).

[193] Zhitian Tu, Kun Zhu, Changyan Yi, and Ran Wang. "Blockchain-Based Privacy-Preserving Dynamic Spectrum Sharing". In: *Proc. Int. Conf. WASA.* Qingdao, China, Sept. 2020, pages 444–456 (cited on pages 128, 140).

[194] Rafael Brundo Uriarte, Huan Zhou, Kyriakos Kritikos, Zeshun Shi, Zhiming Zhao, and Rocco De Nicola. "Distributed service-level agreement management with smart contracts and blockchain". In: *Concurr. Comput.-Pract. Exp.* (2020), e5800 (cited on page 2).

[195] Guy R Vishnia and Gareth W Peters. "AuditChain: A trading audit platform over blockchain". In: *Front. Block.* 3 (2020), page 9 (cited on pages 134, 141).

[196] Abubaker Wahaballa, Zhen Qin, Hu Xiong, Zhiguang Qin, and Mohammed Ramadan. “A taxonomy of secure electronic English auction protocols”. In: *Int. J. Comput. Appl.* 37.1 (2015), pages 28–36 (cited on pages 121, 122).

[197] Baoyi Wang, Xiaoyuan Liu, Shaomin Zhang, et al. “Electric power transaction of electric vehicle based on smart contract and double auction”. In: *Adv. Comp. Signals Syst.* 4.1 (2020), pages 7–12 (cited on pages 126, 140).

[198] Jian Wang, Qianggang Wang, Niancheng Zhou, and Yuan Chi. “A novel electricity transaction mode of microgrids based on blockchain and continuous double auction”. In: *Energies* 10.12 (2017), page 1971 (cited on pages 123, 124, 140).

[199] Jiaqi Wang, Ning Lu, Qingfeng Cheng, Lu Zhou, and Wenbo Shi. “A Secure Spectrum Auction Scheme Without the Trusted Party Based on the Smart Contract”. In: *Digit. Commun. Netw.* (2020). In Press (cited on pages 128, 140).

[200] Naiyu Wang, Xiao Zhou, Xin Lu, Zhitao Guan, Longfei Wu, Xiaojiang Du, and Mohsen Guizani. “When energy trading meets blockchain in electrical power system: The state of the art”. In: *Appl. Energy* 9.8 (2019), page 1561 (cited on pages 6, 121, 122, 136).

[201] Taotao Wang, Chonghe Zhao, Qing Yang, Shengli Zhang, and Soung Chang Liew. “Ethna: Analyzing the underlying peer-to-peer network of ethereum blockchain”. In: *IEEE Trans. Netw. Sci. Eng.* 8.3 (2021), pages 2131–2146 (cited on page 18).

[202] Wenbo Wang, Dinh Thai Hoang, Peizhao Hu, Zehui Xiong, Dusit Niyato, Ping Wang, Yonggang Wen, and Dong In Kim. “A survey on consensus mechanisms and mining strategy management in blockchain networks”. In: *IEEE Access* 7 (2019), pages 22328–22370 (cited on pages 18, 19, 22, 121, 122).

[203] Brendan Ward, Albert Eloyan, and Alex Norta. *Establishing a Blockchain-Enabled, Highly Liquid, Auction-Based Employee Rewards Marketplace.* Accessed: Oct. 04, 2021. URL: `https://icofriends.com/urtoken/URT-WPv07b.pdf` (cited on pages 135, 141).

[204] Milena Wittwer. *Pay-As-Bid vs. First-Price Auctions: Similarities and Differences in Strategic Behavior.* Accessed: Oct. 04, 2021. URL: `http://web.stanford.edu/~wittwer/files/Wittwer2018` (cited on page 15).

[205] Gavin Wood. *Ethereum: A Secure Decentralised Generalised Transaction Ledger.* Accessed: Oct. 04, 2021. URL: `https://ethereum.github.io/yellowpaper/paper.pdf` (cited on page 12).

[206] S. Wu, Y. Chen, Q. Wang, M. Li, C. Wang, and X. Luo. “CReam: A Smart Contract Enabled Collusion-Resistant e-Auction”. In: *IEEE Trans. Inf. Forensics Security* 14.7 (2019), pages 1687–1701 (cited on page 12).

[207] Karl Wüst and Arthur Gervais. “Do you need a blockchain?” In: *Proc. CVCBT.* Zug, Switzerland, June 2018, pages 45–54 (cited on pages 3, 25).

[208] Shengnan Xia, Feilong Lin, Zhongyu Chen, Changbing Tang, Yongjin Ma, and Xinghuo Yu. “A Bayesian game based vehicle-to-vehicle electricity trading scheme for blockchain-enabled internet of vehicles”. In: *IEEE Trans. Veh. Technol.* 69.7 (2020), pages 6856–6868 (cited on pages 126, 140).

[209] Junfeng Xie, Helen Tang, Tao Huang, F Richard Yu, Renchao Xie, Jiang Liu, and Yunjie Liu. “A survey of blockchain technology applied to smart cities: Research issues and challenges”. In: *IEEE Commun. Surveys Tuts.* 21.3 (2019), pages 2794–2830 (cited on pages 30, 121, 122).

[210] Xiaogang Xing, Yuling Chen, Tao Li, Yang Xin, and Hongwei Sun. “A blockchain index structure based on subchain query”. In: *J. Cloud Comput.* 10.1 (2021), pages 1–11 (cited on page 3).

[211] Wei Xiong and Li Xiong. “Anti-collusion data auction mechanism based on smart contract”. In: *Inf. Sci.* 555 (2021), pages 386–409 (cited on pages 133, 141).

[212] Xiwei Xu, Ingo Weber, and Mark Staples. *Architecture for blockchain applications.* Springer, 2019 (cited on page 40).

[213] Y. Yan, B. Duan, X. Wu, and Y. Zhong. “A Novel Generation Right Trade in Blockchain-Enabled Continuous Double Auction System”. In: *Proc. Int. Conf. APSCOM.* Hong Kong, China, Nov. 2018, pages 1–6 (cited on pages 124, 140).

[214] Ruizhe Yang, F Richard Yu, Pengbo Si, Zhaoxin Yang, and Yanhua Zhang. “Integrated blockchain and edge computing systems: A survey, some research issues and challenges”. In: *IEEE Commun. Surveys Tuts.* 21.2 (2019), pages 1508–1532 (cited on pages 121, 122).

[215] Biao Yu, Yingwen Chen, Shaojing Fu, Wanrong Yu, and Xiaoli Guo. “Building Trustful Crowdsensing Service on the Edge”. In: *Proc. Int. Conf. WASA.* Hawaii, USA, June 2019, pages 445–457 (cited on pages 36, 37, 132, 141).

[216] L. Yu, J. Ji, Y. Guo, Q. Wang, T. Ji, and P. Li. “Smart Communications in Heterogeneous Spacecraft Networks: A Blockchain Based Secure Auction Approach”. In: *Proc. IEEE CCAAW Workshop.* Cleveland, USA, June 2019, pages 1–4 (cited on pages 128, 140).

[217] Shuai Yuan, Jun Wang, Bowei Chen, Peter Mason, and Sam Seljan. “An empirical study of reserve price optimisation in real-time bidding”. In: *Proc. ACM SIGKDD.* New York, USA, Aug. 2014, pages 1897–1906 (cited on page 72).

[218] Javad Zarrin, Hao Wen Phang, Lakshmi Babu Saheer, and Bahram Zarrin. “Blockchain for decentralization of internet: prospects, trends, and challenges”. In: *Cluster Comput.* 24.4 (2021), pages 2841–2866 (cited on page 23).

[219] A. Zavodovski, S. Bayhan, N. Mohan, P. Zhou, W. Wong, and J. Kangasharju. "DeCloud: Truthful Decentralized Double Auction for Edge Clouds". In: *Proc. IEEE ICDCS*. Dallas, USA, July 2019, pages 2157–2167 (cited on pages 36, 37, 131, 141).

[220] Zeeve Inc. *Smart Contract Standardization.* Accessed: Aug. 28, 2021. URL: `https://www.zeeve.io/blog/smart-contract-standardization/` (cited on page 33).

[221] Hong Zhang, Hongbo Jiang, Bo Li, Fangming Liu, Athanasios V Vasilakos, and Jiangchuan Liu. "A framework for truthful online auctions in cloud computing with heterogeneous user demands". In: *IEEE Trans. Comput.* 65.3 (2015), pages 805–818 (cited on page 1).

[222] Rui Zhang, Rui Xue, and Ling Liu. "Security and privacy on blockchain". In: *ACM Comput. Surv.* 52.3 (2019), pages 1–34 (cited on pages 18, 121, 122).

[223] Shaomin Zhang, Miao Pu, Baoyi Wang, and Bin Dong. "A privacy protection scheme of microgrid direct electricity transaction based on consortium blockchain and continuous double auction". In: *IEEE Access* 7 (2019), pages 151746–151753 (cited on pages 125, 140).

[224] Xinglin Zhang, Zheng Yang, Wei Sun, Yunhao Liu, Shaohua Tang, Kai Xing, and Xufei Mao. "Incentives for mobile crowd sensing: A survey". In: *IEEE Commun. Surveys Tuts.* 18.1 (2015), pages 54–67 (cited on pages 121, 122).

[225] Y. Zhang, C. Lee, D. Niyato, and P. Wang. "Auction Approaches for Resource Allocation in Wireless Systems: A Survey". In: *IEEE Commun. Surveys Tuts.* 15.3 (2013), pages 1020–1041 (cited on page 128).

[226] Yang Zhang, Chonho Lee, Dusit Niyato, and Ping Wang. "Auction approaches for resource allocation in wireless systems: A survey". In: *IEEE Commun. Surveys Tuts.* 15.3 (2012), pages 1020–1041 (cited on pages 121, 122).

[227] Zibo Zhao, Kiyoshi Nakayama, and Ratnesh Sharma. "Decentralized Transactive Energy Auctions with Bandit Learning". In: *Proc. IEEE TESC*. Minneapolis, MN, USA, July 2019, pages 1–5 (cited on pages 124, 140).

[228] S. Zheng, T. Han, Y. Jiang, and X. Ge. "Smart Contract-Based Spectrum Sharing Transactions for Multi-Operators Wireless Communication Networks". In: *IEEE Access* 8 (2020), pages 88547–88557 (cited on pages 128, 129, 140).

[229] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. "Blockchain challenges and opportunities: A survey". In: *Int. J. Web Grid Serv.* 14.4 (2018), pages 352–375 (cited on pages 121, 122).

[230] Huan Zhou, Yang Hu, Jinshu Su, Cees de Laat, and Zhiming Zhao. "Cloudsstorm: An application-driven framework to enhance the programmability and controllability of cloud virtual infrastructures". In: *Proc. IEEE CLOUD*. San Francisco, USA, July 2018, pages 265–280 (cited on page 105).

[231] Huan Zhou, Xue Ouyang, Zhijie Ren, Jinshu Su, Cees de Laat, and Zhiming Zhao. "A blockchain based witness model for trustworthy cloud service level agreement enforcement". In: *Proc. IEEE INFOCOM*. Paris, France, May 2019, pages 1567–1575 (cited on pages 27, 47, 62, 118).

[232] Huan Zhou, Zeshun Shi, Xue Ouyang, and Zhiming Zhao. "Building a blockchain-based decentralized ecosystem for cloud and edge computing: an ALLSTAR approach and empirical study". In: *Peer Peer Netw. Appl.* 14.6 (2021), pages 3578–3594 (cited on page 46).

[233] Zigurat. *Cloud, Edge, and Fog Computing–Practical Application for Each.* Accessed: Aug. 26, 2021. URL: `https://www.e-zigurat.com/innovation-school/blog/cloud-edge-fog-computing-practical-applications/` (cited on page 131).

List of Abbreviations

ABSS	Auction-Based Service Selection
AE	Asymmetric Encryption
API	Application Programming Interface
AWS	Amazon Web Services
BFT	Byzantine Fault Tolerance
BNE	Bayesian Nash Equilibrium
CA	Certificate Authority
CCHP	Combined Cooling, Heating, and Power
CMN	Collaborative Mining Network
CPSS	Cyber-Physical-Social Systems
CR	Cognitive Radio
CS	Commitment Scheme
DAG	Directed Acyclic Graph
DApp	Decentralized Application
DDoS	Distributed Denial of Service
DevOps	Development and Operations
DEX	Decentralized Exchange
DG	Distributed Generation
DGUI	DApp Graphical User Interface
DHT	Distributed Hash Table
DP	Differential Privacy
DPoS	Delegated Proof of Stake
DS	Digital Signature
DWCM	Decentralized Witness Committee Management
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EIP	Ethereum Improvement Proposal
ENS	Ethereum Name Service
ERC	Ethereum Request for Comments
EV	Electric Vehicle
FCC	Federal Communications Commission
FL	Federated Learning
FPSB	First-Price Sealed-Bid
GDPR	General Data Protection Regulation
GFP	Generalized First-Price
GSP	Generalized Second-Price
GUI	Graphical User Interface
HE	Homomorphic Encryption
IaaS	Infrastructure-as-a-Service
IESDS	Iterated Elimination of Strictly Dominated Strategies

ICO	Initial Coin Offering
IoE	Internet of Energy
IoT	Internet of Things
IoV	Internet of Vehicles
IPFS	InterPlanetary File System
IPO	Initial Public Offering
MBPB	Maximum Batches Per Block
MILP	Mixed-Integer Linear Programming
MPC	Multi-Party Computation
NFV	Network Function Virtualization
NIaaS	Networked Infrastructure-as-a-Service
P2P	Peer-to-Peer
PB	Permissioned Blockchain
PBFT	Practical Byzantine Fault Tolerance
PBE	Perfect Bayesian Equilibrium
PBS	Primary Base Station
PoA	Proof of Authority
PoET	Proof of Elapsed Time
PoS	Proof of Stake
PoW	Proof of Work
QoS	Quality of Service
SAW	Simple Additive Weighting
SCFO	Smart Contract Factory Orchestration
SGX	Software Guard Extensions
SLA	Service-Level Agreement
TC	Traffic Control
TEE	Trusted Execution Environment
TMS	Timed Message Submission
TPS	Transactions Per Second
TTP	Trusted Third Party
UAV	Unmanned Aerial Vehicle
UI	User Interface
V2G	Vehicle-to-Grid
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VCG	Vickrey–Clarke–Groves
VM	Virtual Machine
VNE	Virtual Network Embedding
VPP	Virtual Power Plant
ZKP	Zero-Knowledge Proof

Summary

Industrial applications (e.g., remote live event broadcasting) require high-quality cloud services to deliver business value. These cloud applications often rely on resources and services from different providers due to the distributed geographic location of data sources and diverse access policies. The generation and enforcement of the Service-Level Agreements (SLAs) are crucial for guaranteeing the quality of both application services and underlying cloud infrastructure. Traditional service allocation methods through auctions usually involve a centralized auctioneer to coordinate the auction procedure, which is expensive due to high commission fees. They also suffer from a single point of failure, as auctioneers can potentially be malicious in some cases.

Recently, blockchain has emerged as a decentralized platform to support trustworthy online transactions in various scenarios. A blockchain provides a trustworthy environment among its decentralized users via the immutability, transparency, and security of the ledgers and programs (i.e., smart contracts). It leverages consensus mechanisms to agree on new data and cryptographic technologies to guarantee data integrity and immutability.

Blockchain and smart contracts can provide a decentralized mechanism for service auctions and SLA automation; however, it is still challenging to:

- select cost-effective service providers and customers to sign SLAs due to the lack of an effective auction model;
- detect service violations in the signed auction agreements (namely SLAs in this context) due to the blockchain cannot confirm the veracity of external data;
- design effective incentive mechanisms among different service stakeholders to motivate them to participate in the auction and SLA monitoring.

To tackle these challenges, we propose a novel framework called Auction and Witness Enhanced trustworthy SLA for Open, decentralized service MarkEtplaces (AWESOME). We aim to provide an efficient and trustworthy environment for

generating SLAs and trading services. Especially, our integrated model includes a novel witness mechanism and supports interactions between untrusted service providers, customers, and witnesses to complete decentralized auctions and SLA enforcement on the blockchain. We present the cloud service as an example, and the proposed model can be easily extended to other service transaction scenarios.

The main research of this thesis is how to enhance the efficiency and trustworthiness of the cloud SLAs using decentralized auctions and witnesses. For that, we explained the state-of-the-art technologies and open challenges for building a decentralized service auction framework. We reviewed the existing solutions for integrating blockchain and auction models, with several application-oriented taxonomies generated. Additionally, we highlighted open research challenges and future directions toward integrated blockchain-auction models.

Next, we researched how to automate the decentralized service auction and quality monitoring process in an SLA model. We designed the AWESOME framework based on blockchain and smart contracts. The proposed framework contains four submodules: a customizable graphical user interface, an auction-based service selection model, a witness committee management mechanism, and a smart contract factory orchestration. We also developed a prototype AWESOME decentralized application (DApp) based on the Ethereum blockchain. Extensive experiments were presented to evaluate the proposed model and DApp.

Furthermore, to improve the efficiency of service auctions for managing federated clouds, we modeled the partition of federated cloud services as a graph partition problem. Then, the service selection was modeled as a decentralized auction model based on Bayesian game theory. The derived Bayesian Nash Equilibrium (BNE) enables the selection of cost-effective providers to construct the federated cloud SLAs. Moreover, a timed message submission (TMS) algorithm was proposed to protect auction privacy on the blockchain.

To enhance the trustworthiness of federated SLAs in a decentralized service environment, we designed an incentive mechanism for decentralized witnesses to monitor service quality. Especially, the majority decides whether the SLA is violated, and all the witnesses are motivated to participate and be honest. The monitoring process was also modeled as a Bayesian game. The derived BNE ensures consistent and trustworthy monitoring of federated SLAs. We validated the equilibrium situations of the BNE and implemented the proposed mechanism on the Ethereum blockchain.

An important aspect of operating blockchain services is meeting the scalability requirements of the AWESOME framework. As blockchain is designed as the underlying trust device in our framework, we further conducted an empirical study to evaluate the performance of different blockchain platforms, including their scalability, stability, and resource consumption. The experimental and analytical results provided insightful recommendations for choosing the appropriate blockchain infrastructure for the AWESOME framework.

Samenvatting

Industriële toepassingen (bijv. uitzending van live-evenementen) vereisen hoogwaardige cloudservices om zakelijke waarde te leveren. Deze cloudapplicaties zijn vaak afhankelijk van resources en services van verschillende providers vanwege de gedistribueerde geografische locatie van gegevensbronnen en divers toegangsbeleid. Het genereren en handhaven van de Service Level Agreements (SLA's) is cruciaal voor het waarborgen van de kwaliteit van zowel applicatiediensten als onderliggende cloudinfrastructuur. Bij traditionele methoden voor het toewijzen van diensten via veilingen is meestal een gecentraliseerde veilingmeester betrokken om de veilingprocedure te coördineren, wat duur is vanwege de hoge commissiekosten. Ze hebben ook last van een single point of failure, omdat veilingmeesters in sommige gevallen potentieel kwaadaardig kunnen zijn.

Onlangs is blockchain naar voren gekomen als een gedecentraliseerd platform om betrouwbare online transacties in verschillende scenario's te ondersteunen. Een blockchain biedt een betrouwbare omgeving onder zijn gedecentraliseerde gebruikers via de onveranderlijkheid, transparantie en beveiliging van de grootboeken en programma's (d.w.z. slimme contracten). Het maakt gebruik van consensusmechanismen om overeenstemming te bereiken over nieuwe gegevens en cryptografische technologieën om de integriteit en onveranderlijkheid van gegevens te garanderen.

Blockchain en slimme contracten kunnen een gedecentraliseerd mechanisme bieden voor serviceveilingen en SLA-automatisering; het is echter nog steeds een uitdaging om:

- kosteneffectieve dienstverleners en klanten te selecteer om SLA's te ondertekenen vanwege het ontbreken van een effectief veilingmodel;
- serviceschendingen te detecteren in de ondertekende veilingovereenkomsten (namelijk SLA's in deze context) als gevolg van de blockchain kan de waarheidsgetrouwheid van externe gegevens niet bevestigen;

- effectieve stimuleringsmechanismen onder verschillende belanghebbenden van de dienst te ontwerpen om hen te motiveren om deel te nemen aan de veiling en SLA-monitoring.

Om deze uitdagingen aan te gaan, stellen we een nieuw raamwerk voor met de naam Auction and Witness Enhanced trustworthy SLA for Open, decentralized service MarkEtplaces (AWESOME). We streven ernaar een efficiënte en betrouwbare omgeving te bieden voor het genereren van SLA's en handelsdiensten. Ons geïntegreerde model bevat vooral een nieuw getuigenmechanisme en ondersteunt interacties tussen niet-vertrouwde serviceproviders, klanten en getuigen om gedecentraliseerde veilingen en SLA-handhaving op de blockchain te voltooien. We presenteren de cloudservice als voorbeeld en het voorgestelde model kan eenvoudig worden uitgebreid naar andere scenario's voor servicetransacties.

Hier toe hebben we onderzocht hoe de efficiëntie en betrouwbaarheid van de cloud-SLA's verbeteren met behulp van gedecentraliseerde veilingen en getuigen. Daarvoor hebben we de state-of-the-art technologieën en open uitdagingen uitgelegd voor het bouwen van een gedecentraliseerd serviceveilingraamwerk zijn. We hebben de bestaande oplossingen voor het integreren van blockchain- en veilingmodellen beoordeeld, waarbij verschillende toepassingsgerichte taxonomieën zijn gegenereerd. Daarnaast hebben we openstaande onderzoeksuitdagingen en toekomstige richtingen naar geïntegreerde blockchain-veilingmodellen benadrukt.

Voorts is onderzocht hoe de decentrale serviceveiling en het kwaliteitsbewakingsproces automatiseren in een SLA-model. We hebben het AWESOME raamwerk ontworpen op basis van blockchain en slimme contracten. Het voorgestelde raamwerk bevat vier submodules: een aanpasbare grafische gebruikersinterface, een op veilingen gebaseerd serviceselectiemodel, een mechanisme voor het beheer van getuigencommissies en een slimme contractfabriekorkestratie. We ontwikkelden ook een prototype AWESOME gedecentraliseerde applicatie op basis van de Ethereum-blockchain. Uitgebreide experimenten werden gepresenteerd om het voorgestelde model en DApp te evalueren.

Om de efficiëntie van serviceveilingen voor het beheer van federatieve clouds te verbeteren hebben we eerst de partitie van federatieve cloudservices gemodelleerd als een grafiekpartitieprobleem. Vervolgens werd de serviceselectie gemodelleerd als een gedecentraliseerd veilingmodel op basis van de Bayesiaanse speltheorie. Het afgeleide Bayesian Nash Equilibrium (BNE) maakt de selectie van kosteneffectieve providers mogelijk om de gefedereerde cloud-SLA's te bouwen. Bovendien werd een algoritme voor getimede berichtverzending voorgesteld om de veilingprivacy op de blockchain te beschermen.

De betrouwbaarheid van gefedereerde SLA's in een gedecentraliseerde serviceomgeving hebben we verbeterd door een stimuleringsmechanisme ontwerpen voor gedecentraliseerde getuigen om de kwaliteit van de dienstverlening te bewaken. Vooral de meerderheid beslist of de SLA wordt geschonden en alle getuigen zijn gemotiveerd om mee te doen en eerlijk te zijn. Het monitoringproces werd ook gemodelleerd als een Bayesiaans spel. De afgeleide BNE zorgt voor een consistente en betrouwbare monitoring van federatieve SLA's. We hebben de evenwichtssituaties van de BNE gevalideerd en het voorgestelde mechanisme geïmplementeerd op de Ethereum-blockchain.

Een belangrijk aspect zijn de schaalbaarheidsvereisten van het AWESOME raamwerk. Omdat blockchain is ontworpen als het onderliggende vertrouwensapparaat in ons raamwerk, hebben we verder een empirisch onderzoek uitgevoerd om de prestaties van verschillende blockchain-platforms te evalueren, inclusief hun schaalbaarheid, stabiliteit en resourceverbruik. De experimentele en analytische resultaten leverden inzichtelijke aanbevelingen op voor het kiezen van de juiste blockchain-infrastructuur voor het AWESOME raamwerk.

Publications

Journals:

1. **Zeshun Shi**, Huan Zhou, Cees de Laat, and Zhiming Zhao. "A Bayesian Game-Enhanced Auction Model for Federated Cloud Services Using Blockchain". *Future Generation Computer Systems* (2022): 136, pp.49-66.

2. **Zeshun Shi**, Veno Ivankovic, Siamak Farshidi, Jayachander Surbiryala, Huan Zhou, and Zhiming Zhao. "AWESOME: An Auction and Witness Enhanced SLA Model for Decentralized Cloud Marketplaces". *Journal of Cloud Computing* (2022): 11(1), pp.1-25.

3. **Zeshun Shi**, Cees de Laat, Paola Grosso, and Zhiming Zhao. "Integration of Blockchain and Auction Models: A Survey, Some Applications, and Challenges". *IEEE Communications Surveys & Tutorials.* (To appear)

4. **Zeshun Shi**, Jeroen Bergers, Ken Korsmit, and Zhiming Zhao. "AUDITEM: Toward an Automated and Efficient Data Integrity Verification Model Using Blockchain". *IEEE Internet of Things Journal.* (Under revision)

5. Huan Zhou, **Zeshun Shi**, Ouyang Xue, and Zhiming Zhao. "Building a blockchain-based decentralized ecosystem for cloud and edge computing: an ALLSTAR approach and empirical study". *Peer-to-Peer Networking and Applications* (2021): 14(6), pp.3578-3594. (as co-first author)

6. Rafael Brundo Uriarte, Huan Zhou, Kyriakos Kritikos, **Zeshun Shi**, Zhiming Zhao, and Rocco De Nicola. "Distributed service-level agreement management with smart contracts and blockchain". *Concurrency and Computation: Practice and Experience* (2021): 33(14), pp.1-17. (as co-first author)

7. Zhiming Zhao, Spiros Koulouzis, Riccardo Bianchi, Siamak Farshidi, **Zeshun Shi**, Ruyue Xin, Yuandou Wang, et al. "Notebook-as-a-VRE (NaaVRE): From private notebooks to a collaborative cloud virtual research environment". *Software: Practice and Experience* (2022): 52(9), pp.1947-1966.

Conferences:

1. **Zeshun Shi**, Huan Zhou, Yang Hu, Jayachander Surbiryala, Cees de Laat, and Zhiming Zhao. “Operating permissioned blockchain in clouds: A performance study of hyperledger sawtooth”. *In 2019 18th IEEE International Symposium on Parallel and Distributed Computing (ISPDC)*, pp. 50-57. IEEE, 2019.

2. **Zeshun Shi**, Siamak Farshidi, Huan Zhou, and Zhiming Zhao. “An Auction and Witness Enhanced Trustworthy SLA Model for Decentralized Cloud Marketplaces”. *In ACM International Conference on Information Technology for Social Good (GoodIT)*, pp. 109-114. ACM, 2021.

3. **Zeshun Shi**, Huan Zhou, Jayachander Surbiryala, Yang Hu, Cees de Laat, and Zhiming Zhao. “An automated customization and performance profiling framework for permissioned blockchains in a virtualized environment”. *In 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), workshop on resource brokering with blockchain (RBChain)*, pp. 404-410. IEEE, 2019.

4. **Zeshun Shi**, Huan Zhou, Yang Hu, Spiros Koulouzis, Carlos Rubia, and Zhiming Zhao. “Co-located and Orchestrated Network Fabric (CONF): An Automated Cloud Virtual Infrastructure for Social Network Applications”. *In European Conference on Parallel Processing (Euro-Par), workshop on Large Scale Distributed Virtual Environments (LSDVE)*, pp. 464-475. Springer, Cham, 2019.

5. Jeroen Bergers, **Zeshun Shi**, Ken Korsmit, and Zhiming Zhao. “DWH-DIM: A Blockchain Based Decentralized Integrity Verification Model for Data Warehouses”. *In 2021 IEEE International Conference on Blockchain (Blockchain)*, pp. 221-228. IEEE, 2021.

6. Veno Ivankovic, **Zeshun Shi**, and Zhiming Zhao. “A Customizable dApp Framework for User Interactions in Decentralized Service Marketplaces”. *In 2022 IEEE International Conference on Smart Internet of Things (SmartIoT))*, pp. 224-231. IEEE, 2022. (Best Paper Award).

7. Huan Zhou, **Zeshun Shi**, Yang Hu, Pieter Donkers, Andrey Afanasyev, Spiros Koulouzis, Arie Taal, Alexandre Ulisses, and Zhiming Zhao. “Large distributed virtual infrastructure partitioning and provisioning across providers”. *In 2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 56-63. IEEE, 2019. (Best Student Paper Award).

8. Ruyue Xin, Jardenna Mohazzab, **Zeshun Shi**, and Zhiming Zhao. “CBProf: Customisable Blockchain-as-a-Service Performance Profiler in Cloud Environments”. *In International Conference on Blockchain (ICBC)*, pp. 131-139. Springer, Cham, 2021.

9. Bram Hoogenkamp, Siamak Farshidi, Ruyue Xin, **Zeshun Shi**, Peng Chen, and Zhiming Zhao. "A Decentralized Service Control Framework for Decentralized Applications in Cloud Environments". *In European Conference on Service-Oriented and Cloud Computing*, pp. 65-73. Springer, Cham, 2022.

Code Repositories:

1. AWESOME - An Auction and Witness Enhanced SLA Model for Decentralized Cloud Marketplaces.
   Link: `https://github.com/ZeshunShi/AWESOME`
2. SC4CloudAuction - Smart Contracts for Federated Cloud Auction.
   Link: `https://github.com/ZeshunShi/SC4CloudAuction`
3. ALLSTAR - Performance Analysis for Permissioned Blockchain Platforms.
   Link: `https://github.com/ZeshunShi/ALLSTAR`

Acknowledgments

It has been such a long and challenging journey for me to pursue a Ph.D. at UvA. I still remember coming here alone four years ago, when everything about this unfamiliar country made me curious and difficult to adapt to. The language barrier, the brand new research topic, and the continous rainy weather in Amsterdam had all been very frustrating for me at some point. I have gone through many difficulties in such a long way to finish this thesis. Looking back at what I have achieved in these four years, I am grateful for all I have experienced here. Many vivid images come to my mind, and there are many people I need to thank at this special time.

First of all, I would like to thank my Ph.D. promoter Prof. Cees de Laat. I still remember when we discussed blockchain and Dutch auction topics, he was always funny, encouraging, and trying to motivate me in any way. The regular meetings and discussions with him have inspired me a lot. I would like to thank my secondary Ph.D. promoter Dr. Paola Grosso. I am very grateful for her valuable comments on my publications and thesis, as well as her kindness support of my academic efforts. I also need to thank my Ph.D. co-promoter Dr. Zhiming Zhao, who guided me throughout this research project. He has provided me with careful guidance and selfless help in every step of the process, including selecting the topic, writing, revising, and finalizing the thesis. It has been such an honor to work with you three for the past four years.

Next, I would like to express my gratitude to Prof. Radu Prodan, Prof. Andy Pimentel, Prof. Leon Gommans, Prof. Sander Klous, Dr. Ana Oprescu, and Dr. Zoltan Mann for their kindness to be my Ph.D. defense committee members. I really appreciate the time you have taken to evaluate my thesis and the valuable comments you have made.

I would like to thank my former colleagues Huan Zhou, Yang Hu, and Jayachander Surbiryala, who always gave me enough academic help and guidance. A special thanks to Yuri Demchenko for his guidance and support while I was a teaching assistant in his class. I would like to give thanks to my (both former

and current) colleagues in the SNE and MNS groups, including but not limited to Adam Belloum, Chrysa Papagianni, Cyril Hsu, Florian Speelman, Garazi Muguruza Lasa, Grace Millerson, Henk Dreuning, Jamila Kassem, Joseph Hill, Leonardo Boldrini, Llorenc Escola Farras, Lourens Veen, Marco Brohet, Misha Mesarcik, Ralph Koning, Reggie Cushing, Riccardo Bianchi, Saba Amiri, Sara Shakeri, Siamak Farshidi, Spiros Koulouzis, Uraz Odyurt, Xiaofeng Liao. They are all very nice, and I enjoy the lab's working environment here. I am so lucky to have met and worked with them during my Ph.D.

In addition, I would like to thank some of my Chinese friends and colleagues, including Hongyun Liu, Lu Zhang, Lu-Chi Liu, Na Li, Ning Chen, Qi Wang, Ruyue Xin, Wei Wang, Wenyang Wu, Xiaotian Guo, Yahui Zhang, Yixian Shen, Yuandou Wang, Zenglin Shi. I always enjoy getting together with them and having a good time. I also want to thank my neighbor couple, Caixia Wei and Geng Chen, who have always made me look forward to drinking and playing cards on weekends.

I would like to thank my wife, Qianru Zuo, from the bottom of my heart. I can't even imagine that I could have completed this journey without you. I still remember the first time I saw you in the cafeteria of Science Park. Fate was so amazing that we quickly fell in love and got married. You were always there for me during every tough moment of my Ph.D., helping me, cooking delicious food, and encouraging me to overcome those hardships. Meeting you in Amsterdam was my most precious treasure, and I will always stand by your side for the rest of my life.

Finally, I would like to thank my parents for being proud of me and encouraging me at all times. Thanks to my parents-in-law, who are always kind, generous, and understanding. I would also like to thank all my relatives and friends in China for calling me to help me get through the toughest time in a foreign country. You are the strongest support of my life.

Zeshun Shi
Amsterdam, October 2022