University of Plymouth

PEARL **https://pearl.plymouth.ac.uk**

Faculty of Science and Engineering School of Engineering, Computing and Mathematics

2020-03-28

Synergy of Trust, Blockchain and Smart Contracts for Optimization of Decentralized IoT Service Platforms

Shala, Besfort

http://hdl.handle.net/10026.1/17139

10.1007/978-3-030-44041-1_49
Springer International Publishing

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

Synergy of Trust, Blockchain and Smart Contracts for Optimization of Decentralized IoT Service Platforms

Besfort Shala[1, 2], Ulrich Trick[1], Armin Lehmann[1], Bogdan Ghita[2] and Stavros Shiaeles[2]

[1] Research Group for Telecommunication Networks, Frankfurt University of Applied Sciences, Frankfurt/M., Germany
[2] Centre for Security, Communications and Network Research, University of Plymouth, Plymouth, UK
shala@e-technik.org

Abstract. There is a high potential of creating new and market-ready services for the Internet of Things (IoT) community by enabling end-users to make their resources inside the end-user environment available to others. Thus, end-user friendly service platforms facilitate the process of designing and configuring new local and cooperative services. However, decentralized networks of end-users acting as service providers enable the risk of uncontrollable behavior and trustless activities. Blockchain technology coupled with smart contracts now gain popular attention by providing benefits of data immutability and process automation in open or closed communities. This research summarizes several blockchain-based IoT approaches, concluding with their benefits and limitations. Besides them it proposes an optimized and decentralized IoT marketplace consisting of end-user friendly IoT service platforms, forced by a synergy of trust, blockchain and smart contract elements. To highlight the benefits of this synergy, two smart contract use cases (based on Ethereum) in the IoT marketplace are presented.

Keywords: Blockchain, Internet of Things, Service and Application, Trust, Smart Contract, Ethereum, Peer-to-Peer, Distributed Ledger Technology

1 Introduction

The increasing amount of intelligent devices in end-user environments highlights the focus to include end-users in the service provision process in order to enhance the competition in the IoT marketplace and to provide more service variety to the community. An end-user based IoT service platform is presented in [15], which consists of a fully decentralized architecture and enables every end-user to easily design/configure IoT services for others in the community. However, completely decentralized communities, where all nodes are connected Peer-to-Peer (P2P) with each other suffer from missing trust relationships among the nodes. This results in trustless behavior regarding service and data integrity, service order and service handling, and/or service functionality.

Blockchain, as part of the Distributed Ledger Technology (DLT), provides an immutable database, where every participating node is storing a copy of it and where the

integrity of the database is ensured using cryptographically principles. In this context, smart contracts, as automatically executed code (condition based) and stored within the blockchain, gain high popularity in order to automate several processes in an ecosystem. The potential of blockchain combined with smart contracts and their different integration possibilities in IoT are discussed in several publications [1-6]. However, none of them are providing a comprehensive and trust-based approach to optimize the service provision process in IoT.

The aim of this paper is to optimize the service provision process in IoT environments through the synergy of trust, blockchain, and smart contracts. The optimization approach enables interesting nodes to securely participate in a completely decentralized community and to provide trustworthy services to/from others. Thus, this paper reviews several existing IoT approaches and defines requirements for an optimal decentralized IoT platform. The main contribution of this publication is the integration of smart contracts in end-user based IoT service provision environments with the combination of a blockchain-based trust management system and a trust consensus protocol used for fair decision-making by enabling high trusted negotiations between the participants. This paper is structured as follows: Section 2 presents a review of several existing blockchain-based IoT approaches, highlighting their limitations, which are aimed to be optimized through the novel synergy approach presented in this publication. Section 3 introduces an overview of the decentralized and blockchain-based IoT marketplace [15, 16], which consists of end-user based IoT platforms using a blockchain-based trust framework to ensure trust in the community. Smart contracts and their power of process automation are introduced in Section 4. Moreover, two use case scenarios are presented where smart contracts are integrated in the service provision process inside the presented decentralized IoT marketplace. Finally, section 5 gives a conclusion of the presented framework and highlights the benefits of the presented synergy.

2 Related Work for Blockchain-Based IoT Environments

An optimal decentralized IoT platform, which is part of an IoT marketplace, should fulfil the following initially defined requirements. First, the whole environment including the services providers, consumers and the services should consider trust elements providing a **trusted environment (1)**, whereas a decentralized network could lead to trust issues and uncontrollable behavior of single nodes. Second, the IoT marketplace should be fully decentralized avoiding single entities or super nodes maintaining a part of the network. This avoids single point of failures or monopoly of nodes within the marketplace. In relation to a **decentralized architecture (2)**, the blockchain used in the marketplace should also avoid central instances and give every node the possibility to participate in blockchain processes, such as transaction sending and validation, and block creation. Another important element is the consensus protocol used to agree for the same copy of the ledger among the nodes. The consensus protocol should consider the specific lightweight characteristics of IoT devices and should also provide an objective, fair, and reliable decision-making process considering the decentralized character of the marketplace where different nodes are acting (**reliable and lightweight**

consensus protocol (3)). As mentioned in section 1, end-user environments are considered a powerful potential to provide new or market ready services to the community in a decentralized IoT marketplace (**end-user-based services (4)**). Additionally, all participant nodes in a blockchain-based IoT marketplace should be motivated to participate in a fair and trustful way in all related tasks. Thus, **incentive mechanisms (5)** should be considered to avoid passive or manipulative behavior in the community. Another important requirement is to decide whether to do the **service handling (6)** (means that the service consumer gets the service data (e.g. temperature value) from the service provider after the conditions of the contract are fulfilled) inside the blockchain or outside. Doing the service handling on-chain often leads to privacy issues as all details of the service execution are stored inside the blockchain and are open to all other blockchain participants. Another element is where all the service execution data (e.g. temperature values) in the IoT marketplace is stored. **Data storage (7)** possibilities include centralized storage nodes, or P2P storage including Distributed Hash Tables or/and blockchain storage. Finally, there are two **blockchain types (8)** in the context of blockchain which are used in general and which should be considered for a decentralized IoT marketplace: public blockchains, where every node can participate in the network and the distributed ledger is distributed to all participants; private blockchains, where blockchain processes are maintained by a specific group of nodes (leads to centrality).

The literature review shows that there is an increasing amount of publications dealing with smart contracts and blockchain in the context of Internet of Things. However, a considerable part of it is dealing with optimizing access controls in IoT using smart contracts [7]. Another part is discussing the integration of smart contracts for other use cases, such as for automating business process workflows or service chain handling in IoT. In the following the most recent and relevant publications regarding blockchain-based IoT environments are selected for review.

The authors in [8] propose a smart-contract-based process execution in IoT environments. Therefore, they decide to encode business processes by smart contracts to build trust among untrusted business partners and external IoT services and to enforce transactions among the participants. To avoid the limitations of existing consensus protocols, the authors in [8] propose an alternative Practical Byzantine Fault Tolerance (PBFT)-based consensus protocol.

Another approach for using smart contracts to automate processes for IoT applications is presented in [9]. Therefore, the authors propose to use blockchain technologies and smart contracts in healthcare applications to create secure and private healthcare alerts. To avoid energy constraints of Proof of Work, the authors proposed to use the Proof of Authority (PoA) consensus protocol. Moreover, they grouped the network in clusters lead by a cluster head which maintains the cryptographically keys. Smart contracts are used to allow sellers to register their products in the blockchain and to enable buyers/sellers the ratings of each other.

The authors in [10] introduce a blockchain-based supply chain management system in the Industrial Internet of Things (IIoT) and a penalty-based smart contract solution for fair goods exchange. Blockchain is integrated as a P2P network where smart contracts are running and can be used for judgment between parties to realize a trade. If the

smart contract is not correctly fulfilled by the parties, they will be penalized for their misbehaving actions.

A decentralized data marketplace for smart cities using blockchain technologies and smart contracts is introduced in [11]. Therefore, product sellers are storing a description of their product in a distributed file storage (DFS) and in the blockchain (here only the metadata). A buyer can check the blockchain for specific metadata and can use this information to request a specific product description at the DFS. Afterwards, the buyer can pay for the product and will get the data from the buyer outside the blockchain. The payment is done using a blockchain-based streaming data payment protocol.

The authors in [12] propose a blockchain-based smart home system where a private blockchain is used to store policies. The policies are used in smart contracts for data flow or transaction management between the participating nodes. A local storage at every smart home miner is used to storage service data (sensor values).

A blockchain-based platform embedded with smart contracts for IIoT consisting of an on-chain network for transaction handling and an off-chain network (using Distributed Hash Tables among the nodes) for storage and data processing is presented in [13]. The smart contracts are used for agreements between service consumers and manufacturing services and for creating a trusted data resource trading platform.

Another approach is presented in [14], where smart contacts are used to eliminate trust between service providers and service consumers in a blockchain-based decentralized service marketplaces for IT services. Therefore, smart contracts are introduced to set the rules of interactions between providers and consumers. In case of not fulfilling the rules, smart contracts will trigger punishments for the participating misbehaving entity. Moreover, the authors in [14] propose to use supporting actors to support trustless intermediation in the decentralized marketplace. Supporting actors act as monitoring agents checking the service availability and storing the monitoring results in a smart contract. They are incentivized by payments done for their actions.

The existing approaches [8-14] presented above provide interesting points regarding the integration of smart contracts and blockchain in IoT. However, the overall evaluation of them shows that they are not fulfilling most of the previously defined requirements. For instance, only the approaches in [11, 14] are considering trust respectively rating mechanisms among the participating nodes to increase the trust relationships in the network. However, they are missing to provide a decentralized and tamper-proof trust evaluation system. Another drawback is that most of the reviewed approaches are using semi-decentralized architectures [8-10, 12-13] where some nodes are selected to maintain the blockchain activities, or to monitor activities in the private blockchain [8-9, 12-13]. Only the authors in [11] present a fully decentralized architecture for blockchain activities. However, the paper lacks the information on the type of the used blockchain. Furthermore, none of the reviewed approaches are presenting a reliable and lightweight consensus protocol. The approaches presented in [8, 9] try to optimize the consensus process by introducing an extended version of the PBFT [8] or by using the PoA consensus [9]. However, both approaches lead to centralized elements and avoid using trust elements in the decision-making process. Despite them, none of the reviewed approaches are considering end-users in the service creation and deployment process. Less of the reviewed publication [10-11, 14] are doing the service handling outside the

blockchain by providing in this way more privacy to the participants. Finally, only a few of the approaches are dealing with incentive mechanisms for participants through penalizing actions [10], token rewards [13], or payment rewards [14].

Table 1. Evaluation of existing blockchain-based IoT approaches

Requirements	[8]	[9]	[10]	[11]	[12]	[13]	[14]
Trusted Environment (1)	-	-	-	o	-	-	o
Decentralized Architecture (2)	-	-	o	+	-	o	o
Consensus Protocol (3)	o	o	-	-	-	-	-
End-User based Services (4)	-	-	-	-	-	-	-
Incentive Mechanisms (5)	-	-	+	-	-	+	+
Service Handling (6)	on	on	off	off	on	on	off
Blockchain Type (7)	private	private	n/a	n/a	private	private	public
Data Storage (8)	n/a	c/b	n/a	d	c	d/b	d

Assessment notation: + satisfied, o partial satisfied, - not satisfied, n/a not applicable, c centralized, d DHT, b blockchain

Table 1 summarizes the strengths and weaknesses of existing blockchain-based IoT approaches regarding special requirements, which should be considered in a decentralized IoT marketplace.

3 Decentralized and Blockchain-Based IoT Marketplace

3.1 Blockchain Technology

Blockchain provides a secure and decentralized database that can be shared across a network of multiple nodes (initiated in Bitcoin [23]). All nodes participating in this network will have the same copy of the ledger, which is maintained and secured through cryptographically principles, which among others are incorporated in so called consensus mechanisms. Consensus mechanisms are used to agree for the same copy of the ledger between the nodes. Blockchain technologies provide benefits to a network, such as decentralization, transparency, and immutability. Thus, many application fields are considering the integration of blockchain to optimize their overall state of the system.

Despite its benefits, blockchain is also facing some limitations regarding the inefficiency and untrustworthy features of existing consensus mechanisms. The existing drawbacks are solved by a newly introduced consensus protocol in [16] called Trust-Consensus Protocol, which is integrating trust elements in all steps beginning from sending transactions, selecting the block creator, validating the new block, and reward/punish the participants for their behavior. A detailed description of this novel protocol can be found in [16]. The Trust-Consensus Protocol is not only used to improve the blockchain functionality, but also to improve several aspects in a service and trust management platform (also described in the next subsections).

3.2 End-User based IoT Service Platform

To enable end-users in IoT environments to share services (associated with their local resources) with others in the network, the authors in [15] introduce a fully decentralized service platform where different M2M device technologies are integrated and can be combined with each other. Every end-user can utilize a user-friendly Graphical User Interface which enable the design and configuration of single or cooperative services. The design and configuration of services will result in a machine-readable State Chart XML (SCML) service description which is used by the involved service instances to set up their service for execution. Moreover, the configuration of a cooperative service will lead to an automated and autonomous connection of the specific service instances involved in that configuration. The authors in [15] also introduce a P2P network which is used for communication and information storage between the participating nodes. Additionally, the P2P network is associated with a community for social networking among the participants. To improve the bootstrapping process of new nodes, the service registration of new services, and the configuration of cooperative services, the authors in [19] propose to use the blockchain-based Trust-Consensus Protocol, which ensures that only trusted nodes do democratically the decision-making process in the network.

3.3 Tamper-Proof Trust Management

Trust is a very important component to secure IoT environments by building trust relationships between the nodes. Therefore, the authors in [16, 17, 22] introduce a decentralized and community-based trust evaluation system where every participating node of the network act as a trust agent by evaluating the trustworthiness of others. The proposed trust model includes the evaluation of the services provided by a node and the evaluation of the node behavior. The service evaluation consists of service functionality, service quality and service acceptance. The node behavior consists of participation willingness in several community tasks and the data/service integrity whether the node has something changed intentionally to harm the network. To securely store the evaluated trust scores, the authors in [18] propose the utilization of a blockchain to benefit from its tamper-proof feature.

4 Smart Contract for Service Provision Optimization

4.1 Smart Contracts

The originator of Ethereum [20, 21] defines smart contracts as “systems which automatically move digital assets according to arbitrary pre-specified rules” and provide therefore a suitable blockchain-based framework to create and deploy them. Ethereum is providing a decentralized Turing-complete virtual machine where smart contracts are executed in virtual machines running in every blockchain node. A smart contract is a piece of code that runs in every participating blockchain node (acting as miner) and defines conditions or rules that should be fulfilled in order to trigger actions. Smart contracts are used to create agreements between different entities without relying on a centralized entity or used to set conditions for getting or using data/services. The smart

contract execution is triggered by a contract call which is a transaction containing the function to execute the smart contract. All participating miners receive this transaction and the execution of the smart contract occurs when the miners include this transaction in a new block together with other transactions during their validation. The winning miner propagates the new block to other nodes, which are validating the transactions included in the block (validation includes also the execution of the smart contract). It is also considered to create a chain of smart contracts, where for instance one smart contract can trigger another one through so called message calls. [20, 21]

4.2 Smart Contracts in Decentralized IoT Marketplaces

Although smart contracts are highlighted as self-executed codes, which are stored tamper-proof in the blockchain and which enable trustless intermediation between entities, they can only be powerful in fully decentralized networks by including decentralized trust between the contract partners. This publication proposes to combine smart contracts and trust in order to enable high trusted service handling between service providers and service consumers. Thus, smart contracts will consider checking the trust scores of the participating entities, defining in that way the condition that only trusted nodes can get or provide services. Moreover, the introduced Trust-Consensus Protocol enables only high trusted nodes to participate in the mining process. This reinforces the credibility of their decision respectively their mined blocks which concludes with the outcomes of a smart contract execution. Combining smart contracts with a comprehensive trust model and a trust-based consensus protocol provide a democratic and high trusted decision-making process in a decentralized IoT marketplace. To overcome privacy issues of existing blockchain-based approaches, this publication proposes to perform the service handling outside the blockchain. The blockchain and smart contracts are used to verify whether the conditions are fulfilled (e.g. trust score is above a predefined level) between the service partners.

To rise up the importance of integrating the synergy of trust, blockchain and smart contract in the service provisioning process, the next two subsections define two exemplary use cases.

4.2.1 Use Case 1: End-User Acting as Service Provider

A Service Provider B announces a new service called temperature service to the community. Besides those, B creates a smart contract defining the conditions for other users to get access to that service. Service Consumer A wants to use the new temperature service provided by B. The smart contract part of the service handling is shown in Figure 1. Service Consumer A sends (1) a transaction to the address of the recently published smart contract temperature service where A requests access to the service. All nodes part of the blockchain network will receive this unconfirmed transaction and will store it in their local storage. The next step is the consensus mechanism which happens in the network. For consensus, the Trust-Consensus Protocol is applied which firstly determines the block creator (2) based on the trust score of the participating peers. After a peer is selected as a block creator, it validates the collected transactions which are

used to create a new block. Transactions sent to a smart contract will trigger the execution of the respective smart contract code. In this case, the block creator will execute (3) the code of the smart contract for getting access to the temperature service. The smart contract defines that only users with a high trust score and which have done the payment for the service will get access to that service. The smart contract execution will check the trust score of the requesting node (by getting the trust scores stored in the blockchain, which are derived using the trust management system described in section 3) and the fulfillment of the conditions (4). The outcomes of the smart contract execution (5) will be included in the block by the block creator (6). The new created block will be forwarded to all blockchain nodes for verification (7). If everything is in line with the block requirements, the block will be accepted by all the nodes and the transactions in it will be valid (8). The updated blockchain results with an event (additional information: it is possible to store the event message as a log in the blockchain, where nodes can check the outcome of the smart contract execution) sent to the contract entities (A and B) with the service usage permission (9). Afterwards, the service consumer A and the service provider know that all conditions are fulfilled and the service handling outside the blockchain can start. Thus, A will send (10) a SUBSCRIBE out of the blockchain (off chain) to B in order to request the temperature service. Service provider B will check the SUBSCRIBE and if the requirements are fulfilled (if there exists a service usage permission for service consumer A) it will response (11) positively to the request with a NOTIFY by providing information about the temperature.

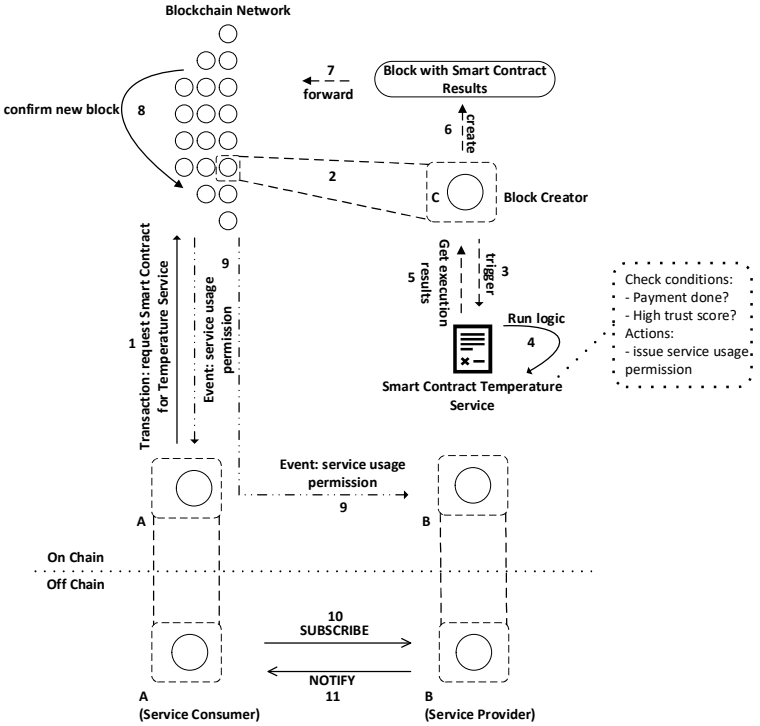

Figure 1: Service handling between Service Consumer and Service Provider

4.2.2 Use Case 2: End-User Designs Cooperative M2M Application Service

An end-user is using a Graphical User Interface (GUI) (part of the Service Creation Environment) to create an Application Building Surveillance service (cooperative M2M application service) which is a combination of three services: temperature service, monitoring service and alarm service. Any service consumer can register to this cooperative service in order to be alarmed for a specific temperature value in a specific room. The temperature service is providing sensor values for a specific room. The monitoring service evaluates sensor values by determining the consumers that should get alarmed. The alarm service provides a messaging service by alarming the consumers via a message for the specific temperature value.

After the end-user creates a cooperative M2M application service using its GUI, the system will generate an SCXML application description where the configuration of the three services is described. Each of the participating service will receive its part of the configuration and will start to set up its service. At the same time the three involved services respectively their service providers will create an overall smart contract called Smart Contract Application Building Surveillance. Here the service providers define the conditions which should be met by the service consumer in order to be able to use these services. They define the trust score of the consumer and the payment as a condition that should be done by the consumer. Additionally, each of the service provider will create an individual smart contract defining the conditions how a service can be used by another service. This process will create three smart contracts for each of the three services (Smart Contract Temperature Service, Smart Contract Monitoring Service, and Smart Contract Alarm Service).

The smart contract part, which is part of the service handling, is shown in Figure 2. Service Consumer D wants to use the Application Building Surveillance, therefore D needs to send a transaction to the Smart Contract Application Building Surveillance requesting to use that service (1). All nodes will receive the transaction and will store them in their local storage. Using the Trust-Consensus Protocol one of the nodes will be selected as a Block Creator (2). The Block Creator will start creating a block by including collected transactions in it. The process of creating a block includes also the process of executing the smart contract codes which are triggered by the transactions, in this case by the transaction sent from consumer D (3). The code execution of smart contract Application Building Surveillance will run the logic to check if the conditions by the consumer are fulfilled (4). These conditions are: Has the consumer made the required payment and has the consumer had a high trust score? If the conditions are fulfilled, the transaction will trigger a message call with the payment confirmation and information about the service consumer (5). This message call will trigger the execution of the three individual smart contracts Temperature Service, Monitoring Service, and Alarm Service (6). Each of these three smart contract executions will check the received details. The smart contracts Monitoring Service and Alarm Service will trigger a message call to the appropriate smart contract for requesting the temperature service respectively the monitoring service (7). Afterwards, the smart contracts Temperature Ser ice and Monitoring Service will be triggered to run their logic defined in the code (8).

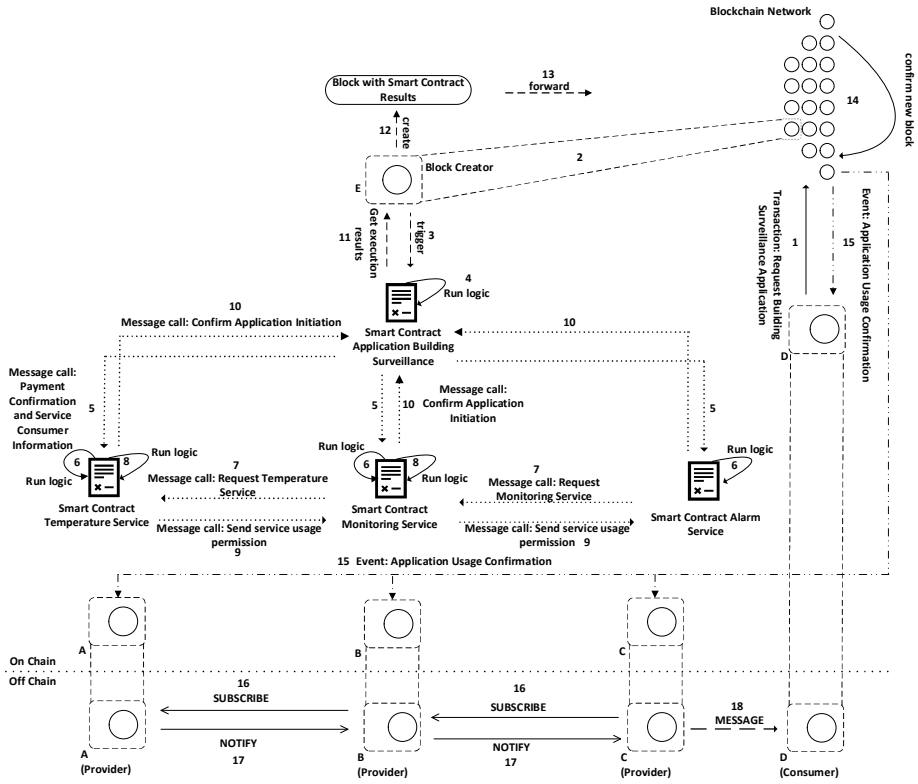

Figure 2: Smart Contracts for Cooperative IoT Services

The logic run checks in this use case the conditions if the requested service is a trustworthy service or not. If yes, it will send a Service Usage Permission (9), notifying the service that it can request service deliverables. If not, it will deny the request. Then, a confirm application initiation event will be sent by all the three individual smart contracts to the smart contract application building surveillance (10). The block creator will get the execution results (11) and will start creating a new block including these details and other information in the block (12). The new block will be forwarded to the blockchain network (13) where other nodes verify the new created block and confirm their acceptance to add this block to the blockchain (14). This will lead to an update of the same copy of the ledger for all participating nodes in the blockchain network. The new entries in the blockchain will contain the information that the end-user is able to use the services and the three services also can use each other. These results will be sent as an event to the service providers (15). Afterwards, the application building surveillance is running, where service provider B requests via SUBSCRIBE (16) the service A, and service provider C from B. These subscribe requests will get a positive response (NOTIFY) (17), as in the blockchain layer the service usage confirmation is already defined. Afterwards, the service consumer will receive a MESSAGE in order to be informed for a specific temperature value in the room (18).

5 Conclusion

Decentralized networks with end-user-based IoT services are in common and provide service flexibility and variety for service consumers. However, a completely decentralized community is facing problems regarding trust and security because of missing centralized entities to monitor the overall behavior of the nodes. Thus, blockchain and its features enable a trustless intermediation between the nodes where transactions and data are stored tamper-proofed in a distributed ledger. However, a missing component remains trust and the trust relationships between the nodes in the community.

This paper reviews several existing blockchain-based IoT approaches and highlights their limitations. To overcome several elaborated issues, this publication proposes the synergy of trust, blockchain and smart contracts to optimize the overall IoT service provisioning process and to enhance the trust level in a decentralized community. The trust scores are securely stored in the blockchain and smart contracts are introduced to automate the service provisioning process. Moreover, a comprehensive trust model is used to derive trust scores among the participants. These trust scores are incorporated to the smart contracts and are used to realize trusted intermediation between the nodes. Finally, the service conditions are checked on-chain whereby the service handling is done outside the blockchain in order increase the privacy level in the community.

References

1. Deloitte, Riddle & Code: IoT powered by Blockchain: How Blockchains facilitate the application of digital twins in IoT. (2018). https://www2.deloitte.com/content/dam/Deloitte/de/Documents/Innovation/IoT-powered-by-Blockchain-Deloitte.pdf, last access 2019/12/16.
2. Dai, H., Zheng, Z., Zhang, Y.: Blockchain for Internet of Things: A Survey. IEEE Internet Things J. 6, 8076–8094 (2019).
3. Hanada, Y., Hsiao, L., Levis, P.: Smart Contracts for Machine-to-Machine Communication: Possibilities and Limitations. In: 2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS). pp. 130–136 (2018).
4. Fotiou, N., Polyzos, G.C.: Smart Contracts for the Internet of Things: Opportunities and Challenges. In: 2018 European Conference on Networks and Communications (EuCNC). pp. 256–260 (2018).
5. Christidis, K., Devetsikiotis, M.: Blockchains and Smart Contracts for the Internet of Things. IEEE Access. 4, 2292–2303 (2016).
6. Xie, J., Tang, H., Huang, T., Yu, F.R., Xie, R., Liu, J., Liu, Y.: A Survey of Blockchain Technology Applied to Smart Cities: Research Issues and Challenges. IEEE Commun. Surv. Tutorials. 21, 2794–2830 (2019).
7. Riabi, I., Ayed, H.K.B., Saidane, L.A.: A survey on Blockchain based access control for Internet of Things. In: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). pp. 502–507 (2019).
8. Xu, L.D., Viriyasitavat, W.: Application of Blockchain in Collaborative Internet-of-Things Services. IEEE Trans. Comput. Soc. Syst. 6, 1295–1305 (2019).

9. Dwivedi, A.D., Malina, L., Dzurenda, P., Srivastava, G.: Optimized Blockchain Model for Internet of Things based Healthcare Applications. In: 2019 42nd International Conference on Telecommunications and Signal Processing (TSP). pp. 135–139 (2019).
10. Alahmadi, A., Lin, X.: Towards Secure and Fair IIoT-Enabled Supply Chain Management via Blockchain-Based Smart Contracts. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC). pp. 1–7 (2019).
11. Ramachandran, G.S., Radhakrishnan, R., Krishnamachari, B.: Towards a Decentralized Data Marketplace for Smart Cities. In: 2018 IEEE International Smart Cities Conference (ISC2). pp. 1–8 (2018).
12. Aung, Y.N., Tantidham, T.: Review of Ethereum: Smart home case study. In: 2017 2nd International Conference on Information Technology (INCIT). pp. 1–4 (2017).
13. Bai, L., Hu, M., Liu, M., Wang, J.: BPIIoT: A Light-Weighted Blockchain-Based Platform for Industrial IoT. IEEE Access. 7, 58381–58393 (2019).
14. M. Klems, J. Eberhardt, S. Tai, S. Härtlein, S. Buchholz, A. Tidjani: „Trustless Intermediation in Blockchain-based Decentralized Service Marketplaces", In: Maximilien M., Vallecillo A., Wang J., Oriol M. (eds) Service-Oriented Computing. ICSOC 2017. Lecture Notes in Computer Science, vol 10601. Springer, Cham
15. Steinheimer, M., Trick, U., Fuhrmann, W., Ghita, B., Frick, G.: M2M Application Service Provision: An autonomous and decentralised Approach. Journal of Communications Vol. 12, no. 9, pp. 489-498 (2017).
16. Shala, B., Trick, U., Lehmann, A., Ghita, B. and Shiaeles, S.: Novel Trust Consensus Protocol and Blockchain-based Trust Evaluation System for M2M Application Services. Internet of Things – Engineering Cyber Physical Human Systems, Elsevier Journal (2019)
17. Shala, B., Trick, U., Lehmann, A., Ghita, B., Shiaeles, S.: Trust-Based Composition of M2M Application Services. In: International Conference on Ubiquitous and Future Networks, ICUFN (2018).
18. Shala B., Trick U., Lehmann A., Ghita B., Shiaeles S.: Blockchain-Based Trust Communities for Decentralized M2M Application Services. In: Xhafa F., Leu FY., Ficco M., Yang CT. (eds) Advances on P2P, Parallel, Grid, Cloud and Internet Computing. 3PGCIC 2018. Lecture Notes on Data Engineering and Communications Technologies, vol 24. Springer, Cham
19. Shala B., Trick U., Lehmann A., Ghita B., Shiaeles S. (2020) Trusted, Decentralized and Blockchain-Based M2M Application Service Provision. In: Barolli L., Hellinckx P., Enokido T. (eds) Advances on Broad-Band Wireless Computing, Communication and Applications. BWCCA 2019. Lecture Notes in Networks and Systems, vol 97. Springer, Cham
20. Buterin, V.: A Next Generation Smart Contract & Decentralized Application Platform. Ethereum Whitepaper. http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf, last access 2019/12/13.
21. Wood, G.: Ethereum: A Secure Decentralised Generalised Transaction Ledger. Yellowpaper. (2019). https://ethereum.github.io/yellowpaper/paper.pdf, last access 2019/12/16.
22. Shala, B., Wacht, P., Trick, U., Lehmann, A., Ghita, B., Shiaeles, S.: Trust integration for security optimisation in P2P-Based M2M applications. In: Proceedings - 16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 11th IEEE International Conference on Big Data Science and Engineering and 14th IEEE International Conference on Embedded Software and Systems (2017).
23. Nakamota, S.: Bitcoin: A Peer-to-Peer Electronic Cash System. (2008). https://bitcoin.org/bitcoin.pdf, last access 2019/12/16.