Blockchain based IOT platform for smart coffee machines

Sebastian Kanz
Distributed Ledger Technologies
MaibornWolff GmbH
Frankfurt am Main, Germany
sebastian.kanz@maibornwolff.de

André Mundo
Distributed Ledger Technologies
MaibornWolff GmbH
Frankfurt am Main, Germany
andre.mundo@maibornwolff.de

Prof. Michael Braun
Department of computer science
University of Applied Sciences, h_da
Darmstadt, Germany
michael.braun@h-da.de

Abstract—**The blockchain and the internet of things are popular topics for modern research and have many areas of application in industry. Combining these relatively young and highly innovative technologies may be able to unleash their full potential.**
In this paper the synergy between IOT and DLT is shown by the analyzation of a representative, practical IOT use case and the prototypical implementation based on the blockchain platform Ethereum: Manufacturers of smart coffee machines rent their devices via a blockchain based platform to companies, who enable their employees to consume coffee and pay via smartphone. Other parties like service providers or suppliers can integrate their processes and services into the platform. Payment, contract processing of e.g. rental contracts and identities are managed on-chain via smart contracts. To overcome the scaling problem, state channels—more precisely payment channels—are utilized to enable asynchroneous communication and allow offline payment. In addition to solving the scaling problem this usecase's context, it is shown that many IOT use cases require some form of asynchronicity due to device outage or connection loss. With the clever utilization of DLT based payment channels a good solution that addresses asynchronicity and scalability is shown.

Index Terms—**Blockchain, Distributed Ledger Technologies, Internet of Things, State Channel**

I. INTRODUCTION

Manufacturers of household appliances have different opportunities to sell or rent their devices not exclusively to private consumers but also to companies and the gastronomic environment. Transforming those devices into IOT devices by making them "smart" with the concept of a "digital twin" enables manufacturers to utilize new technology and offer new payment models to their customers. This allows the development of an ecosystem in which new business processes can emerge and serveral stakeholders can participate: A platform that connects those smart devices and handles the communication, data persistency, security and the automated processing of the business logic. An evident approach for a suitable solution to create such a platfrom might be the application of blockchain technology—or in general the distributed ledger technology. Due to their characteristics like distribution, no-trust environment, integrity of data and the ability of automated code-processing it is worth investigating on that topic.

A. Pay-as-you-use renting of smart gastronomy devices

The use case investigated in this paper is as follows: Manufacturers offer their smart household appliances on a platform where customers can rent those devices and pay for the usage by a pay-as-you-use payment model [1]. Therefore the devices have onboard sensors which are able to track and analyze the usage and the internal machine state. To communicate with the customer and the platform, the device needs interfaces like Near field communication (NFC) and access to the internet. Furthermore other stakeholders like service providers or suppliers take part on the platform and offer their services. The platform enables all parties to integrate and connect their services and business processes and offers new oppurtunities and use cases. The renting contracts as well as the service contracts—like the service provided by the service provider or the supply contract for supplying customers with goods—are automatically processed by the platform. This includes the communication with the devices and their onboard sensors, the payment for consuming services and goods and the information providing to the involved stakeholders. If the contract contains any conditions or time limits as well as a benefit or penalty system for involved parties, this results in contract actions like value or asset transfer, contract termination, etc.

The present use case focuses on smart coffee machines rented to companies. Imagine the following setup: A company—here in the role of a customer—has an active renting contract with the manufacturer. The supplier is commissioned to deliver the machine as well as optinal goods to the customer. The payment and the delivery tracking are defined through the supply contract, which is processed by the platform. When the supply contract is fullfilled the supplier receives the contractually agreed value from the contract. As soon as the machine is plugged in at the customer's office and connected to the internet, the employees of the customer can order their coffee. They can authorize as company employees and pay the coffee with their company-wallet via a smartphone app—the communication between
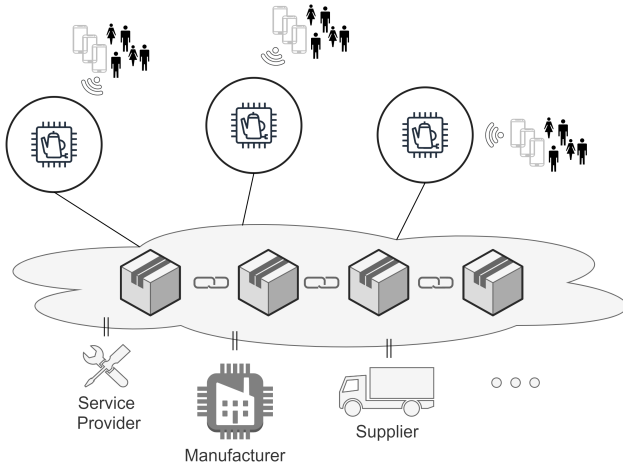
Fig. 1: Use case overview with involved stakeholders

their smartphone and the coffee machine is done through near field communication (NFC). The coffee machine is able to check whether the company has enough credits for the coffee to be served. If the machine requires maintenance or if an error occures the service provider is contacted. This process as well as the payment to the service provider is fully automated through the platform.

Contracts between involved parties can be setup, signed and revoked via an UI if several conditions are met. The platform has some form of identity management to ensure that a user is only acting in a specific role if he or she is allowed to.

II. RELATED WORK

The topics DLT and IOT are popular research areas, and there is also some work on the synergies.

Papers like [2] outline possible IOT use cases that might benefit from DLTs including but not limited to the sectors health, logistics or smart cities. A more technical review on the suitability of blockchain consensus algorithms for the internet of things is done by [3] or [4]. Similar to this work the authors of [5] desibe the synergies between DLT and IOT but evaluate technical aspects like latency, throughput and ressource consumption while not differentiating between different IOT use cases. Others like [6] also investigate on the meaningful application of smart contracts in IOT use cases such as this work.

In contrast to other mentioned papers, who try to find a general DLT solution for the whole topic IOT, this paper identifies use cases and investigates on that topic from a more practical view. The aim is to find a suitable solution for the use case presented in this work as well as related use cases but not cover the complete field of IOT.

III. ARCHITECTURE

The design and the architecture of the present use case requires the platform to implement a set of features. To find a suitable platform solution the key requirements were

elaborated and transferred into the DLT context:

- Smart contracts allow automated information processing on-chain.
- Oracle services provide off-chain information to the DLT.
- Payment is needed for processing invoices of services and goods.
- Asynchronicity empowers IOT use cases.
- Performance is needed by every application and adjusted to the specific use case.
- Encryption needs to be ensured to protect customer data and privacy.

	Smart Contracts	Payment	Oracle Services	Async.	Encrypt.
Bitcoin	(yes)	yes	(yes)	yes	no
BitcoinCash	(yes)	yes	(yes)	no	no
Corda	yes	(yes)	yes	yes	yes
EOS	yes	yes	yes	yes	no
Ethereum	yes	yes	yes	yes	yes
Hyperledger	yes	(yes)	yes	yes	yes
IOTA	no	yes	no	yes	yes
Quorum	yes	yes	yes	yes	yes
Ripple	no	yes	no	yes	no
Stellar	yes	yes	(yes)	yes	no
Tron	yes	yes	yes	yes	no

TABLE I: DLT selection based on key features

As shown in table I, different DLT implementations were elaborated based on the key features mentioned above. The best fitting solution seems to be Quorum, but since this implementation is directly based on Ethereum, the origin solution will be the best choice.

In the following the above mentioned features are embedded into the architectural context of the use case, which is presented in the following and described in more detail where needed. For the prototypical implementation the blockchain protocol Ethereum [7] is used as it provides the necessary features or at least the possibility to implement them.

For the evaluation of security and encryption mechanisms in DLTs reference is made to [?]. Those topics are evaluated in detail in the given references and are not in the focus of this paper. The integration of payment can be done via a third-party or by built-in DLT functionality like Ether or utility-tokens [?].

An important concept that needs to be mentioned is the smart contract which enables this use case to be fully automated and process contracts and contract logic onchain as well as providing oracle services. Due to the fact that most modern DLTs offer smart contract functionality, this topic is not investigated on in detail. The key concept that empowers this use case is the need of asynchroneous communication and information processing. Most IOT use cases have too few ressources for providing user-friendly synchroneous communication. Due to connection-loss, power-outage or less computing ressources, IOT devices can not guarantee a constant communication channel to other devices. Pushing

the concept of asynchronicity a little further and increasing the out time between connections this is referred to as offline capability. To be able to offer the customers a satisfying user-experience, the most critical point is the user interaction—in this use case namely the serving of the coffee. If this process takes to long or aborts due to errors, the customers will decline the use case and the acceptance decreases as well as the economic efficiency.

Therefore the technology of the underlying platform needs a mechanism for this offline functionality: The processing, the payment and the synchronization needs to take place in different time intervals. DLTs as a possible, modern solution offer so called state channels to tackle this issue. With offline communication, hashing and signing algorithms and the utilization of smart contracts DLTs can ensure security, data integrity and payment processing in this scenario. The concept of state-channels is examined in detail in the following section.

A. Excursus: State channels for empowering and scaling IOT use cases

Blockchains are a compromise due to their nature. Due to the consensus protocol, independent of block size and network capacitya, a natural limit is given: For example, the Bitcoin network [8] is limited on a block time of 10 minutes by the complex calculation of Proof-of-Work (PoW). If the block size is very small, blocks are propagated very quickly over the network, but only a few transactions can be transferred at once. If the block size is very large, it is very difficult for nodes to synchronize but more transactions can be transferred at once. This limitation of block size and block duration means that in the case of Bitcoin currently (as of 01/2020) on average about 7 transactions per second are possible [9].

Other implementations may use other consensus mechanisms and other parameters, but there are also natural barriers there. It becomes clear that with increasing demand mainly to transaction processing per time interval (mostly transactions per second (TPS)) an improved performance and new solutions become necessary. [9]

As a possible answer to the scaling problem of blockchains so-called state channels were developed. The aim is to be able to transmit any kind of off-chain processing of status-changing operations, which are typically processed and stored on-chain. Thus the number of of transactions can be reduced and the interaction time between involved parties can be improved. In the context of payments, so-called micro-payments are made possible. State channels, which are limited to payment processing, are called payment channels. These can be faster and cheaper than normal transactions take place. The costs of such micro-payments can be kept very low, since not all transactions must be stored on-chain. [10]

Another big advantage of state channels is the empowerment of asynchroneous transactions. If participants are not in the blockchain network (e.g. due to connectivity problems), then no transactions are carried out. If real-world actions like the opening of a barrier or the processing of an action are dependant from an on-chain status update, then with loss of connectivity the real process will stand still until the connection is restored. State-channels could be used here in order to create an alternative processing that can also be used when the blockchain is not reachable. Some practical example implementations of state channels are Bitcoin's Lightning Network [11], Ethereum's Raiden Network or the implementation of Neo called Trinity [12].

B. Asynchroneous payment processing

The payment-channels in this usecase are a practical realization of one-way payment-channels for IOT devices and implemented as smart contracts on the Ethereum blockchain. When a user submits to a rental offering and the renting contract is created, the corresponding smart contract, that models a rental agreement, creates the data structure on-chain and initializes a payment agreement that is designed by a separate smart contract. This creates the payment-channel between the involved parties and holds all relevant data.

Initially the payment-channel between the coffee machine and the customer is charged with a predefined amount of credits transferred on contract creation. The payment-channel is implemented as a one-way payment-channel, because the value is exchanged only to the wallet of the machine. Credits that remain in the channel after the machine closes the channel will be transferred back to the customer. Apart from the general concept of payment-channels, the present one implements a redeem function that pays out the transferred credits the receivers. Due to this functionality the payment-channels can be reused and recharged and do not need to be recreated; this is for performance and cost issues.

While the payment agreement data structure is an on-chain information container for the payment-channel, the communication between the participants happens off-chain. The data is only needed for data persistency purposes and for synchronization of the participants. To be able to transfer data from the coffee machines to the employees and back there must be a communication channel besides the blochchain. This is implemented by exchanging data via NFC: After pressing the coffee serving button, the coffee machine sends the payment data as a receipt via NFC to the employee's smartphone. The data is signed with the employee's private key and sent back from the smartphone to the coffee machine. The machine is able to verify the signature and persists the latest receipt locally. This process can be repeated as many times as there are enough credits in the payment-channel. Because there might occure a connection-loss the coffee machine might not be able to check the balance on-chain and has to rely on its local information. Because the payment-channels are one-way, there is no possibility for the customer to pay out his own credits and order a new coffee while the machine is offline. This must be ensured to avoid data inconsistency.

If the balance of the payment-channel reaches a critical threshold or a defined time interval is reached, the machine

redeems the receipts and the money is transferred on-chain. The customer—in this use case the company or an authorized employee—can charge the payment-channel again to order more coffee in the future.
The described payment process with the utilization of the payment-channel as part of the renting agreement can be transferred to the other payment processes between more stakeholders like the service provider mentioned above. The base concept and implementation remain the same.

C. Identity verification an privacy aspects

Another important part of the use case is the identity management and verification of the participating users. It is necessary, that each user acts in a well-defined role, as this information is important to use case itself: For example, a coffee machine should only be allowed to trigger certain tasks, while only manufacturers should be able to administer their machines. For that reason, an on-chain smart contract acting as an oracle service is implemented. During an onboarding process, e.g. the registering on the platform, the user is identified and assigned a role, which is persisted in that smart contract. The owner of the smart contract is able to maintain the data as well as adding new roles and identities. To make this apporach more decentralized, many owners could be possible in future. The administrative tasks could be commissioned by a vote system.
The present implementation did not focus on privacy concerns, as they do not belong to the key functionality and can be implemented later. Nevertheless privacy is an important topic to spent on investigation when it comes to a real world adoption. Some privacy vulnerabilities are listed below, without claiming to be complete:

- Third parties are able to track the coffee consumption per customer account. Due to the fact, that in the current implementation an account is used by all employees of an office, no assertion can be made for a single person. The cunsumption is not related to a single day but to a period of time.
- Information of costs, usage, contract periods are visible to third parties.
- Third parties can determine the role of an account, wheter it acts as a machine, a customer, a manufacturer or another role.

Those vulnerabilities can be avoided through the utilization of techniques like zero knowledge proofs [**?**] or providing another identity management solution than an on-chain smart contract. In the context of the present proof of concept, the on-chain solution was preferred because of the easy integration.

D. Architectural overview of involved components

The present use case was evaluated by a prototypical implementation with the Ethereum blockchain. The focus of this was to show the general feasability and the synergy between DLT and IOT. The use case is not covered as a whole as this is not needed to approve the benefits of DLT for IOT use cases.

An overview of the application's architecture is shown in figure 2. On-chain information is processed and stored via three smart contracts: The IdentityProvider takes care of the contracting parties and manages access control. The PaymentProvider holds the payment-channels for the renting contracts and handles payment processing. The RenalProvider models the rental contracts on-chain and controls and calls the other smart contracts.
Information on rentable devices, active contracts, contract handling and payment is accessible by all participants through a web app. The browser-integrated wallet MetaMask is used to sign and verify transactions to and from the blockchain.
To interact with the coffee machine the customer needs a smartphone with enabled NFC service. A mobile app enables the customer to authorize payments and sign incoming receipts from the coffee machine via NFC. The wallet used on the smartphone is the same as the one in the browser, so all information and contracting can be done across device boundaries.
To upgrade coffee machines and make them smart (as long as they do not support smart functionality out of the box) a RaspberryPi mini-computer is used in conjunction with a NFC breakout-board. This board is used for communication between the coffee machine and the smartphone. To measure the internal device state and gather usage data the device needs onboard sensors which must be available for the RaspberryPi to call (e.g. via REST or hardware-near protocols like I2C). The RaspberryPi itself can call the blockchain to redeem the receips or to report errors or request service inquiry.

IV. RESULTS

With the successful implementation of this usecase the general feasability is shown. The following subsections examine different aspects in detail.

A. Performance

The performance from the user's point of view must be divided into two phases of use: The initial contract creation is an asynchronous process that requires the intervention of both contract parties and therefore cannot be measured exactly. If this process is transferred to the analog process—the contract request until the conclusion of the contract—a significant increase in speed can be observed through digitalization of that process. The processing time from the request to the conclusion of the contract corresponds to three onchain transactions and thus lasts on average only a few minutes. The actual usage, i.e. the interaction of the user with the coffee machine, is instantaneous and offers the user optimal performance and thus a good user experience. From the manufacturer's point of view, there are no time-critical processes, as it is not important that a contract is concluded within seconds, as the time at which the customer confirms the contract cannot be affected. A receipt can be cashed out with one onchain transaction within
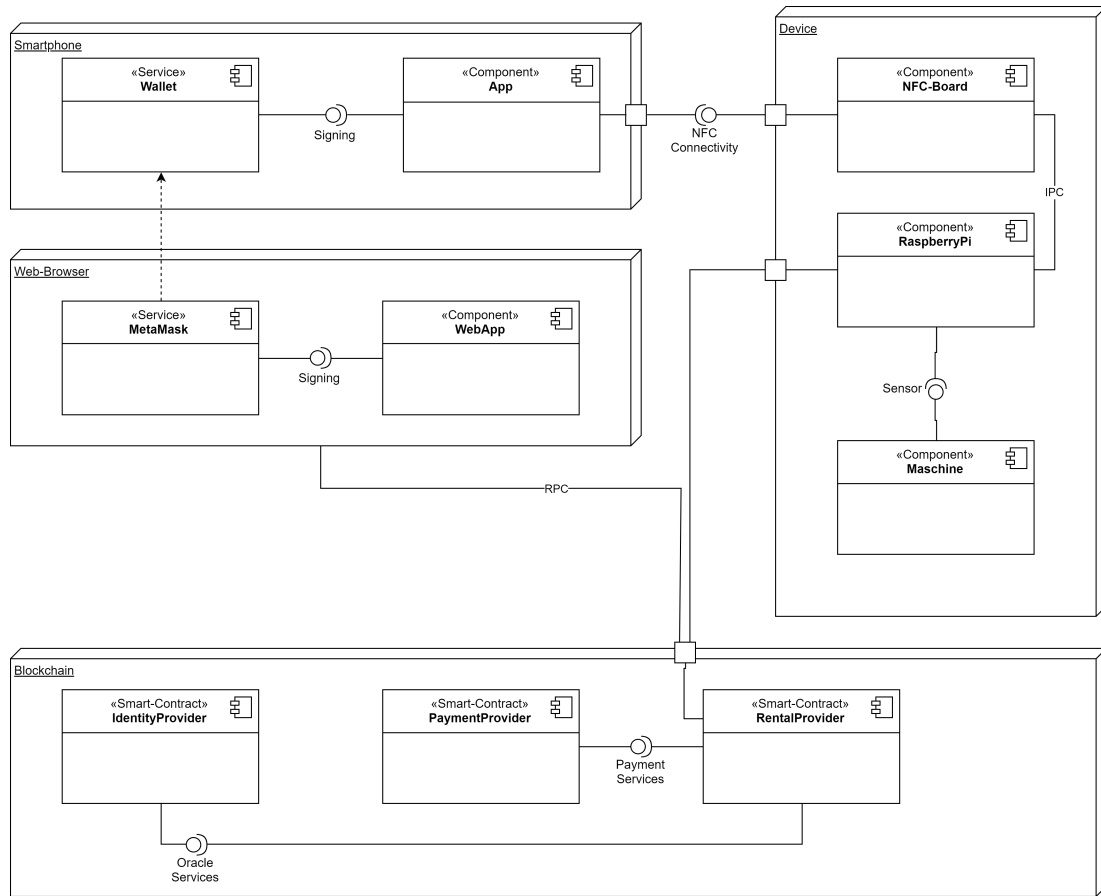
Fig. 2: UML components diagramm; architectural overview

a few seconds (less than a minute) and is therefore within an acceptable time frame.

B. Security

The use case is based on a role and user management system, which checks permissions and restricts access. The central security aspect is the private key of every user. Currently there exists no implemented mechanism to prevent the loss or theft of the of a private key. Furthermore the certification of users as a company's employee is currently only rudimentary implemented and can be improved by utilizing verifiable credentials and decentralized identifiers in that context.

The Ethereum blockchain itself is classified as extremely safe; manipulations or attacks on the system are only possible with extremely high effort and immense costs. Nevertheless, the security of the implemented smart contracts must be audited by experts before a go-live to a productive environment, so that damages caused by third parties or errors in the code can be avoided.

C. Operating costs

The use case is calculated as an industry-realistic quantity structure for the German market based on customer experience of MaibornWolff GmbH. The following numbers and data

are based on the tests from the protoype or marked as assumptions.

A final configuration level of the present use case might include a total of 10.000 coffee machines on rent and 40 employees per office on average as well as a per capita consumption per year of coffee 164 litres.

At the time of calculating the Ethereum price was at 196,04 /ETH; the GAS price is set to 5 GWEI. A more detailed overview of the assumptions made and the underlying data is shown in the appendix.

V. CONCLUSION

The implementation has been successful; a suitable implementation has been made, which prototypically evaluates the use case on the basis of Ethereum. A general statement for the entire IOT application area is thus not made and cannot be fully validated. Use cases that show similar characteristics as the present implementation can profit with a high degree of certainty from a DLT implementation and the associated advantages. On the other hand, time-critical or performance-heavy IOT applications that cannot be implemented on the basis of a DLT because, for example, the required real-time communication is not available. In addition, applications that affect only a single stakeholder or are only executed at a
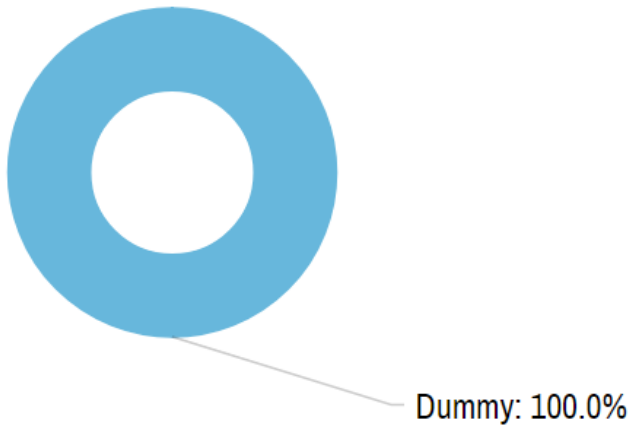
Fig. 3: chart

single location are not designed for DLTs either, as they cannot benefit from the decentralized and no-trust environment.

It is shown, that the suitability of DLTs for IOT use cases from the performance perspective is a tradeoff between on-chain state replication on the one side, where all data is stored on the blockchain and can be reproduced, and off-chain communication on the other side via state channels. For every similar use case the question must be answered, how long a state channel can be kept open without syncing with the blockchain? The more this time can be increased, the less performance consuming due to increased asynchronicity is the IOT use case and the easier it can be implemented in DLTs. If possible, to time should be increased as much as possible provided that the user experience does not suffer from that. On the other side the downside of those time gaps in such use cases must be considered, e.g. the delayed asset or value transfer.

VI. FUTURE WORK

As possible improvements of the shown use case two approaches will be pointed out below.

Privacy is an important topic that was not intentianaly not implemented in detail in this proff of concept as it was not in the focus of the work. Nevertheless, to implement a market-ready IOT use case this topic must be addressed as much as possible to offer a good and secure user-experience. A possible approach and a current research topic are so-called zero knowledge proofs. Without going into detail, a provider of such a proof can show the possessing of a secret (e.g. the birthdate) and reveal that a certain information is true (e.g. to be old enough to buy alcohol) without sharing the secret with the requester. The requesting entity can check and verify the proof mathematically. With that technique, privacy can be improved in that use case e.q. by replacing the on chain identity smart contract. Further research in this area might show more potential.

On the other side the costs of a industry use case like the one

shown in this paper is a relevant factor to consider. To further reduce the costs, one approach might be the outsourcing of on-chain saved data to the off-chain world (e.g. like IPFS), because on-chain data storage is expensive. As IPFS and similar solutions are still under heavy development, there might be a suitable solution to improve this use case.

REFERENCES

[1] winterhalter. (2020, apr) Pay per wash. [Online]. Available: https://www.pay-per-wash.biz/de_de/

[2] T. M. Fernndez-Carams and P. Fraga-Lamas, "A review on the use of blockchain for the internet of things," *IEEE Access*, vol. 6, pp. 32 979–33 001, 2018.

[3] M. Salimitari and M. Chatterjee, "A survey on consensus protocols in blockchain for iot networks," 2018.

[4] R. Han, V. Gramoli, and X. Xu, "Evaluating blockchains for iot," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Feb 2018, pp. 1–5.

[5] M. Maroufi, R. Abdolee, and B. M. Tazehkand, "On the convergence of blockchain and internet of things (iot) technologies," *CoRR*, vol. abs/1904.01936, 2019. [Online]. Available: http://arxiv.org/abs/1904.01936

[6] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[7] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," 2013. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[9] M. Macdonald, L. Liu-Thorrold, and R. Julien, "The blockchain: A comparison of platforms and their uses beyond bitcoin," 02 2017.

[10] J. Coleman, L. Horne, and X. Li, "Counterfactual: Generalized state channels," 2018. [Online]. Available: http://l4.ventures/papers/statechannels.pdf

[11] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016. [Online]. Available: https://lightning.network/lightning-network-paper.pdf

[12] J. XU, Z. Ji, and Y. Li, "An off-chain scaling solution for neo," 2018. [Online]. Available: https://trinity.tech/#/writepaper