Blockchain-Based Trust Communities for Decentralized M2M Application Services

Shala, Besfort; Trick, Ulrich; Lehmann, Armin; Ghita, Bogdan; Shiaeles, Stavros

Published in:
Advances on P2P, Parallel, Grid, Cloud and Internet Computing

DOI:
10.1007/978-3-030-02607-3_6

Publication date:
2019

Blockchain-based Trust Communities for Decentralized M2M Application Services

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. Trust evaluation in decentralized M2M communities, where several end-users provide or consume independently M2M application services, enable the identification of trustless nodes and increase the security level of the community. Several trust management systems using different trust evaluation techniques are presented in the application field of M2M. However, most of them do not provide a secure way to store the computed trust values in the community. Moreover, the trust agents participating in the trust evaluation process are not securely identified and could lead to misbehavior among the trust agents resulting in non-reliable trust values. This research identifies several problems regarding decentralized M2M application services and the trust evaluation process. In order to overcome these issues this research proposes a novel approach by integrating blockchain technology in trust evaluation processes. Moreover, this publication presents a concept for using blockchain within the system for decentralized M2M application service provision. Finally, the combination of P2P overlay and blockchain network is introduced in order to verify the integrity of data.

Keywords: Blockchain · M2M · Security · Service and Application · Trust

1 Introduction

The authors in [1] introduce a decentralized approach for Machine-to-Machine (M2M) application service provision where every end-user has the possibility to provide easily M2M application services. All participating peers (service providers/service consumers) are using a Peer-to-Peer (P2P) network for communication and information storage. In contrast to traditional service platforms the approach presented in [1] has several advantages such as avoiding single point of failures, enabling resource flexibility and platform independency. However, one disadvantage is that there is no centralized entity controlling the service creation process made by the end-user (peer). Also, there is no authority that ensures that the created services meet the conditions for being deployed in the community. Moreover, the decentralized character of the approach could lead to

several security issues performed by attackers [2]. Therefore, trust relationships between the peers and the services are necessary to mitigate possible security attacks.

To overcome several issues in decentralized M2M application service environments the authors in [2-4] propose a framework for functional verification and trust evaluation. The trust evaluation is realized using a decentralized test architecture and through the combination of several model-based testing techniques. Trust evaluation is done by end-users acting as Test Agents and cooperating with each other in order to compute an overall trust-level of M2M application services. The trust results are also used for trust-based selection and composition of M2M application services. The trust computation generates a significant amount of trust data among end-users and one aim of this research is to make these data tamper-proof.

This research paper aims to identify unsolved security issues for decentralized M2M application services. Moreover, it optimizes the decentralized service creation process and improves the trust evaluation of M2M application services using blockchain technology. This paper is structured as follows: Section 2 will summarize the decentralized approach for M2M application service provision. Section 3 will give an overview about the framework for functional verification and trust evaluation. Section 2 and 3 will also highlight some existing issues considered for optimization. The integration of blockchain technology for trust evaluation and data storage is introduced in Section 4. Finally, section 5 will introduce the combination of P2P overlay and blockchain within the M2M community.

2 Decentralized M2M Application Services

2.1 Autonomous Decentralized M2M Application Service Provision

A completely decentralized M2M system architecture was presented in [1] where the M2M service platform itself is not provided by a platform operator but by end-users of the platform itself. End-users are able to design individual M2M application services and make them available for other end-users or central service providers. End-users have also the possibility to cooperate with each other in order to provide complex M2M application services.

The architecture framework for decentralized M2M application service provision presented in [5] includes a Service Management Framework (SMF) which consists of a local Service Creation Environment (SCE) and a Service Delivery Platform (SDP). Moreover, the SMF includes all available devices and services present in the personal environment of the end-user and integrates also remote services which are provided by other end-users. The Service Creation Environment (SCE) provides a Graphical User Interface (GUI) for designing graphically the behavior of a M2M application service. This GUI enables the end-user to combine building blocks representing the M2M service components, M2M devices and multimedia service components (Fig. 1). M2M application services are described by machine-readable State Chart XML (SCXML). In order to be consumable for other entities the application requires an application interface (described by an Interface Description) with which it forms an application service.

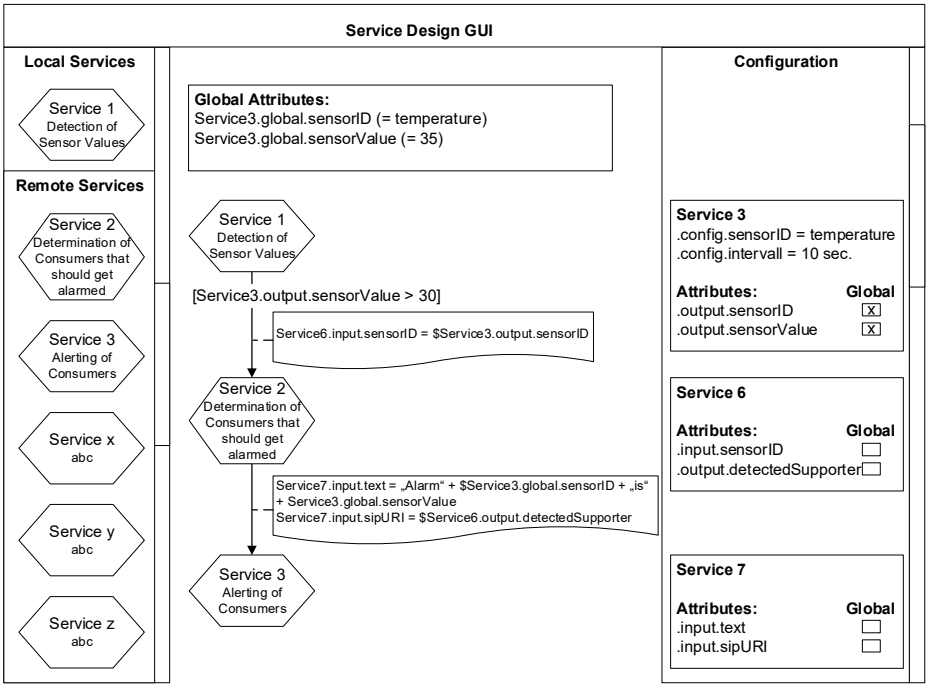

Fig. 1. GUI to Design M2M Application Services

The authors in [5] introduce an M2M community as a social network that is built between users of the M2M service platform. This community can be used to create interest groups by providing different sub-communities. It can also be used in order to address different application fields or geographical locations. End-users and the application services they provide can be part of several sub-communities at the same time. The M2M community can be organized by using the Interface Description (IFD) of the application services. The parameters inside the IFD can be used to derive the specific sub-communities for the nodes [1].

After the M2M application service is modelled by the service provider/end-user it will be configured automatically and autonomously by connecting the specific instances of services that are involved in the distributed M2M application service and described by the modeled state machines. [1]

2.2 Limitations and Challenges as Basis for Optimization

Although there exist various benefits of the decentralized M2M application provision approach, several limitations and challenges can be derived and are described in the following.

For instance, an end-user may have not so much prior technical knowledge for creating a M2M application service. Wrong services can be linked together or wrong con-

figurations can be made by the end-user and thus would lead to wrong or malfunctioning M2M application service. Another security issue is the way the service provider (end-user) is addressed after service registration and the history about past provided services. The authors in [1] define that a service instance is "an actual implementation of a service by an end-user that can be addressed via a service endpoint". Service endpoint means in this context the point of access or the Uniform Resource Identifier (URI). During lifetime the service provider could change his contact information/URI, moreover the authors in [1] do not consider the possibility to store the history of all services provided by an end-user. It could be that an end-user continuously changes his identities or provides fake or trustless services without being identified. The entry and exit of peers in the M2M community is not defined. A missing or bad authentication mechanism leads to the entry of malicious peers which could harm the system with their misbehaving. A composed M2M application service can consist of several single M2M application services. Same services can be provided by multiple peers. The peers providing the same service are selected randomly in order to be part of a composed M2M application service. Randomly selecting is not secure and could lead in selecting unsecure or untrusted peers. A trust selecting principle could be defined in order to assign peers to a M2M application service based on the trust level.

In order to deal with several trust related issues, a framework for functional verification and trust evaluation is introduced in [2] and is going to be explained in the next section.

3 Functional Verification and Trust Evaluation for Decentralized M2M Application Services

For testing the functionality of new M2M application services, the authors in [3] propose an approach by introducing a test architecture consisting of a Test Master, Test Agents, and a Test Generation Environment (TGE). The Test Master coordinates the overall testing framework by sending and exchanging information with the TGE and the Test Agents. Furthermore, the Test Master gets test instructions from the TGE and forwards them to the Test Agents for test execution on the System under Test (SUT) which in this research are M2M application services. The obtained results of the test execution from the Test Agents are then evaluated by the Test Master. The TGE collects information about the M2M application services and derives based on that information suitable test cases which are then sent as test instructions to the Test Master. For evaluating the initial trust level of new M2M application services the authors in [2] propose to integrate the trust evaluation process within the functional testing process by using the test architecture and the outcomes of the test execution for evaluating the trust level. However, the approach presented in [2] and [3] contains centralized elements such as the Test Master, which represents a drawback regarding single point of failure or centralized management about the test and trust reports. To overcome centralized entities, the authors in [4] propose an optimization of the overall framework by distributing the role of the Test Master among other peers part of the M2M community, which will autonomously do the test execution and the evaluation of the obtained test results. First

of all, the service provider designs a M2M application service logic using the GUI (see Fig. 1) which is part of the SCE as described in [5]. Therefore the end-user graphically (see Fig. 2) creates a state machine that represents the behavior of the system. The SCE generates from this logic a formal Service Description and deploys the M2M application service to other users by providing the Service Interface Description of the M2M application service. The Service Description also containing the Service Interface Description is sent to and stored in the P2P network. [1].

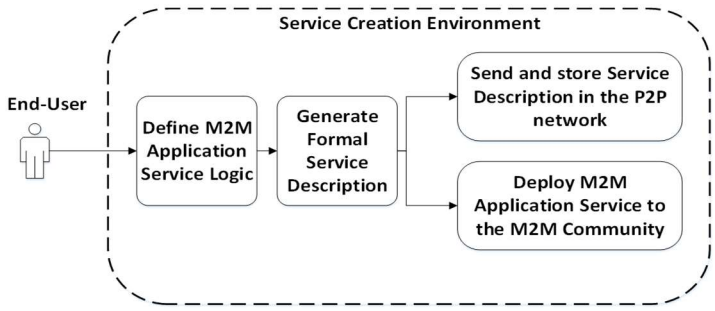

Fig. 2. Service Creation Environment

After the deployment of a new M2M application service all peers will receive a notification and are also able to pick up the Service Description and the Service Interface Description of the new M2M application service from the P2P network. This information is used from the TGE (see Fig. 3) to create a Test Application Description (TAD) [3]. The TAD is used to generate a behavior model from which test cases for functional verification and trust evaluation of new M2M application services are derived. The test instructions together with the test cases are sent to the Test Agents for test execution.

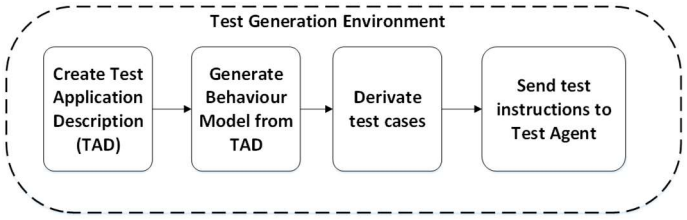

Fig. 3. Test Generation Environment

In order to assess efficiently the trust level for M2M application services, it is proposed to combine the results of functional and performance testing in [3]. Furthermore, the test execution (see Fig. 4) is done independently by one ore many peers/end-users acting as Test Agents. The obtained tests results are evaluated and an initial trust level [4] for the new M2M application service is assigned and stored among all other peers in the P2P network. A service could be evaluated by many end-users independently and the different test results obtained by the end-users can be combined to calculate an

overall verdict about the new M2M application service. The calculation of the verdict also considers the trust level of the different end-users performing the tests. The total trust level of an end-user consists of the trust levels the M2M application services it provides. End-users with better trust levels are more weighted in the calculation process than end-users with low trust levels. Thus, this approach enables a distributed and efficient way to verify new M2M application services. Moreover, the computed trust values of the M2M application services and end-users are considered for the service selection and composition process.

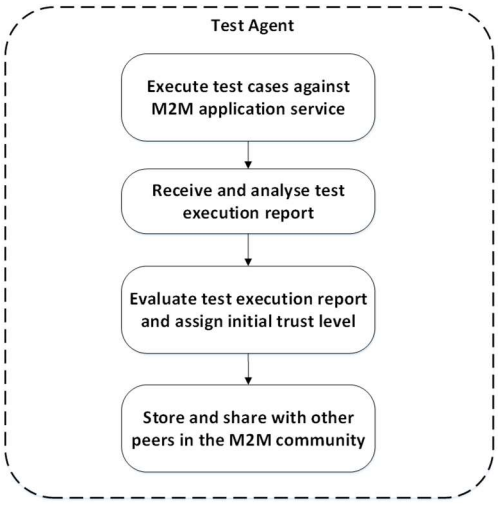

Fig. 4. Test Agent Activities

The random selection [1] of instances providing one of the M2M application services part of the created service chain is not secure and could lead to the selection of M2M application services provided by unsecure or trustless peers. This could result to an unstable and not efficient composed M2M application service. Therefore, the authors in [4] propose to consider the trust level of M2M application services and end-users for the application service selection and composition process and introduced an algorithm for selecting trustworthy M2M application services (Fig. 5).

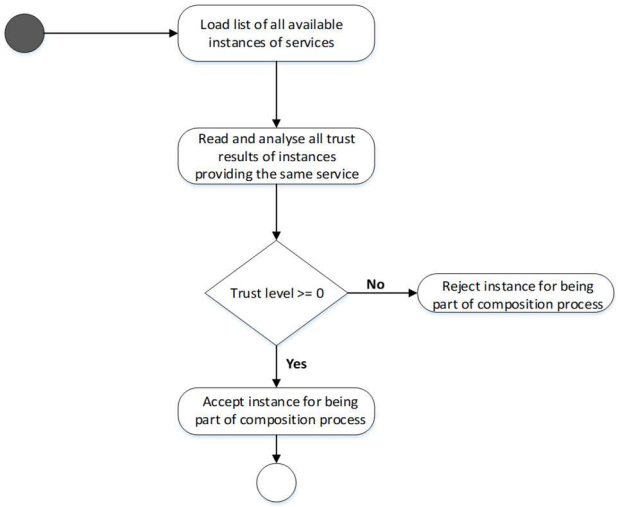

Fig. 5. Algorithm for Selecting Trustworthy M2M Application Services

Another issue is the way computed trust data are stored in the network. As any end-user can act as a Test Agent, there is a possibility that the user modifies or changes past evaluation results in order to manipulate the current trust level of the observed M2M application service. The authors in [4] have reviewed several existing trust management systems for trust evaluation of new and existing services. Based on that evaluation the existing trust approaches do not provide or consider any solution for a secure data storage system of trust related data.

4 Blockchain Integration for Trust Evaluation in M2M

Nowadays, there is an increasing hype for using blockchain to secure systems in several application fields. Blockchain is a distributed database which records all transactions, agreements, contracts and/ or other digital assets between peers participating in that community. In Bitcoin [6], blockchain is used for storing all transactions in the network. According to [7], trust is established due to the fact that everyone in blockchain has a direct access to a shared "single source of truth". All transactions, which are public, comprise specific information, such as date, time, and number of participants. Every peer in the network has a copy of the blockchain and the transactions are validated by the so-called miners using cryptographic principles. These enable nodes to automatically recognize the current state of the ledger and every transaction in it [7]. As stated in [7], a corrupted transaction will be immediately refused by the nodes since they do not reach a consensus for validating that transaction. The authors in [8] state that "once a new block is formed, any changes to a previous block would result in different hash-code and would thus be immediately visible to all participants in the blockchain".

The main benefits of using blockchain are ensuring data integrity and non-repudiation. Moreover, blockchain also provides secure access and identity management possibilities. Regarding the integration of blockchain in the application field of M2M/Internet of Things (IoT) the literature review provides several publications [9-12] dealing with secure data storage and data integrity in relation with blockchain. However, none of the publications [9-12] consider the blockchain technology for using in connection with trust management systems and the computed trust values.

Managing trust values in a decentralized M2M community is challenging because of the increasing number of nodes joining and leaving the network and of their possible malicious behavior by removing or changing data which harm the system. As mentioned in [4], the behavior of an M2M application service can be evaluated by one or more end-users acting as Test Agents. The evaluation process includes functional and performance tests executed against the M2M application service. The combination of the verdicts obtained from the tests done by the Test Agents are used to calculate the trust level of the observed M2M application service. Based on this trust level, other end-users are able to check how much they can trust an M2M application service. As mentioned in the previous sections one issue is that the evaluated trust data can be manipulated or removed from the network. In order to optimize the storage system of the trust management system and to ensure tamper-proof trust data this research propose to store all the evaluated trust data in the blockchain.

After a Test Agent has performed the trust evaluation steps and has computed the trust level (range from -1 to 1) of an M2M application service, it will send a blockchain transaction to the end-user providing the evaluated M2M application service. The blockchain transaction consists of the trust level, Service ID, Service Instance (contact information about the service provider) and the Test Agent Username. This transaction will be broadcasted to all nodes part of the blockchain network and is going to be part of a block. Next, this block has to be validated from other nodes in order to achieve a consensus for an identical version of the blockchain. An overview of the integration of blockchain for storing trust data is shown in Fig. 6.

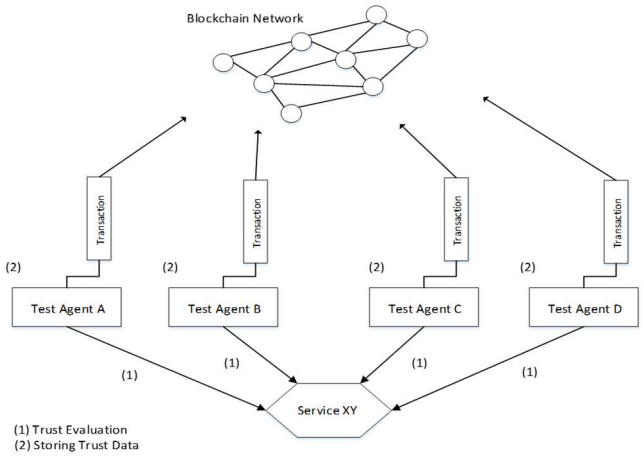

Fig. 6. Integration of Blockchain for Storing Trust Data

To achieve a consensus for validating a transaction and creating a block the literature provides several consensus mechanisms, which are summarized in [13]. The first consensus mechanism introduced in blockchain is the Proof of Work (PoW) [6]. The PoW consists of nodes acting as "miners" by trying to solve a computationally puzzle called hash. The node who solves first this puzzle will validate and add a new block of transactions to the blockchain. The performing activities are rewarded for the first successful miner for validation transactions and creating new blocks [6]. However, performing PoW requires hardware with high computational power and the mining process is an energy-intensive process. Another consensus mechanism is Proof of Stake (PoS) which tries to offer a more efficient way to validate transactions in the blockchain. This mechanism does not require high computing power and selects randomly nodes for mining based on several criteria (depends on the PoS version) [14]. In order to benefit from this energy saving character, this research proposes to also consider the PoS consensus mechanism for validating new transactions and creating blocks within the presented blockchain approach.

The data which is going to be included in a blockchain transaction after trust evaluation by the test agents consist of information about the trust level, observed service and service provider and contact information about the test agent (Fig. 7).

Data			
Trust Level	**Service ID**	**Service Instance**	**Test Agent**
(-1, 1)	Service XY	Peer YZ + contact info	Peer Z + contact info

Fig. 7. Structure of the Data Part of a Blockchain Transaction

5 Combining Blockchain Network and P2P Overlay for More Trust

To increase the reliability in the M2M community this section introduces the combination of the P2P overlay used in [1] and the blockchain network for trust data storage and for trust verification. Moreover, this section proposes to use this combination for several aspects such as end-user and service registration, and Interface Description verification for triggering the trust evaluation process. Figure 8 shows an overview about the different aspects the combination of P2P overlay and blockchain network is proposed to use for in decentralized M2M application services and their trust computation.

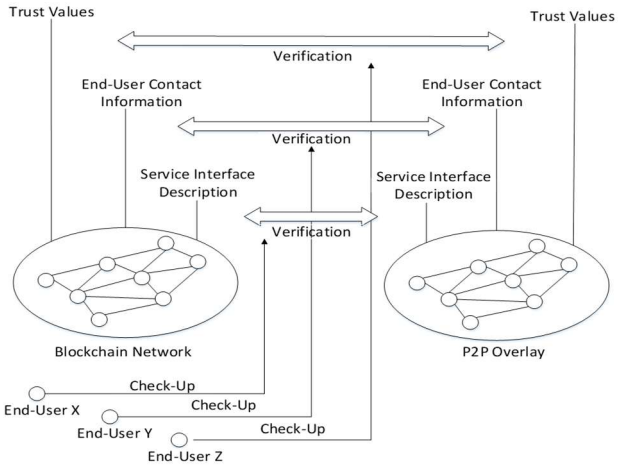

Fig. 8. Overview about the Usage of P2P and Blockchain Combination

5.1 Trust Value Verification for End-Users

The data stored in the blockchain is tamper-proof and is used for integrity check-ups. However, to enable efficiency the P2P overlay introduced in [1] is also proposed to use for storing the evaluated trust data of the M2M application services. The P2P overlay will only store the current status of the trust level and every end-user part of the M2M community has write permission for it. For simple and quick look-up about a trust value the end-user is able to do it firstly in the P2P overlay. Moreover, the end-user has the possibility to check the trust history about a M2M application service in the blockchain for past trust data entries in warily situations. For integrity check-ups the end-user requires the data from the P2P overlay and compares/verifies it with the data from the blockchain. If the compared values match, the end-user can rely on these entries and can continue selecting or not selecting the M2M application service.

5.2 Storage of End-User and Service Information

The P2P overlay introduced in [1] is used for the management of M2M application services. The end-user providing a service will register the service using the P2P overlay network by storing the Interface Description (IDS) of the service and its associated personal temporary contact information. The storage in the P2P overlay could lead to several security attacks as evaluated in [2]. Besides, two main issues can appear. First, after storing the first version of an IDS in the P2P overlay, the service provider could change the IDS and then overwrite the first version without any notification. Thus, other end-users would not have the opportunity to check if there are many versions of the IDS, or what kind of changes happen. This information obtained by the end-user could help for a better overview about the behavior of the service provider. Another problem arises for Test Agents which are continuously performing tests for functional verification and trust evaluation of M2M application services in the community. The Test

Agents will test M2M application services when they are first deployed. However, they do not know when and after which time intervals to test an existing M2M application service. Another issue is that end-users providing a service are only identified by their temporary contact information which can be changed during their lifetime. An attacking peer is able to register several times with different identities and providing harmful services trying to break down the M2M community.

To solve the above mentioned problems, this section proposes to also use blockchain for storing the information about service providers and the services such as the Interface Descriptions. Due to the fact that data is overwritten and changed in the P2P overlay, Test Agents would have the possibility to identify this change in an Interface Description in order to perform functional verification and trust evaluation of M2M application services. Moreover, other end-users would also have the possibility to verify the information about a service or service provider by comparing the information in the P2P overlay and the blockchain network. The Interface Description of M2M application services is retrieved in the P2P overlay at regular intervals. This description is compared to the description stored in the blockchain. If this information stays permanently stable, no further action must be taken from other nodes. If there are changes, the M2M community respectively the end-users acting as Test Agents will start testing the M2M application service in order to verify its functionality. This verification could increase or decrease the trust value of the M2M application service.

The blockchain within the M2M community can also be used in order to check which end-user has provided what services and when in the past. In order to mitigate the problem with end-users and their different temporary contact information and the many services they can provide, this research proposes to define a permanent username for every end-user and to store these information both in P2P and blockchain for further integrity analysis and to comprehend what services they offer.

6 Conclusion

In this publication, the unsecure storage problem of trust management systems in M2M is highlighted and optimized by the integration of blockchain technology. Therefore, blockchain is used for different aspects of the decentralized M2M application service provision and trust evaluation process. First, blockchain is used to securely store all the computed trust data and to make them tamper-proof. Second, the coordination and cooperation among the trust agents (Test Agents) during the trust evaluation is done using blockchain. Besides, the M2M application service provision process is improved by registering the service provider in the blockchain and enabling an efficient identity management which mitigates fault-providing M2M application services by malicious service providers. Moreover, Interface Descriptions of M2M application services are stored in the blockchain in order to include all possible versions of them. Finally, the combination of P2P overlay and blockchain is introduced for verification and comparison of data such as trust values, end-user contact information, and service information. This increases the overall reliability of the decentralized M2M application service environment and the trust evaluation process.

Acknowledgment. The research project P2P4M2M providing the basis for this publication is partially funded by the Federal Ministry of Education and Research (BMBF) of the Federal Republic of Germany under grant number 03FH022IX5. The authors of this publication are in charge of its content.

References

1. Steinheimer, M., Trick, U., Fuhrmann, W. and Ghita, B.: Autonomous decentralised M2M Application Service Provision. Seventh International Conference on Internet Technologies & Applications (ITA 17), IEEE, Wrexham, UK (2017).
2. Shala, B., Wacht, P., Trick, U., Lehmann, A., Ghita, B. and Shiaeles, S.: Trust Integration for Security Optimisation in P2P-based M2M Applications. International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 17), IEEE, Sydney, Australia (2017).
3. Shala, B., Wacht, P., Trick, U., Lehmann, A., Ghita, B. and Shiaeles, S.: Framework for Automated Functional Testing of P2P-based M2M applications. 9th International Conference on Ubiquitous and Future Networks (ICUFN 2017), IEEE, Milan, Italy (2017).
4. Shala, B., Wacht, P., Trick, U., Lehmann, A., Ghita, B. and Shiaeles, S.: Trust-based Composition of M2M Application Services. 10th International Conference on Ubiquitous and Future Networks (ICUFN 2018), IEEE, Prague, Czech Republic, (2018).
5. Steinheimer, M., Trick, U., Fuhrmann, W. and Ghita, B.: P2P-based community concept for M2M Applications. Proc. of Second International Conference on Future Generation Communication Technologies (FGCT 2013), London, UK (2013).
6. Nakamota, S.: Bitcoin: A Peer-to-Peer Electronic Cash System. (2008) available from: https://bitcoin.org/bitcoin.pdf (Accessed 12 January 2017).
7. Hashemi, S., Faghri, F. and Rausch, P.: World of Empowered IoT Users. International Conference on Internet-of-Things Design and Implementation, Berlin, Germany (2016).
8. Samaniego, M., Deters, R.: Using Blockchain to push Software-Defined IoT Components onto Edge Hosts. International Conference on Big Data and Advanced Wireless technologies (BDAW´16), Blagoevgrad, Bulgaria (2016).
9. Huang, Z., Su, X., Shi, C., Zhang, Y. and Xie, L.: A Decentralized Solution for IoT Data Trusted Exchange Based-on Blockchain. 3rd IEEE International Conference on Computer and Communications, Chengdu, China, (2017).
10. Jung, M. Y., Jang, J. W.: Data Management and Searching System and Method to Provide Increased Security for IoT Platform. International Conference on Information and Communication Technology (ICTC), IEEE, Jeju, South Korea, (2017).
11. Liang, X., Zhao, J., Shetty, S. and Li, D.: Towards Data Assurance and Resilience in IoT Using Blockchain. Military Communications Conference (Milcom 2017), IEEE, Baltimore, MD, USA, (2017).
12. Liu, B., Yu, X. L., Chen, S., Xu, X. and Zhu, L.: Blockchain based Data Integrity Service Framework for IoT data. 24th International Conference on Web Services, IEEE, Honolulu, HI, USA, (2017).
13. Chalaemwongwan, N. and Kurutach, W.: State of the Art and Challenges Facing Consensus Protocols on Blockchain. International Conference on Information Networking (ICOIN), IEEE, Chiang Mai, Thailand, (2018).
14. King, S., Nadal, S.: Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. (2012) available from: https://peercoin.net/assets/paper/peercoin-paper.pdf (Accessed 12 April 2018).