A Blockchain-based Housing Rental System

Chen Qi-Long[1], Ye Rong-Hua[1] and Lin Fei-Long[1]
[1]*College of Mathematics and Computer Science, Zhejiang Normal University, Yingbin Avenue 688, Jinhua, China*

Keywords: Blockchain, Smart contract, Ethereum, IPFS, Rental system.

Abstract: This paper proposes a blockchain-based solution for housing rental system to achieve peer-to-peer sharing of listings information in an intermediary-free manner. The new solution programmatically converts traditional lease agreements into smart lease contracts, improve the efficiency of the leasing process, and store transaction records for the leasing process to provide legal protection for tenants and landlords. The emerging technologies, including smart contract and IPFS (Inter Planetary File System) are exploited in the new system. IPFS is used to store listings information. The Ethereum's Smart Contract is used to manage the hash of IPFS listings information, and a smart lease contract is written to make the housing rental process traceable and visible.

1 INTRODUCTION

The 21st century is an information age, and the value of data is getting higher and higher. In a traditional Internet application, it tends to have more users and more data, so that it can obtain greater business value, and this data can only be controlled by one enterprise. Internet applications such housing rental system has such a problem, it has a large enough listings and users can freely change the listing price (Yu, 2018), control the market, even monopoly housing rental market may occur (Zhuang, 2017). Provide listings information publishing and display is unable to meet demand between the landlord and the tenant. Tenant and the landlord would normally use a traditional lease agreement to protect their interests during the lease, the two sides funds use third-party payment instruments in the transaction process of turnover, can't ensure the realization of irreversible transactions, nor will it be able to completely store the transaction records generated during the lease period, so it is easy for both parties to generate inevitable coordination disputes.

Blockchain technology can solve the problems of the current rent areas centralized access control and the problem of not being able to systematically manage the leasing process. The blockchain uses peer-to-peer technology (Yu, 2017), which makes it possible to implement decentralized applications (Wilkinsom, 2016). The proof of work (POW) (Gupta, 2017) is used to ensure that the stored block information cannot be tampered with (Cheng, 2018). The emergence of blockchain also laid the foundation for smart contracts. A smart contract is a new decentralized infrastructure and distributed computing paradigm that uses automated scripting code to program and manipulate data.

This paper proposes a decentralized housing rental system based on blockchain technology. The main contributions can be summarized as follows:

- Use blockchain technology, to ensure that in an environment without a central authority, share listings on the premise of preventing listings information from being tampered.
- Compile traditional lease agreements into a smart lease contract for runs automatically and operational data.
- This paper introduces the core operation mechanism of the blockchain housing rental system and the functional division of the smart contract module of the system.

The reminder of this paper is organized as follows: Section 2 describes the related work; Section 3 introduces our blockchain-based shared listing information and solutions that smart lease contract; Section 4 presents the analysis and evaluation of the system; Section 5 summarizes the paper.

2 RELATED WORK

Blockchain (Yuan, 2016) is a new decentralized infrastructure and distributed computing paradigm

Qi-Long, C., Rong-Hua, Y. and Fei-Long, L.
A Blockchain-based Housing Rental System.
DOI: 10.5220/0008097201840190
In *Proceedings of the International Conference on Advances in Computer Technology, Information Science and Communications (CTISC 2019)*, pages 184-190
ISBN: 978-989-758-357-5

that has emerged with the increasing popularity of digital cryptocurrencies such as Bitcoin (Nakamoto, 2009). It has the ability to eliminate friction and reduce risk (Kim, 2018), while also limiting the ability to recoup or cancel transactions (Kim, 2018). Peer-to-peer distributed network records the open history of transactions. Ethereum (2015) has become the second largest blockchain platform after Bitcoin. It has all the features of the blockchain and has developed Turing's complete EVM (Ethereum virtual machine) (Wood, 2014). Users can build arbitrarily complex and precisely defined smart contracts and decentralized applications based on Ethereum, it is called DApp (Decentralized Application) (David, 2015).

The smart contract is a computerized trading agreement that enforces contract terms (Kim, 2018). It translates contract terms into code and embeds them into self-enforcing resources, minimizing the need for trusted intermediaries between transactions, as well as malicious or accidental occurrences (Cong, 2018). The smart contract is a self-executing script that resides on a blockchain. Each node in the blockchain that enables smart contracts runs a virtual machine (VM). Blockchain network acts as a distributed VM. Smart contracts are deterministic, and the same input will always produce the same output. The smart contracts operate as autonomous actors and their behaviour is completely predictable.

Oraclize external services can provide data for smart contract (Adler, 2018), and the data obtained from the outside world by its proof of authenticity to prove the original data source is authentic and has not been tampered with (de Pedro, 2017).

IPFS is a peer-to-peer distributed file system that uses cryptographic hash content addressing to prevent file content from being tampered with and is designed to connect all computing devices to the same file system (Benet, 2014). At the same time, IPFS provides a high-throughput, content-addressable, fast-storage model and a hyperlink with content addressing.

The application of blockchain technology in the real estate field has been proposed. Stephen King and McLernon (2016) proposed that centralized multi-listed real estate services occupy a market leading position in the market and users, which in turn sets rules, prices and conditions. It describes the limitations of the existing MLS (Multiple Listing Service) structure (Stephen, 2016), which limits the rules, regulations, and fees for users to choose the default platform, or does not access the platform at all, thus missing a huge global market. It provides a simple white paper, raised questions and ideas, but did not see specific solutions and implementation.

Dickason proposed the Midasium (Dickason, 2016) blockchain system, which provides smart rental products that allow automatic payments to landlords, contractors, councils and property managers, using rent or bonds as a source, and through smart contract for transparent disbursement of funds and automated reconciliation, the purpose is to change the ecology of real estate transparency, efficiency, safety, and improve profitability for investors. It provides a simple process for a smart lease contract, and without seeing a specific implementation, a detailed flow will be given in this article.

3 SOLUTION

The solution in this paper is based on Ethereum's Smart Contract, IPFS and Oraclize services. Own to the blockchain is an expensive data storage medium, it is not suitable for storing digital content of listings information with multiple images. This article will make Ethereum's smart contract and IPFS to cooperate with each other, use IPFS to store listing information, and IPFS hashes are stored on blockchain smart contracts to provide traceability and authenticity of IPFS hashes. Smart leasing contract can be written to run on the blockchain, save landlord and tenant leasing transactions in the process, it can't be tampered, as the legal basis for the future, while safeguarding the personal interests of landlords and tenants. In order to ensure the authenticity of the lease bill, this article will automatically generate the lease bill for each period during the lease period through the smart lease contract. In this paper, Oraclize will be used as the trigger tool for each lease bill creation. Because Oraclize is a service for smart contracts and blockchain applications, just like platform services. There are already a lot of blockchain platforms are using it, in addition to Ethernet Square there Rootsocks, R3 Corda, Hyperledger, Fabric and EOS are using the service. The most important thing is Oraclize ensures that the data source is real, credible, so select the Oraclize service to start the scheduled task in this scenario.

3.1 System Core Operating Mechanism

Figure 1 shows the core operating mechanism of the Ethereum Smart Contract, IPFS, and Oraclize services in this system, and the interaction process between smart contracts and participants.

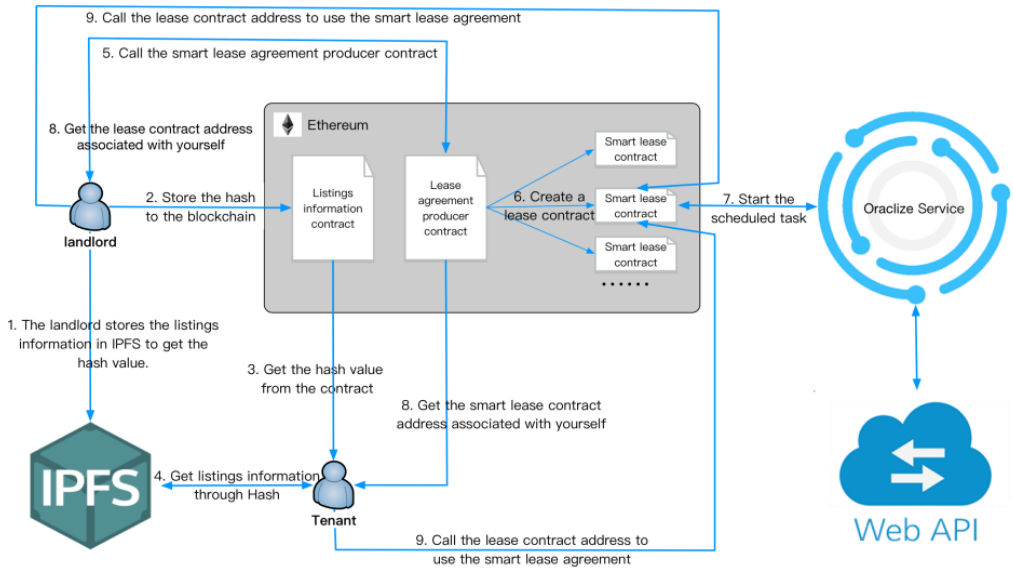

Figure 1: The core operating mechanism of the rental system based on Ethereum, IPFS, and Oraclize.

In Figure 1, the landlord needs to send the listings information stored in the JSON (JavaScript Object Notation) (Feiler, 2018) format to the IPFS. After the IPFS obtains the request, it will return the hash of the listings information and store it in the listings information contract. The tenant obtains the hash of the listing information by visiting the listing information contract, and the tenant obtains the listings from the IPFS through the hash. If the landlord agrees with the tenant, the landlord will create a smart lease contract through the lease agreement producer contract and save the smart contract address. The landlord and the tenant obtain the own smart lease contract address through the lease contract creator, and operate the smart lease contract through the address.

3.2 Smart Contract Module Function Division

Through the module function division, we divide the contracts designed in this paper into three categories: listings contract, lease agreement producer contract, smart lease contract. This section will describe the specific features of the three contracts in detail:

1. **Listings Contract (LC)**: It's used to store the hash of the listing information generated by IPFS. The landlord can control whether the stored hash can be retrieved by the outside world. Therefore, the hash has the following three statuses:

① Unavailable status: When the landlord does not want to rent the house or the house is being renovated, we can set the status of the current listing information hash to the unavailable-status through the listing information contract, and the hash cannot be retrieved by the outside world;

② Rented-status: When the house is rented, the landlord can mark the current listing as rented, and the hash cannot be retrieved by the outside world;

③ Available-status: The house is looking for tenants and the hash can be retrieved from outside.

2. **Lease Agreement Producer Contract (LAPC)**: The landlord will send information about the lease process, such as the lessee's public key, lease term, rent, deposit, date of lease, etc., to create a smart lease contract. The created contract address will be saved in the lease agreement producer contract, and the participant will only be able to access the smart lease contract address associated with himself, which makes the smart lease contract traceable and easy to manage.

3. **Smart Lease Contract (SLC)**: This paper abstracts the traditional lease agreement. There will be five statuses in the lease process: status of signing the contract, paying the deposit status, status of renting process, returning status of the deposit, and status of completing the contract. In these five statuses, the status of lease process and the retuned deposit status are sub-statuses of the contract. In the leasing process, a rental bill is generated, which is divided into four statuses: pending record meter status, settlement status, paid status and take out the rent status. After the lease is over, the landlord needs to

return the deposit to the tenant. The landlord sets the refunded the amount of the deposit. The tenant decides whether to accept the amount of rented deposit by the landlord. Therefore, the retuned deposit status is divided into 4 statuses: set the refundable deposit amount, waiting for confirmation status, unaccepted status, and accepted status. In this article, because each period of bills needs to be automatically created in the lease contract, it is necessary to send the Ether into the smart lease contract to run the Oraclize service. This step is referred to activating contract status. Figure 2 shows the specific process of the smart lease contract divided into six statuses changes.

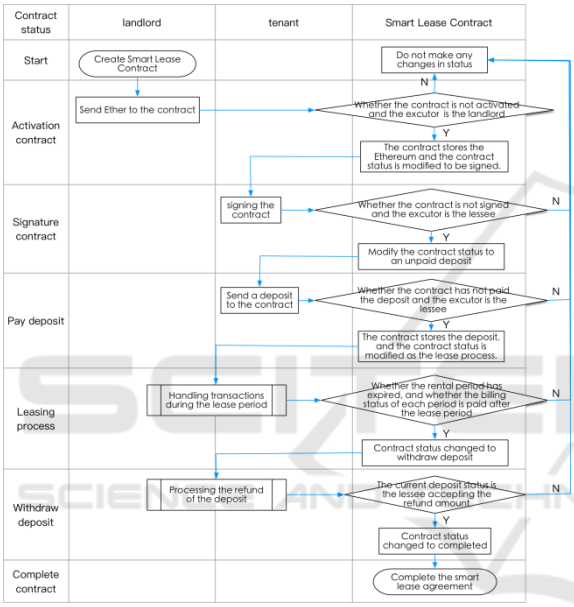

Figure 2. Smart lease contract flow chart.

3.3 Status of the Smart Lease Contract

The operation process of the six statuses of the smart lease contract is as follows:

1. **Activate contract status**: Since the Oraclize service requires gas during the operation, a certain amount of ether must be submitted in the contract. The landlord sends enough ethers to the contract to modify the status of the contract to activated. This operation can only be executed by the landlord and the current contract is not activated.
2. **Sign contract status**: The tenant can check the terms of the contract. If tenant agree to the terms of the landlord's lease agreement, you can sign the contract and change the contract status to signed. This operation can only be executed by the tenant and the current contract is not activated.
3. **Pay the deposit status**: The tenant needs to pay the deposit amount required in the contract to make the contract come into effect and change the contract status to the paid deposit. This operation can only be executed by the lessee and the current contract has not yet paid the deposit.
4. **Rental process status**: This is the most time consuming process for a smart rental contract. After the entry into force of the contract, the lease will automatically create bills of the first period. Each subsequent period will be based on the lease date of the contract as the collection date, and the rental bill for each period will be automatically generated by the Oraclize timing service on a monthly basis. Each period of the rental bill will have four statuses. Figure 3 shows the specific process of the transaction between the landlord and the lessee during the lease period.

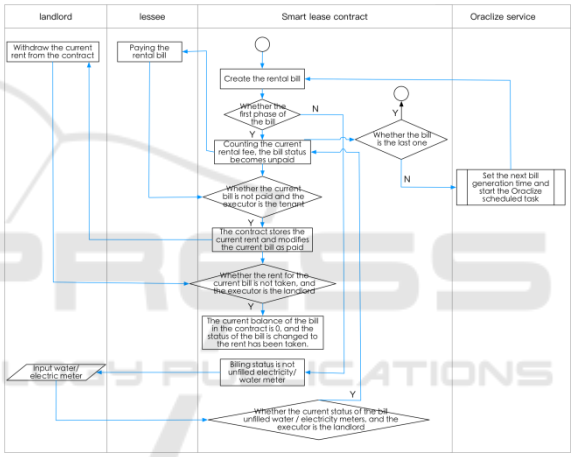

Figure 3. Work flow chart of the smart lease contract during the lease period.

① Pending record meter status: This status is the initialization status of the lease bill. (Note: the rent of the first period paid by the lessee has not yet generated hydropower, so the lease status of the first period is unpaid, and the renter only needs to pay the rent.).

② Settlement status: The landlord records the water meter and electricity meter of the tenant this month. The smart contract will automatically calculate the water and electricity charges used this month, and count the fees payable this month, and update the current bill to settled. This operation can only be performed by the landlord and the billing status is unrecorded meter.

③ Paid Status: The tenant sends an Ethernet to the smart rental contract based on the cost of the settled bill. The smart lease contract stores the

rent sent by the tenant and modifies the billing status to the paid status. This operation can only be performed by the tenant and the lease bill status is the settled status;

④ Take out the rent status: the landlord can take out the rent paid by the lessee in the smart lease contract at any time, and the lease bill status is changed to the status of the rent has been taken out. This operation can only be performed by the landlord and the lease bill status is paid.

When the lease contract has reached the lease term and the lease bills is all in the paid status, the contract participant can change the status of the contract to the return of deposit status.

5. **Return deposit status**: The landlord will deduct the relevant fees from the contract deposit according to whether there is any loss of fixed assets during the lease period. This fee will be negotiated with the lessee, so there are four statuses as follows during the refund of the deposit. Figure 4 shows the flow chart of returned deposit:

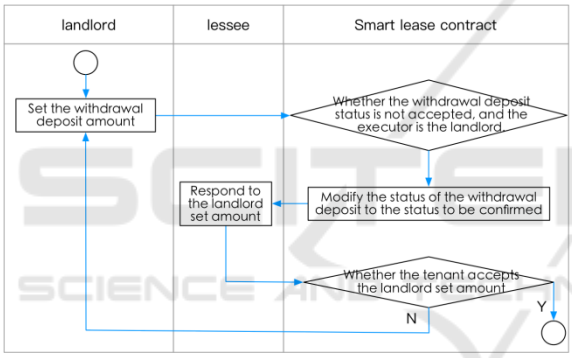

Figure 4. Work flow chart for returned deposit

① Set the refundable deposit amount: The landlord sets the pre-returned deposit and modifies the status of returned deposit to the wait for confirmed status. This operation can only be executed by the landlord and the contract status is not " 5.④".

② Waiting for confirmed status: The tenant can choose whether to accept the amount set by the landlord and modify the current status to unaccepted or accept status, the operation can only be operated by the tenant.

③ Unaccepted status: If the tenant does not accept the amount of returned deposit by the landlord, need to go back to " 5.①" and let the landlord reset the amount of returned deposit.

④ Accepted status: If tenants choose to accept, the contract status changes have been completed status.

6. **Completion status**: The landlord and the lessee have completed the entire lease, and the landlord can retrieve the remaining balance in the lease contract, and the lessee can retrieve the deposit from the contract.

4 ANALYSIS AND EVALUATION

4.1 Case Study

In this section, we will conduct case studies based on blockchain-based housing rental platforms by publishing listings, managing listings, visiting listings, creating lease agreements and all aspects of the rental process.

When a user wants to publish a listing, the user does not need a complicated registration process, and only needs the user to have an Ethereum account and a small amount of Ethereum to store the hash information of the listing information generated by the IPFS. The hash value is stored in the LC, and the smart contract will be shared and used by all nodes to share the listing information.

When users post their own listings, they may renovate their homes, or they may not want to rent their own homes, or the listings are already rented out. The user can call the LC to modify the status of the hash of the apartment on any Ethereum node to change to the unavailable-status or rented-status. If the user wants to re-let the house, he only needs to change the status of the house to the available-status.

Users can access the LC contract on any node of Ethereum and get information about the listings that the Hash has visited. For different roles, access also have some differences. The tenant can quickly obtain a list of all the hashes of the available-status in the LC through the smart contract. The landlord not only has access to all available-status listings, but also quickly obtains the hash of the listings published by himself from the LC to quickly manage the listing hash.

If the tenant finds a suitable listing and will contact the landlord, the landlord can send the lease terms to the LAPC at any Ethereum node to create the SLC contract. The LAPC will automatically save the user's SLC address so that the user can quickly obtain their SLC address.

After the lease agreement is created, the landlord and the tenant will sign the contract and the tenant will pay the deposit. After the lease takes effect, the landlord will calculate the monthly utility fee for each period in the leasing process and add the rent to the total rent for each period, then collect rent. When the lease expires, if the tenant does not violate the lease agreement during the lease period, the deposit will be returned. Otherwise, the landlord will deduct some of

the fee from the deposit. If the tenant agrees, the lease ends. In this system, both the landlord and the tenant will perform every stage during the lease period through the SLC. To ensure the credibility of the bill, the SLC automatically generates a lease bills for each period based on the amount set when the lease contract was originally created. After the bill is created, the landlord only needs to fill in the current water meter and electricity meter to the SLC. The SLC calculates the water fee and electricity fee used this month according to the unit price of the water fee set at the time of the contract creation and the unit price of the electricity bill. These links are recorded in the SLC, and the information in the SLC can only be called by the landlord and tenant in the contract, and the transaction process between the landlord and the tenant during the entire lease period is fully recorded. The SLC can be an effective legal document when a dispute arises between the landlord and the tenant.

4.2 Cost and Practicality Analysis

We have completed the development of a blockchain-based rental system, and the system's smart contract prototype will be compiled and deployed on the Rinkeby network of the Ethereum network. During the analysis period, the exchange rate of Ethereum and US dollar in November 2018 was 1Ethere≈179.56USD. The minimum gas price for executing a trade is 1GWEI, 1GWEI=0.000000001 ETHER. The lower the price of gas, the longer the time to complete the verification transaction. However, the speed of time is not the focus of this paper, and the average time for each interface of the system's smart contract is 18 seconds, so the gas price is set to 1GWEI.

We used two different nodes on the Rinkeby network (Péter, 2017), one node operated by the landlord and the other operated by the tenant. Table 1, Table 2, and Table 3 respectively show the gas and cost of LC, LAPC, and SLC performing operations. Next we will calculate the total cost of executing the three contracts.

We do not calculate the cost of deploying contract usage, only the execution cost of the main functions in each contract. The LC contract executes a total of $0.572 for adding a listing hash and updating the listing hash. LAPC executed the creation of an SLC contract costing $0.8103. This test sets the SLC lease period to 6 periods. Since the first period bill does not need to pay for utility bills, there is no need to record the water/electricity meter. Therefore, the SLC will be executed six times to pay the rent and five times to record the water/electricity meter during the leasing process, for a total cost of $0.4666. The total cost of other operations is $0.1164, which gives the total cost of implementing SLC at $0.583. As a result, the user executes 3 smart contracts, the total cost is about $1.9653, which greatly reduces the extra cost of the lease.

Table 1 The gas and cost of LC performing operations

Operation	Gas Used	Actual Tx Cost (Ether)	USD ($)
Add a listing hash	279311	0.000279	0.0500
Update listing hash status	41177	0.000041	0.0072

Table 2 The gas and cost of LAPC performing operations

Operation	Gas Used	Actual Tx Cost (Ether)	USD ($)
Create an SLC contract	4512562	0.004513	0.8103

Table 3 The gas and cost of SLC performing operations

Operation	Gas Used	Actual Tx Cost (Ether)	USD ($)
Activate contract	66162	0.000066	0.0119
Sign contract	30693	0.000031	0.0055
Pay the deposit	383779	0.000384	0.0690
Record water/electricity meter	429517	0.00043	0.0800
Pay rent	61778	0.000062	0.0111
Complete the rental process	32450	0.000032	0.0057
Set the refundable deposit amount	70723	0.000071	0.0112
Accept or not accept refunded deposit amount	38499	0.000038	0.0068
Retrieve the remaining balance in the lease contract	35012	0.000035	0.0063

5 CONCLUSIONS

The research and application of blockchain technology has shown explosive growth and is considered to be the fifth disruptive innovation in the computing paradigm following mainframes, personal computers, the Internet, and mobile/social networks. It is the fourth milestone in the evolutionary history of human credit following blood credit, precious metal credit, and central banknote credit (Swan, 2015).

Block chain technology enables many impossible possible. The blockchain-based housing rental system has added a good case for the application of blockchain technology. The system uses the blockchain to share the listing information, preventing large companies from having too many users to control the market price. The system realizes a truly decentralized housing rental system, and the system no longer requires a high amount of intermediary fees. This system realizes the real release of peer-to-peer listings and listings. The system converts the traditional leasing method into a way of coding through the Ethereum Smart Contract during the rental period for the landlord and the lessee. The contract can only be run forever in the way it was originally set. The contract permanently preserves all transaction information and records of the lessee and the tenant during the leasing process, which provides legal protection for both the lessee and the landlord.

ACKNOWLEDGEMENTS

This work was supported in part by NSF of China under the grant 61877055 and by NSF of Zhejiang, China under the grant LY18F030013.

REFERENCES

Yu, Y., Dong, Y. & Guo, X., 2018. Pricing for sales and per-use rental services with vertical differentiation. European Journal of Operational Research, 270(2): 586–598.

Zhuang, W., Chen, J. & Fu, X., 2017. Joint dynamic pricing and capacity control for hotels and rentals with advanced demand information. Oper. Res. Lett.

Yu, X.L., Xu, X. & Bin Liu, 2017. EthDrive-A Peer-to-Peer Data Storage with Provenance. CAiSE-Forum-DC.

Wilkinson, S., 2016. Storj A Peer-to-Peer Cloud Storage Network. Available at: https://storj.io/storj.pdf.

Cheng, Z. et al., 2018. Deterministic Proof of Work. arXiv.org, cs.CR.

Gupta, D., Saia, J. & Young, M., 2017. Proof of Work Without All the Work: Computationally Efficient Attack-Resistant Systems. arXiv.org, cs.DC.

Y. Yuan, F.-Y. Wang, 2016. Blockchain: The State of the Art and Future Trends[J]. Acta Automatica Sinica, 42(4): 481-494.

Kim, H.M., Laskowski, M. & Nan, N., 2018. A First Step in the Co-Evolution of Blockchain and Ontologies-Towards Engineering an Ontology of Governance at the Blockchain Protocol Level. CoRR, cs.CY.

Nakamoto S, 2009. Bitcoin: a peer-to-peer electronic cash system. Available at: https://bitcoin.org/bitcoin.pdf.

Ethereum White Paper, 2015. A next-generation smart contract and decentralized application platform. Available at: https://github.com/ethereum/wiki/wiki/White-Paper.

Wood, D., 2014. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Available at: https://gavwood.com/paper.pdf.

David Johnston, 2015. The General Theory of Decentralized Applications, Dapps. Available at: https://github.com/DavidJohnstonCEO/DecentralizedApplications.

Kim, H., M., 2018. A perspective on blockchain smart contracts: reducing uncertainty and complexity in value exchange. Social Science Electronic Publishing, 1-6.

Cong, W., Z., 2018. Blockchain Disruption and Smart Contracts. Social Science Electronic Publishing, 7-10.

Adler, J. et al., 2018. Astraea: A Decentralized Block-chain Oracle. arXiv.org, cs.CR.

de Pedro, A.S., Levi, D. & Cuende, L.I., 2017. Witnet: A Decentralized Oracle Network Protocol. arXiv.org, cs.CR.

Benet, J., 2014. IPFS - Content Addressed, Versioned, P2P File System. CoRR.

Stephen K., Russell M. 2016. A Decentralized Global Multiple Listing Service and Real Estate SmartContract Application. Available at: https://github.com/rexmls/White-Paper-Abstract-/blob/master/Rex%20White%20Paper%20Exec%20Summary%20.pdf.

Dickason, 2016. The Blockchain of Real Estate. Available at: http://midasium.herokuapp.com/smart-contracts.

Feiler, J., 2018. Reading and Writing JSON Data. In Beginning Reactive Programming with Swift. Using RxSwift, Amazon Web Services, and JSON with IOS and macOS. Berkeley, CA: Apress, Berkeley, CA, pp. 31–41.

Péter Szilágyi, 2017. Clique PoA protocol & Rinkeby PoA testnet. Available at: https://github.com/ethereum/EIPs/issues/225.

Swan, M., 2015. Blockchain: blueprint for a new economy. Sebastopol, CA: O'Reilly Media.