Distributed Ledger Technology for Trust Management Optimisation in M2M

Besfort Shala[1,2], Ulrich Trick[1], Armin Lehmann[1], Bogdan Ghita[2], Stavros Shiaeles[2]
[1]Research Group for Telecommunication Networks, Frankfurt University of Applied Sciences, Frankfurt am Main, Deutschland, shala@e-technik.org
[2]Centre for Security, Communications and Network Research, University of Plymouth, Plymouth, UK

Abstract

Decentralised environments with a high number of end-users and M2M devices used in several M2M services increase the importance of a secure trust management and comprehensive trust evaluation system to avoid frauds from malicious nodes. Blockchain, as part of the distributed ledger technology (DLT), helps to improve the overall security in a decentralised M2M community in various aspects. This research publication summarises several existing trust management and evaluation approaches, concluding with their benefits and limitations. Besides these, it highlights the advantages of the distributed ledger technology with the focus on blockchain and consensus mechanisms. In this context, most relevant consensus mechanisms are reviewed and an optimisation concept is proposed. Finally, a blockchain-based trust evaluation system is presented which can be used for trust evaluation and computation of decentralised M2M services part of an M2M community.

1 Introduction

Intelligent end-user devices have a great potential to be accessible for external entities as a service for different applications or processes. In order to enable every end-user without technical knowledge to act as a service provider, the authors in [1] propose a fully decentralised Machine-to-Machine (M2M) system architecture where M2M services can easily be designed and deployed for the personal environment of a peer or to other peers part of a so called M2M community. The decentralised character of service provisioning disables single point of failures and increases the efficiency by sharing resources among the participants. However, an underestimated problem in decentralised networks is that peers potentially behave in a malicious or malfunctioning manner which renders their controllability difficult. A good measure to prevent these risks in the M2M community is building trust relationships between the peers. Therefore, the trust score of peers should be evaluated and maintained by a trust management system. [1]

The literature review in [2] provides a wide range in several application fields regarding trust management and evaluation. However, most of them focus only on direct and indirect observations with special consideration of service satisfaction between the peers but fail to consider several other elements of a peer for trust evaluation. Moreover, most of the existing approaches do not provide a tamper-proof storage system for trust related information among the peers. Another issue of trust management system is that most of them consider new peers and services as trusty without taking into consideration that a malicious or malfunctioning peer could enter the community and provide untrustworthy services.

Distributed ledger technologies are assigned with favourable features, such as non-reversable/modifiable data entries, privacy and security capabilities, automated data synchronization, decentralisation, and transparency [3]. Thus, they provide a great potential to optimise the overall security of a system, specifically trust evaluation and management systems for decentralised M2M services and peers.

This paper covers the topics of trust management and distributed ledger technology and is structured as follows. Section II presents a review of several existing trust approaches in several domains. Section III introduces the distributed ledger technology with special focus on the way how consensus is achieved in the network. A novel trust evaluation system is presented in Section IV. This system integrates trust aspects of a peer and advantages from distributed ledger technology. Moreover, other blockchain-based approaches are reviewed and a blockchain optimisation concept regarding the consensus is presented.

2 Existing Trust Management and Evaluation Approaches

There are a lot of existing trust approaches regarding trust evaluation and management in the literature. This section summarises and evaluates the most relevant ones introduced in M2M or Internet of Things (IoT) and in other related fields such as P2P systems [4-12].

A trust management approach to “support service composition applications in SOA-based IoT systems” using direct trust and indirect trust for evaluation is proposed in [4]. The trust evaluation is made by every user part of the network where direct trust is established through direct interaction based on non-functional characteristics (response time, failure probability, prices, etc.) between nodes whereas indirect trust through recommendations from users with social similarities.

A distributed trust management system where the trust evaluation process is done by the peers themselves is presented in [5]. It separates the nodes in the network in “alpha nodes" and normal nodes. The "alpha nodes" are considered with higher resource capabilities and therefore selected to maintain the whole trust evaluation processes. For trust computation the trust score between nodes are calculated using the rating entries from the nodes and their weights. For new nodes without a rating history an average rating is set.

The trust management system in [6] deals with an IoT environment consisting of community managers who have a supervision role in the network and the other normal nodes consisting of service requester and service provider nodes. All the normal nodes are considered without an initial trust score. The community manager is a trustful entity who coordinates the collaborations between the nodes and also acts as a trust manager by evaluating the trust score of others using recommendations and their credibility.

Considering the different characteristics of IoT, the authors in [7] propose a centralized trust management system with different trust management servers responsible for their specific geographical locations to evaluate the trust score using the history from past behaviours. The proposed trust management system assumes that all peers are trustworthy from the beginning. Recommendation and the quality of recommendation are defined as trust metric parameters and used for evaluating the trust score.

The authors in [8] introduce a trust model for assessing the trust score of new IoT devices before interactions with others occur. In order to determine the initial trust score of a new device, the authors in [8] use the challenge-response mechanism which evaluates initially the uncertainty level of a device based on its behaviour and stores these results within the tested device. This step is done by a centralized controller of the personal space IoT system which performs different challenges on the new device in order to evaluate its behaviour.

A trust management system where peers are divided based on their interest and "similarity of communication history" in several clusters is proposed in [9]. The communication of peers which are part of the same cluster as well as the communication between peers of different clusters will be part of the trust evaluation process. The nodes within a cluster are managed by a cluster node. For trust evaluation the following parameters are considered: history of communication; number of successful/unsuccessful trading; success/failure rate of communication; feedback.

The authors in [10] propose a so-called "trust bootstrapping" process where services without any trust score are going to be rated. The initial trust score will ensure interaction between the new services in the community and will also be used for further trust evaluation processes. The technique for trust bootstrapping in [10] considers the subjectivity of trust where different nodes have a different opinion about the trust score of the observed service. Moreover, it includes also the different trust score of a service in different situations. Two categories of trust metrics, the Service Trust Metric, which contains the Objective and Subjective Trust Metric, and the Provider Trust Metric are defined in [10] in order to realise bootstrapping.

Another approach is presented in [11] which proposes trust bootstrapping for web services. According to possible characteristics of new web services, they employ three generic mechanisms presented in the following: the inheritance mechanisms where the web service gets the trust score from the service provider, the referral mechanisms where web service gets the trust score based on the referrals from other communities, and the guarantee mechanisms where the web service gets a temporary trust score under guarantee conditions.

To fix several trust-related problems for IoT devices, the authors in [12] introduce a cloud-based smart service community, which can be used by users to register services, to report service satisfaction and recommendation, and recommender credibility. The smart service community is accessible "via a mobile application installed in user-owned IoT devices". The cloud utility is a central storage where service ratings and trust scores are managed. Moreover, service providers have to register and advertise their service to the cloud utility, by also attaching performance data for service quality performance metrics about the service. Service requesters will also register to the cloud utility for using services and reporting measurement reports of certain performance metrics related to a service.

An optimal trust management and evaluation system for decentralised M2M services and peers should fulfil the following characteristics. There should be no single entity or master peer which maintains a part of the whole trust evaluation and management process. This avoids single point of failures or monopoly of peers within the M2M community. Only the trust projects presented in [4, 10, 11] support a fully decentralised architecture. Other approaches [5-9, 12] use a centralised entity or many super nodes for the trust computation. The trust evaluation of a new peer or service enables other peers to figure out more quickly the trustworthiness and to decide to use the service or not. Most of the reviewed approaches [4, 6, 7, 9, 12] do not consider the initial trust score of a new peer or service. This opens the door for malicious peers to enter the community and to harm the network. The approaches presented in [5, 8, 10, 11] support initial trust scores computation but lack evaluation techniques and parameters. Besides the initial trust score, the ongoing trust score of an existing service or peer should be evaluated continuously to provide updated trust information to the community (is not considered in [8, 10, 11]). Most of the trust approaches also do not provide or consider any solution for a secure data storage system of trust related data. The authors in [4] try to solve the storage management problem by considering only nodes with good trust values and with high impact on the community. However, the framework should consider all trust values because bad trust scores of nodes are also very important in order to mitigate bad behaviour in the community as well as trust values from nodes with low impact on the community. Another important characteristic of a trust management system is a comprehensive trust model which covers several trust-related aspects of a peer. The work in [9, 11, 12] provide interesting trust parameter for trust evaluation but do not cover other aspects of a peer, such as the modification of data. However, most of the presented trust metrics in the literature are for a specific application field and do not consider the characteristics of decentralised M2M services.

Table 1 summarises the strengths and weaknesses of the trust approaches against several requirements mentioned in [2].

Table 1: Evaluation of Trust Approaches

Trust Management Approaches	Decentralised Evaluation	Initial Trust Score	Ongoing Trust Score	Tamper-proof Storage	Comprehensive Trust Model
[4]	+	-	+	o	-
[5]	o	o	+	-	-
[6]	o	-	+	-	-
[7]	-	-	+	-	-
[8]	-	o	-	-	-
[9]	o	-	+	-	o
[10]	+	o	-	-	-
[11]	+	o	-	-	o
[12]	-	-	+	-	o

The following notations are used to assess the satisfaction for the requirements: + satisfied; - not satisfied; o partially satisfied; / not available.

3 Distributed Ledger Technology

3.1 Basics

The distributed ledger technology provides a secure and decentralised database that can be shared across a network of multiple sites, geographies or institutions where all participants can have their own identical copy of the database. The security and accuracy of assets stored in the ledger are maintained cryptographically using “keys” and signatures to control what can be done by whom within the shared ledger. Thus, any changes to the ledger are reflected in all copies in minutes, or in some cases, seconds. [13]

According to [14], distributed ledger approaches rely on two architectures: blockchain, where transactions are stored in a block and all blocks are linked cryptographically with each other rendering the data tamper-proof; Directed Acyclic Graph (DAG), where nodes in a graph are representing transactions and the edges of them indicate the direction of confirmation between the transactions.

The authors in [15] state that blockchain architectures provide more decentralization, transparency, and immutability in comparison to DAG architectures. The benefits of blockchain are used in several application fields and one of the most famous representatives to use it for storing all transactions in the network is Bitcoin [17], where every peer in the network has a copy of the blockchain and the transactions are validated by the so-called miners using cryptographic principles.

The consensus plays a crucial role in blockchains, as there is no centralised entity to coordinate the records made by the participants. Consensus mechanisms are used to agree for the same copy of the ledger between the nodes. The author in [16] defines consensus as “a set of steps that are taken by all, or most, nodes in order to agree on a proposed state or value”. Thus, to achieve a consensus for validating a transaction and creating a block the literature provides several consensus mechanisms [17-22].

3.2 Related Consensus Mechanisms

To benefit from the blockchain and distributed ledger technology, it is required to ensure that a reliable and secure consensus mechanism is selected. Therefore, in the following the most relevant consensus mechanisms are reviewed.

The first mechanism for achieving consensus on the same ledger between nodes in a blockchain is Proof of Work (PoW). PoW is introduced in Bitcoin and consists of nodes acting as “miners” by trying to solve a computational puzzle which requires to iterate through a so-called nonce value until the target hash is reached. 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. [17]

A more efficient way to validate transactions in the blockchain is provided by the Proof of Stake (PoS) consensus protocol. This mechanism does not require high computing power and randomly selects nodes for mining based on several criteria (depends on the PoS version). The first version of PoS defines proof of ownership of a currency and the coin age consumed by a transaction as criteria to select nodes for being able to add new blocks. In case of different concurrent chains in the network the blockchain with the highest score is selected as the main chain. This score is computed based on the consumed coin age of every transaction which is part of the block. [18]

The block producers are elected by other nodes in the Delegated Proof-of-Stake (DPoS) consensus mechanism (is one of the extended versions of PoS). The election votes are weighted based on the network stake of each voter. In DPoS the number of block producers is fixed, and every block producer is allowed to produce one block per round. If a block producer does not perform the right actions, he can be voted out by the community. The DPoS provides a pretence democracy by providing voting. [19]

Nano is a consensus approach which uses a block-lattice structure where each block contains only one transaction and every node (account holder) has its own blockchain (not the whole but only a single view about it). A sending transaction from the sender of the funds and a receiving transaction from the receiver are required to send funds from one account to another. If there is a conflict on a transaction, the system can start a voting mechanism where representatives chosen on behalf of account holders vote for transactions. Their voting power is calculated based on the sum of all balances of account that have chosen them. In order to mitigate double-spending attack, Nano uses a light-weight version of the PoW. [20]

In the Ripple consensus approach each node has a Unique Node List (UNL) and a ledger. The UNL contains a list of nodes which are chosen by a node with the assumption that they will not behave maliciously. The consensus works in that way that a node votes by comparing the transaction received by the UNL with other transactions from previous rounds or other nodes if they are matching. Transactions that receive a negative vote will be considered for the next round of the consensus or they are going to be discarded from the network. The voting mechanism runs until the transaction receives 80% of the votes which qualifies them to be included in the ledger. The consensus in Ripple is done by a fixed number of peers. [21]

IOTA uses Tangle as an underlying distributed ledger technology and consensus approach. Tangle is based on a directed acyclic graph (DAG) where all transactions are linked with each other. Unconfirmed transactions are called tips and confirmed transactions sites. In order to deploy a new transaction to the community, a node first needs to validate two previous transactions by performing a light-weight PoW (used to avoid double-spending) and other validating steps (for instance, if the transaction is in conflict with the history of the tangle). After validating two other transactions, the current transaction is linked to them and waits until it is validated by others. Additionally, IOTA adds weight to the sites (transactions). The weight of each site is calculated based on the time a node spent to do PoW for confirming that transaction. Each transaction also has a cumulative weight which is the sum of its own weight plus the sum of own weights of all transactions that approve this transaction. To protect the system from double-spending IOTA uses a coordinator to confirm transactions using so-called milestones. [22]

Existing consensus mechanisms have several limitations which should be considered for the integration in trust management systems. One problem, which is mainly present in PoW, is the computational effort to create and verify blocks in the blockchain. Performing PoW requires hardware with high computational power and the mining process is an energy-intensive one. However, other approaches [18, 19, 20, 22] try to minimize this effort by using a light version of PoW and provide more energy efficiency. The approach presented in [21] removes the need to solve a computational puzzle and relies on transaction similarity checks using a voting system. Another limitation of the reviewed consensus mechanisms is the way how a block creator is selected. Most of the approaches require the mentioned computational effort by selecting the challenge winner as a block creator. Other approaches use the amount of stake to decide for the block creator. However, selecting leaders based on the stake ownership percentage could lead to the problem of centralization and monopolization. Some of the existing consensus mechanisms [18-22] add centralised components to the system in form of representatives or coordinator or fixed number of block creators or consensus nodes. Centralised elements or super nodes maintain a part of the blockchain system could lead to single point of failures and monopoly in the network. Another drawback is that some mechanisms [18-20, 22] do not consider fake transactions which are flooded by malicious nodes in order to harm the system. Others partially solve this problem by requiring high computing power for mining [17] or transaction similarity checks [21].

The consensus mechanism is one of the key components in the blockchain network. Therefore, this research proposes to optimise the consensus process by integrating trust to it. Thus, a “Proof of Trust” consensus mechanism is initially and conceptionally introduced. This consensus mechanism should consider the trustworthiness of nodes throughout blockchain processes including the maintenance of the blockchain, the block creator (miner) selection process, and the block validation and acceptance. Furthermore, for all these steps the nodes are selected based on their trust score. Also, the block validation is done considering the trust score of the validating nodes.

4 M2M Trust Evaluation System

4.1 Blockchain-based Trust Management

A decentralized M2M community bears the risk that several nodes join or leave the network and try to harm the system through their malicious/malfunctioning behaviour. One malicious behaviour could be the modification of data across the network. Specifically, the trust data giving information about the trustworthiness of a node can be modified and lead to wrong trust relationships in community.

To benefit from the tamper-proof feature of blockchain, it is proposed to store all the evaluated trust data in the blockchain [23, 24]. Only a few publications [25, 26] in several application fields have also considered the integration of blockchain in order to improve the trust management system. For instance, the authors in [25] propose to use blockchain for storing public keys of well-behaved nodes in vehicular networks. The workflow of the trust management system starts with vehicles generating ratings of received messages based on their credibility which depends on the distance between the message sender and the event location. These ratings are uploaded on the Roadside units (RSU) which then calculate the offset of trust values for every involved vehicle and integrates these values inside a block. Afterwards using a miner election method (a modified algorithm of PoW and PoS) an RSU node tries to be elected as a miner in order to include a new block to the blockchain. The modified algorithm “takes the sum of absolute values of offsets in the candidate block as the stake” and based on the stake amount a node is selected as a miner. Another approach considering the blockchain technology for trust management is presented in [26]. They introduce a trust management architecture where trust values of service providers are stored in the blockchain. The system architecture proposed in [26] consists of a first level with distributed IoT devices providing services to each other and a second layer with distributed fog nodes also maintain a blockchain which is used by the IoT devices to store trust information in it. The transactions in the blockchain are validated by the fog nodes using the PoS algorithm. The trust model used in [26] to evaluate the trust score of IoT objects considers only honest IoT devices for reporting recommendations (based on the interaction experience) about other IoT service providers to its managing fog. Besides the benefit of securing the data, none of the publications [25, 26] consider the security aspect of the consensus protocol e.g. how a block is created; how it is evaluated by others. Both approaches consider PoS as a consensus protocol, which provides no fair method to be selected as a miner and adds more centralization in a decentralized environment. Another problem is that they consider only the blockchain for storing the data, which leads to higher time consumption for data lookup. Moreover, their trust evaluation systems do not consider the initial trust score of a peer and focus only on recommendations and ratings.

This research proposes, that after a node acting as a so-called test agent has performed all trust evaluation steps and has computed the trust score ranging from 0 to 5 of an M2M service, it will send a blockchain transaction to the end-user providing that service. Moreover, the transaction is going to be broadcasted to all other nodes part of the network for verification. Finally, one of the nodes is going to add this transaction to a block before sending it to the network for consensus achievement. The transaction consists of the trust score, Service ID, Service Instance (contact information about the service provider) and the Test Agent Username. Additionally, this research proposes to combine P2P overlays (such as Chord) with blockchain for enabling data integrity and less time consumption for data lookup. Every end-user in the M2M community could verify the information stored outside the blockchain with the data in the blockchain. [24]

4.2 Architecture of Trust Evaluation System

Figure 1 shows an overview about the architecture of the proposed Trust Evaluation System. This system consists of the Service Trust Evaluation part and the Behaviour Trust Evaluation part. The trust score of a peer is computed based on its services it provides and its behaviour.

Service Testing, Service Monitoring and Service Rating are used in the Service Trust Evaluation part to evaluate the service of a peer. Service Testing enables the computation of the initial trust score of a peer. It includes testing the functionality and the performance of the service after it joins the community. The functional testing verifies the functionality of a service and concludes with the result if the service is behaving like it is mentioned in its system model. In parallel, performance testing is done in order to confirm the participation willingness of the service regarding several requests by service consumers. The Service Testing solves the problem of existing trust evaluation approaches regarding the missing or not correct initial trust score information.

To evaluate the trust score of ongoing services, this research proposes to monitor the behaviour of a service by considering parameters such as the number of online/offline actions or ratio of positive/negative responses. Besides them, it is proposed to consider the rating scores of a service by other users based on their experience in using this service. The results of Service Testing, Service Monitoring, and Service Rating are used to compute the Partial Trust Score of the service through a Service Trust Evaluation Function.

Next to Service Trust Evaluation, the Behaviour Trust Evaluation part aims to identify a malicious or malfunctioning action of a peer regarding a service. One considered aspect is the integrity of service information which also includes trust information, such as the trust score. Therefore, this research introduces the integration of blockchain to benefit from its tamper-proof feature to check the integrity of data which is stored in the blockchain (called on-chain) with the data which is stored outside the chain (called off-chain). As mentioned in the previous subsection, it is considered that after the trustworthiness of a service is evaluated, the trust information is stored in the P2P overlay (such as Chord), and in order to enable every user to check if the trust score has been changed by a malicious peer, the trust information is stored in the blockchain. Thus, the integrity check is used to support end-users in their decision to use a service or not. Therefore, one should use the integrity check feature to reward or punish peers if the data in the off-chain for instance is different with that in the on-chain. This means that the information has been changed by an end-user – concluding with its punishment by decreasing its trust score. Future work will also consider other elements for the Behaviour Trust Evaluation part, which results at the end of the evaluation with a Peer Trust Score. The Peer Trust Score is combined with the Service Trust Score to calculate the Total Trust Score of the Peer.

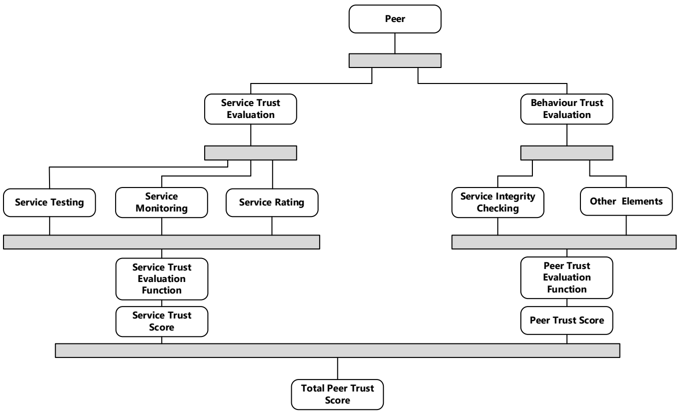

Figure 1: Trust Evaluation System

5 Conclusion

This publication shows that existing trust approaches have several limitations, such as the level of decentralisation, missing trust score information, mutability of data, and incomplete trust model. Therefore, the reviewed trust approaches are not efficient to evaluate the trustworthiness of decentralised M2M services and peers.

For ensuring data immutability, this research proposes the integration of distributed ledger technology. Moreover, it reviews existing blockchain-based trust systems, evaluates consensus mechanisms, and gives some optimisation aspects for a future trust-based consensus.

Additionally, an overall trust evaluation system covering service and peer aspects is presented. Finally, this paper proposes to integrate blockchain for storing trust scores in the blockchain and for checking their integrity against data stored in the off-chain.

6 Acknowledgments

The research project P2P4M2M providing the basis for this publication was 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.

7 References

[1] Steinheimer, M., Trick, U., Fuhrmann, W., Ghita, B.: Autonomous decentralised M2M Application Service Provision. 7th IEEE International Conference on Internet Technologies & Applications (ITA 17), Wrexham, UK, 2017

[2] Shala, B., Trick, U., Lehmann, A., Ghita, B., Shiaeles, S.: Trust-based Composition of M2M Application Services. 10th IEEE International Conference on Ubiquitous and Future Networks (ICUFN 2018), Prague, Czech Republic, 2018

[3] Distributed Ledger Technologies and Financial Inclusion. ITU-T Focus Group Technical Report, 2017

[4] Chen, I., Guo, J., Bao, F.: Trust Management for SOA-Based IoT and Its Application to Service Composition. IEEE Transactions on Services Computing, Vol. 9, No. 3, 2016

[5] Asiri, S., Miri, A.: An IoT Trust and Reputation Model Based on Recommender Systems. 14th IEEE Annual Conference on Privacy, Security and Trust (PST 2016), Auckland, New Zealand, 2017

[6] Benkerrou, H., Heddad, S., Omar, M.: Credit and Honesty-based Trust Assessment for Hierarchical Collaborative IoT Systems. IEEE SETIT 2016, Hammamet, Tunisia, 2017

[7] Saied, Y. B., Oliverau, A., Zeghlache, D., Laurent, M.: Trust management system design for the Internet of Things: A context-aware and multiservice approach. Elsevier Journal, Computer & Security. Vol. 39, Part B, 2013

[8] Nguyen, T., Hoang, D., Nguyen, D., Seneviratne, A.: Initial Trust Establishment for Personal Space IoT Systems. IEEE INFOCOM WKSHPS, Atlanta, USA, 2017

[9] Ma, Y., Wang, D.: A Novel Trust Model for P2P Networks. 12th IEEE International Conference on Natural Computation, Fuzzy systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 2016

[10] Aljazzaf, Z., Capretz, M., Perry, M.: Trust Bootstrapping Services and Service Providers. 9th IEEE Annual International Conference on Privacy, Security and Trust, Montreal, Canada, 2011

[11] Nguyen, H., Yan, J., Zhao, W.: Bootstrapping Trust and Reputation for Web Services. 14th IEEE International Conference on Commerce and Enterprise Computing, Hangzhou, China, 2012

[12] Al-Hamadi, H., Chen, I.-R., Cho, J. H.: Trust Management of Smart Service Communities. IEEE Access. Vol. 7, 2019

[13] Distributed Ledger Technology: beyond blockchain. Government Office for Science, 2016 available from: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledgertechnology.pdf (accessed 19 March 2019).

[14] Burkhardt, D., Werling, M., Lasi, H.: Distributed Ledger. IEEE ICE/ITMC, Stuttgart, Germany, 2018

[15] Blockchain vs. DAG Technology, available from: https://medium.com/nakamo-to/blockchain-vs-dag-technology-1a406e6c6242 (access. 25 March 2019).

[16] Bashir, I.: Mastering Blockchain. Packt, Birmingham, UK, 2017, ISBN 978-1-78712-544-5.

[17] Nakamota, S: Bitcoin: A Peer-to-Peer Electronic Cash System. 2008 available from: https://bitcoin.org/bitcoin.pdf (access. 12 June 2017).

[18] 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).

[19] Snider, M., Samani, K., Jain, T.: Delegated Proof-of-Stake: Features & Tradeoffs. 2018 available from: https://multicoin.capital/wp-content/uploads/2018/03/DPoS_-Features-and-Tradeoffs.pdf (access. 18 March 2019).

[20] LeMahieu, C.: Nano: A Feeless Distributed Cryptocurrency Network, available from: https://nano.org/en/whitepaper (access.18 May 2018).

[21] Schwartz, D., Youngs, N., Britto, A.: The Ripple Protocol Consensus Algorithm, 2014, available from: https://ripple.com/files/ripple_consensus_whitepaper.pdf (accessed 20 March 2019).

[22] Popov, S.: The Tangle, Whitepaper, 2018, available from: https://docs.iota.org/ (access. 20 March 2019).

[23] Shala, B., Wacht, P., Trick, U., Lehmann, A., Ghita, B., Shiaeles, S.: Ensuring Trustworthiness for P2P-based M2M Applications. 7th IEEE International Conference on Internet Technologies & Applications (ITA 17), Wrexham, UK, 2017

[24] Shala, B., Trick, U., Lehmann, A., Ghita, B., Shiaeles, S.: Blockchain-based Trust Communities for Decentralised M2M Application Services. 13th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3GCIPP 2018), Springer, Taichung, Taiwan, 2018

[25] Yang, Z., Yang, K., Lei, L., Zheng, K., Leung, V.: Blockchain-based Decentralized Trust Management in Vehicular Networks. IEEE Internet of Things Journal 2018

[26] Kouicem, D., Bouabdallah, A., Lakhef, H.: An Efficient Architecture for Trust Management in IoE Based Systems of Systems. 13th IEEE Annual Conference on System of Systems Engineering (SoSE), Paris, France, 2018