2020

Trusted, Decentralized and Blockchain-Based M2M Application Service Provision

Shala, Besfort

10.1007/978-3-030-33506-9_19
Advances on Broad-Band Wireless Computing, Communication and Applications
Springer International Publishing

Trusted, Decentralized and Blockchain-based M2M Application Service Provision

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. Decentralized M2M service platforms enable the integration of end-user-based M2M applications and end-user-located M2M resources without the use of central entities or components in the system architecture. Sharing end-user-based M2M applications with other users' part of an M2M community allows the creation of new and complex M2M applications. However, a fully decentralized system often leads to several trust issues regarding the behavior of end-users and M2M applications. A powerful measure to overcome possible limitations of decentralized M2M service platforms and to replace the missing control authority are trust relationships among the nodes. Therefore, this publication proposes a novel concept for trusted M2M application service provision. Moreover, it introduces the integration of blockchain elements and trust evaluation techniques to optimize the M2M application service provision. A trust consensus protocol is integrated in order to secure the decision-making process among the stakeholders which optimizes several aspects, such as peer joining, service registration and application configuration.

Keywords: Trust, P2P, Blockchain, M2M, Security, Service and Application

1 Introduction

Being part of the end-user environment, intelligent M2M devices have a great potential for supporting smart environments when they are used in the creation of new complex M2M applications which are accessible for other users. To realize the integration of such devices, the participation of the end-user in the M2M application service provision process is required. In this context, it is important to provide a fully decentralized architecture for M2M application service provision in order to avoid end-user environment limitations and problems, such as the need for large resources for service development and maintenance, high costs for operating service platforms and the lack of reliability of the platform as result of single point of failures.

The literature review provides a considerable number of different M2M service platforms [1]. However, analyses regarding decentralized architectures and end-user integration have shown that most of them do not completely fulfill these requirements [2]. The authors in [3] introduce an enhanced Dynamic Service Overlay Network (e-DSON) platform with a focus on distributed service provision which supports the provision of user individual services and operates on distributed servers in the Internet. However, the platform is maintained by a centralized operator and does not integrate the end-user in the service composition process. Another approach for M2M application service provision is presented in [2, 4] where every end-user part of the network can provide or consume M2M services and can act as a decentralized M2M service provider.

The authors in [9] highlight the necessity of trust because of the increasing "risks, threats and vulnerabilities at component, device, system, service and human levels" in the world of Information and Communication Technology (ICT). However, existing limitations of end-user-based and decentralized M2M services are the lack of trust in the network and the controllability or access control of joining and leaving peers. Moreover, a trustworthy mechanism, which considers only highly trusted and fully accepted services by all the members in the service composition process, is non-existent.

To overcome trust issues in decentralized M2M communities, the authors in [5, 6, 7] introduce a fully decentralized trust evaluation system which covers the trustworthiness of new and existing entities. Besides those, the presented trust evaluation system covers several aspects of a peer and a service in the trust evaluation process. Additionally, it optimizes the data storage system of the trust results by integrating the blockchain technology.

For using the blockchain technology and its elements in the M2M environment, several integration possibilities are proposed in [7]. However, the consensus protocols used in different blockchain applications have several limitations, such as high computational effort, high energy consumption and lack of trustworthiness. In order to integrate blockchain with its features to M2M service platforms, a fully suitable and trust-based consensus protocol is proposed in [5].

The aim of this publication is to optimize several levels of M2M application service provision starting with the M2M community admission, the M2M service registration and the M2M application configuration. Therefore, blockchain elements and trust provision principles are combined to introduce a novel trusted and blockchain based M2M application service provision system. The remainder of this paper is organized as follows. Section 2 describes the M2M application service provision according to [2, 4]. The decentralized trust evaluation system and the integrated blockchain elements are introduced in section 3. Section 4 presents the optimization approach by integrating a trust-based consensus protocol to the M2M service provision life cycle. Finally, section 5 gives a conclusion of the presented methodologies and approaches.

2 P2P-based M2M Application Provision

The authors in [2] propose an M2M service platform where the end-user is integrated into the application creation process. To ensure independency in the application creation process, central entities, such as central platform or network element provider, are removed from the architecture. To reduce the cost of operating an M2M service platform and to increase the acceptance of an M2M solution, existing resources in the end-user environment are reused. Moreover, the proposed M2M service platform integrates different M2M device technologies and allows them to be combined with each other.

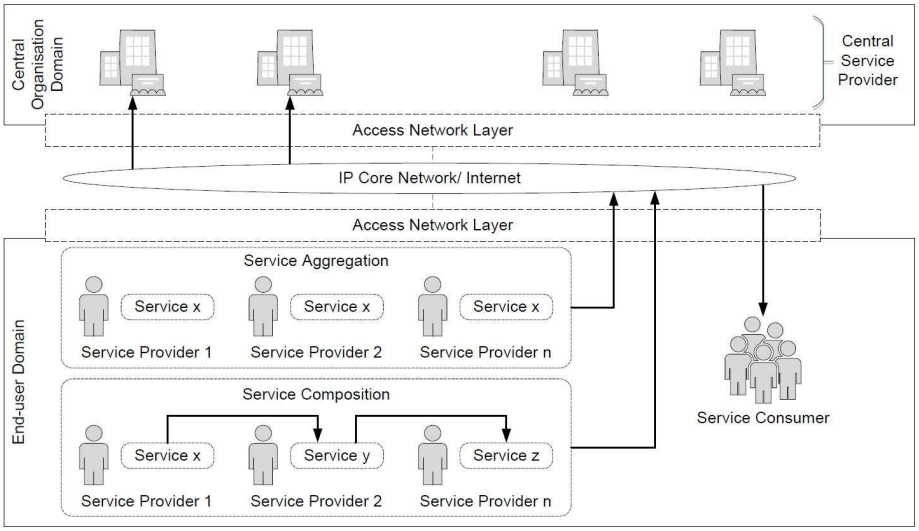

Figure 1: Cooperative M2M Application Service Provision [2]

Using a GUI, even with less technical knowledge, the end-users have the possibility to design individual M2M application services and make them available for other end-users or central service providers. They additionally have the possibility to cooperate with each other in order to provide complex or so called cooperative M2M application services. After this kind of M2M application service is modelled, it will be configured automatically and autonomously by connecting the specific instances of services that are involved in the cooperative M2M application service. Figure 1 shows that a complex service consists of the combination of several distributed services that are networked together. The combination may consider same services with same functionality (service aggregation) or services with different functionality (service composition) [4].

In order to ensure a decentralized system architecture, the author in [2] proposes to use a Peer-to-Peer (P2P) network for communication and information storage between the peers. To create a social network and interest groups among the participating nodes, the authors in [2, 4] introduce an M2M community.

M2M application services are described by machine-readable State Chart XML (SCXML). The application requires an application interface (described by an Interface Description (IDS)) with which it forms an application service in order to be consumable for other entities.

Some additional information regarding the M2M service platform presented in [2, 4] are provided in section 4. Moreover, section 4 lists several limitations and introduces a novel, trust- and block-based optimization approach for M2M application provision.

3 Blockchain-based Trust Model

In order to ensure a secure environment in an M2M community, trust relationships between the participating peers are required. Therefore, the trustworthiness of peers and the services they provide should be evaluated. Several trust definitions depending on the application domain and the context exists. Regarding ICT environments, the authors in [8] state that the preference of an entity for decision-making with other entities and service consumption is affected by trust. Specifically, they claim that "trust evaluation is especially significant in ICT environments where a huge number of entities mutually interact with each other to provide and consume information or resources". The literature provides several trust evaluation and management approaches which are evaluated in [6]. Most of them do not provide a solution for bootstrapping peers or new services which are provided to the community. Moreover, they do not provide a secure mechanism to store the trust scores computed through the trust evaluation process. Therefore, the authors in [5] propose a trust model which optimizes the storage system and includes other trust aspects for evaluation. Moreover, the trust model includes blockchain elements combined with a newly introduced Trust Consensus Protocol. The trust model (from now on called trust evaluation system) and the blockchain are going to be explained in the following.

3.1 Trust Evaluation System

Several security and trust limitations on existing M2M service platforms [1] can be mitigated if there is an overview about the trust scores of the participating peers and services. Therefore, the authors in [5] introduce a completely decentralized and community-based trust evaluation system. An overview of the architecture of the trust evaluation system is shown in Figure 2. For the trust evaluation, several aspects in the M2M community are considered, such as the service functionality, service quality, service acceptance, peers' behavior and participation willingness in several community tasks. To increase the reliability of the data integrity and to support integrity check-ups for data stored in the P2P overlay, the authors in [7] proposes to include blockchain elements in the trust evaluation system. To ensure a decentralized environment without centralized entities, the authors in [5] suggest to distribute all community tasks among the participating nodes.

More specifically, the Trust Evaluation System consists of three main parts, the Service Trust Evaluation, Behavior Trust Evaluation and Task Trust Evaluation. The Service Trust Evaluation includes Service Testing, which covers functional and performance testing and aims to identify the initial trust score of a new service. Moreover, it includes Service Monitoring and Service Rating where the behavior and the performance of a service is monitored by considering several parameters. Besides them, services are rated by other users based on their individual experience on using the services. Service Monitoring and Service Rating are used to evaluate the trust score of an existing service. The results of Service Testing, Service Monitoring and Service Rating are computed using a Service Trust Evaluation Function which concludes with a Partial Trust Score of the Service. Another part, the Behavior Trust Evaluation, is used to check the integrity of a service by comparing the data stored in the blockchain with the data stored in the P2P overlay. The results of the integrity check-up are used to increase (if integrity remains) or decrease (if integrity fails) the trust score of the peer. The third part, the Task Trust Evaluation, evaluates the participation of a peer in community tasks, such as acting as a Test Agent or Blockchain node.

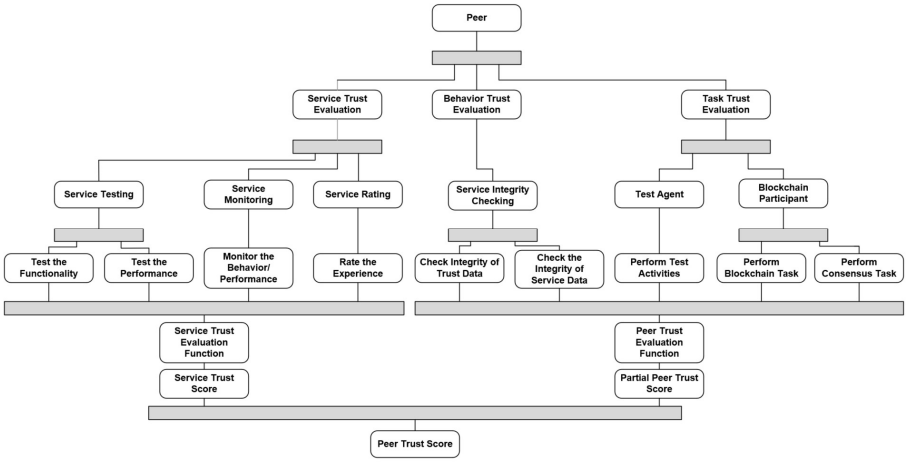

Figure 2: Trust Evaluation System [5]

3.2 Blockchain and Trust Consensus Protocol

The blockchain technology is a subbranch of the so-called distributed ledger which can be defined as an asset database that can be shared across a network of multiple sites, geographies or institutions [9]. The ledger is maintained through cryptographical principles where changes are made available for all network members. One of the key elements of a blockchain is the consensus protocol which is used to agree for the same copy of the ledger among the participating nodes. Specifically, the term consensus protocol is defined in [10] as “a series of procedures from approving a transaction as an official one and mutually confirming said results”. Several publications in scientific libraries and in the industry have proposed different consensus protocols with specific

characteristics where the most relevant ones are reviewed and evaluated in [5]. The review has shown that most of them require computational effort for achieving consensus and validating new transactions. Moreover, they do not provide a fair way to select a node to become the leader who proposes a new block. Another drawback of existing approaches is that they do not consider the trustworthiness of blockchain inputs and nodes who are storing something in the blockchain.

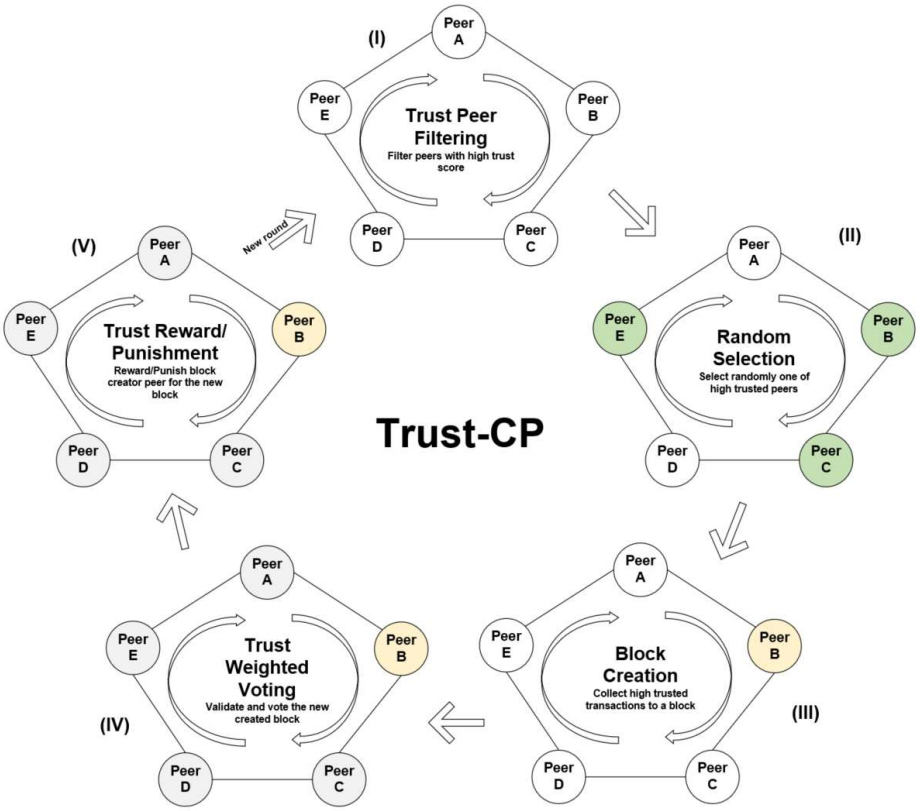

Figure 3: Trust Consensus Protocol [5]

To overcome these limitations, the authors in [5] introduce a novel Trust Consensus Protocol (Trust-CP). The protocol consists of five main phases (see Figure 3): Trust peer Filtering, Random Selection, Block Creation, Trust Weighted Voting, Trust Reward/Punishment. The key aspect of the proposed consensus protocol can be described as follows: All nodes part of the M2M community also participate in the blockchain network. The trust score of every node is continuously evaluated to ensure trustworthiness in the network. During the lifetime, transactions are sent from one node to other nodes. All transactions are assigned with the trust score of the transaction initiator (trust score if the sender node). These transactions are unconfirmed and are waiting to be approved by the blockchain network. Before the approval process starts, the transactions have to be included in a block. This is done by so-called block creators (leaders)

which are nodes selected from the blockchain network to perform these tasks. The authors in [5] propose that for every round of block generation, an algorithm is going through the nodes to select randomly one of them as block creator based on its trust score (see Figure 1, Phase I and II). After that, the block creator will collect pending transactions to a block (see Figure 1, Phase III), where it should consider only transactions with a good trust score. The generated block will be broadcasted to other nodes for validation and confirmation (see Figure 1, Phase IV). Other nodes will receive and verify the block by checking the trust score of the block creator node, the trust score of the transactions part of the block and the hash values of the block. If the block contains the right information and also fulfills the criteria of the system, it will then be positively voted by the validating node and the block is forwarded to other nodes. The criteria are fulfilled if the block is created by a trusted block creator, the block contains the right hashes and the transactions part of the block are also trusted. If the block does not meet the conditions, it will receive a no-vote. The votes are weighted based on the trust score of the validators. The different actions performed by the nodes part of the blockchain are rewarded or punished accordingly by increasing or decreasing the trust score of the performing node (see Figure 1, Phase V).

4 Integration of Trust Consensus Protocol for M2M Application Service Provision

4.1 Joining P2P Network and M2M Community

The peers acting as service providers and service consumers are connected P2P in [2, 4]. To fully decentralize the whole M2M application service provision, the authors in [2, 4] integrate a P2P layer into their layer model of decentralized networking. The P2P layer consists of the P2P communication layer which enables the information exchange between the peers using M2M communication protocols and the P2P overlay layer which realizes the distributed data storage using protocols, such as Chord or Gnutella. To join the P2P network, the authors in [4] propose to use a webpage/server for registration and for finding the contact information about the bootstrapping nodes of the P2P network. Afterwards, the new node will receive the necessary information to join the network. Additionally, the authors in [2, 4] introduce an M2M community to enable social networking in the P2P network. This M2M community is organized through the interface descriptions of the services which are provided by the peers. However, the authors in [4] do not consider the security aspects regarding joining a network and do not provide a concept how this process can be achieved in a secure and trustful manner. To optimize the entry to the P2P network and the associated M2M community, this publication proposes to integrate blockchain elements which have several benefits. Therefore, it is proposed to use the introduced trust evaluation system and the trust consensus protocol to manage the joining/leaving process in the M2M community.

Therefore, this publication proposes a novel joining mechanism (an overview is shown in Figure 4) for peers interested to enter the P2P network/M2M community. First, a new peer who wants to enter the M2M community needs to contact a bootstrapping node. The subdomains for these nodes can be resolved by using DynDNS. After a new peer has contacted a bootstrapping node (1), the bootstrapping node will test the peer and its services to issue an entry trust score for the new peer (2). The testing consists of evaluating the functional behavior and the performance of the new service. The computed entry trust score is sent (3) to all the other peers' part of the P2P network. Continuously, all participating peers will collect and store all joining proposals sent from different bootstrapping nodes in their local storage (4). Afterwards, the Trust Consensus Protocol (5) is going to be applied by all peers in the network. The leader is first selected based on a trust-based and random selection algorithm (only peers with high trust scores are considered). Based on the list of proposals, the leader will create a transaction consisting of a list of interesting nodes which should be elected for joining the M2M community. This list will be sent afterwards to other nodes which all act as validating nodes by checking the transaction and vote on it. The voting consists of analyzing the list of joining nodes and their trust score. Furthermore, the trust score of the leader is checked. If the predefined criteria are fulfilled (trust score of joining proposals and leader should be high), the transaction will get a positive vote which are then weighted with their own trust scores. If the required trust threshold for the voting is achieved, the list of the selected joining nodes (mentioned in the transaction) will be admitted to the M2M community. Further information regarding the Trust-CP can be found in section 3.

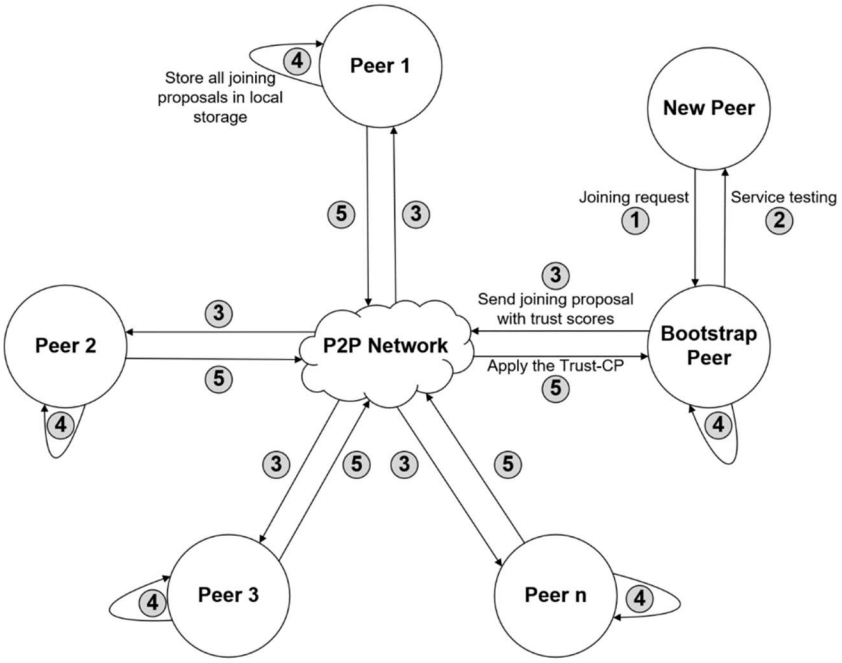

Figure 4: Trust-CP-based Joining

4.2 M2M Service Registration

The authors in [2, 4] design a Service/Application Registry (SAR) which operates in the overlay and is used to manage all services in the M2M community. A new service is registered in the SAR by a peer which is acting as a service provider. The service provider then stores the combination of service ID and contact information and makes the Interface Description (IFD) of the service available to other end-users. It could be that several end-users offer the same services in terms of identical IFDs but with different contact information. The same services acting as individual instances of their services are stored with the same service ID in the SAR. End-users acting as service consumers can look up using keywords for specific services and have the possibility to select an instance of a service for the application configuration. A disadvantage of the existing approach is that every end-user acting as service provider can register any service without considering the functionality or the security of it. Other end-users (service consumers) will not have the possibility to check if the instance of a service is trustworthy or not. The authors in [6] propose to test the functionality and performance of the new services and to obtain the initial trust score which is made available for the whole community. However, even though the approach presented in [6] gives a better trust overview for all M2M community members, it does not mitigate that untrustworthy services are registered in the SAR.

To optimize the service registration process, this publication proposes to use the proposed Trust-CP for decision making among the nodes in order to agree on the same trust score for a new service. Figure 5 shows the registration of a new M2M service and the creation of a new and extended Interface Description (IFD) (containing trust information about the service) based on the tests performed by the community members. First, every service provider has to register the individual instance of a new service with the SAR (1). The IFD of the new service will be stored in the SAR. Other peers' part of the M2M community can make a request to the SAR for new services (2). The SAR will then provide the service IFDs of the new registered services to the requesting peers (3). The different community peers will test and evaluate the new services and every testing peer will create based on the test results a new extended IFD of the new service (4). The extended IFDs will be shared among the M2M community and a consensus for the same extended IFD is required (5). The Trust-CP ensures that all participating nodes agree on the same copy of the IFD. On basis of the Trust-CP, the selected leader and the validating nodes decide whether or not to accept the registration of that service. The new extended IFD will be registered (6) in the SAR and can be used by the community to decide whether or not they want to subscribe to the service. Same can also be done for deregistration of a service, where the community members test an existing service and agree using the Trust-CP for deleting the IFD from the SAR. Moreover, the service registration approach presented in this section helps to figure out if the joining process of a peer (described in section 4.1) is done correctly or not by comparing the trust score computed by testing the service during the joining process with the trust score computed in the service registration step.

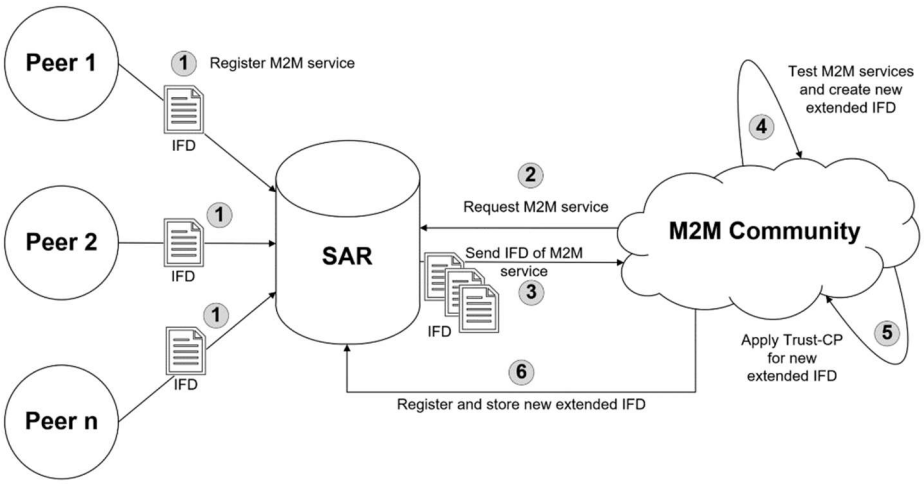

Figure 5: M2M Service Registration and extended M2M Service IFD

4.3 Cooperative M2M Application Design

Every end-user in the M2M community has the possibility to design new cooperative M2M applications by configuring and selecting different available services based on the own interest. However, as mentioned before, multiple end-users can offer different instances of the same M2M service and the decentralized M2M service architecture does not support a centralized coordination regarding the direction of the peers to which specific instances of a service they should connect to realize the cooperative M2M application. The authors in [2, 4] introduce a combination of random and manual selection of service instances which is not a fair and secure solution for M2M application configuration. As result, an unstable or malfunctioning cooperative M2M application is created and provided in the M2M community. To overcome this problem, the authors in [6] propose to consider only service instances with a good trust score for being part in the random selection and application composition process. However, the approach in [6] does not provide a fair way to select the service instances and also relies only on the subjective decision of every single service consumer.

In order to optimize the trust-based selecting approach, this publication proposes to include all participating peers in a fair voting system regarding the designed cooperative M2M application. This also enables load balancing among the peers regarding the participation of their services in a cooperative M2M application. The novel application configuration for a cooperative and distributed M2M application service is shown in Figure 6. Specifically, this means that after one end-user has designed a cooperative M2M application, the corresponding SCXML application description is stored in the P2P overlay where other end-users can retrieve the description and can analyze the service chain defined in the application description. If a service consumer (SC) wants

to use the application defined in the SCXML application description (1), it will send a request (2) to the M2M community where other participating nodes will propose (based on the application description) an instance of a service for every position in the service chain (3). These proposals will be sent to all community members (4). Using the Trust-CP (5), a leader is selected which takes one proposal (based on predefined trust criteria) from the pool of service chain proposals and sends them to all nodes for voting and validation. Other nodes receive the proposal and check if the leader and the instances selected for being part of the service chain are trustworthy. The voting will conclude with the decision about the definitive list of instances upon which all community members agreed. This list of service chain instances will be sent to the service consumer (6) which then will start contacting the specific service instances in order to allow the application configuration. Finally, after the application configuration, the application execution will start, considering only trustworthy services part of the service chain.

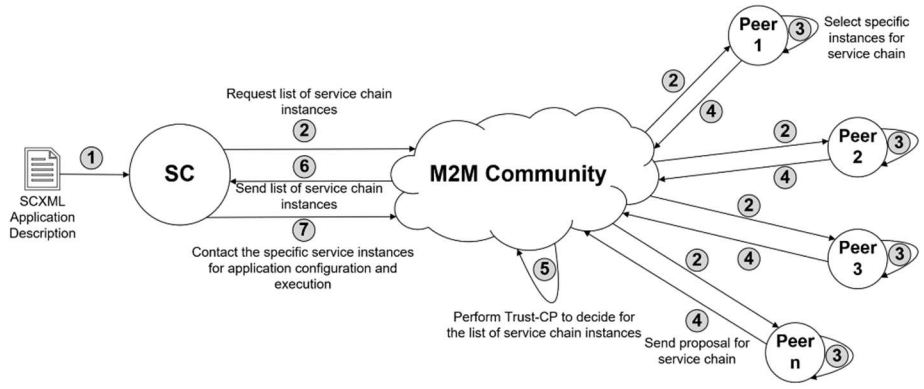

Figure 6: M2M Application Configuration

5 Conclusion

The benefits of using the blockchain technology are ensuring data integrity and non-repudiation. Moreover, blockchain-based consensus protocols enable the agreement of the same state of art in a fully decentralized network and motivate the participating nodes to be actively involved in decision-making processes. Trust evaluation systems provide the possibility to measure the trust score of entities in a community. Trust relationships between participating nodes acting in a network without a central manager are very important to overcome potential security risks. The features of blockchain and trust evaluation systems are used to improve existing end-user based M2M service platforms and communities. Therefore, this research publication proposes to integrate trust and the trust evaluation processes in different parts of the M2M application service

provision lifecycle. Considering decision making and data integrity, the blockchain and a novel Trust Consensus Protocol is also integrated. The improvements of the M2M application service provision process includes a secure and trustworthy joining mechanism to the P2P overlay and the M2M community. Moreover, the service registration and the cooperative M2M application design is improved through the use of trust and consensus protocol.

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. Kim, J., Lee, J., Kim, J., Yun, J.: M2M Service Platforms: Survey, Issues, and Enabling Technologies. IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 61-76 (2014)
2. Steinheimer, M., Trick, U., Fuhrmann, W., Ghita, B.: Autonomous decentralised M2M Application Service Provision. Proceedings of the Seventh International Conference on Internet Technologies and Applications (ITA 17), pp. 18-23, Wrexham, UK, IEEE (2018).
3. Kim, Y. J., Kim, E. K., Nam, B. W., Chong, I.: Service composition using new DSON platform architecture for M2M service. International Conference on Information Networking (lCOIN), pp. 114-119 (2012).
4. 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).
5. 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)
6. Shala, B., Trick, U., Lehmann, A., Ghita, B. and Shiaeles, S.: Trust-based Composition of M2M Application Services. 10th IEEE International Conference on Ubiquitous and Future Networks (ICUFN 2018), Prague, Czech Republic (2018).
7. 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.
8. ITU-T, Recommendation Y.3052 (2017): Overview of trust provisioning in information and communication technology infrastructures and services.
9. Distributed ledger technology: beyond blockchain, Government Office for Science (2016) available from: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf (accessed 10 July 2019).
10. Survey on Blockchain Technologies and Related Services FY 2015 Report, Japan, 2016 available from: https://www.meti.go.jp/english/press/2016/pdf/0531_01f.pdf (accessed 22 July 2019).