On the use of blockchain technologies in smart home applications

IOP Conf. Series: Materials Science and Engineering **878** (2020) 012023 doi:10.1088/1757-899X/878/1/012023

On the use of blockchain technologies in smart home applications

G Pazhev[1], Gr Spasov[1], M Shopov[1] and G Petrova[2]

Technical University of Sofia, Plovdiv branch, [1]Computer Systems and Technologies Department, [2] Department of Electronics, 4000 Plovdiv, Bulgaria, georgpajev@gmail.com, {gvs, mshopov, gip}@tu-plovdiv.bg

Abstract. The paper presents an implementation of a smart home architecture with two gateways based on collaboration of blockchain technology and Internet of Things (IoT) Message oriented middleware (MOM) architecture for management of internal home appliances. A basic overview of IoT MOM, MQTT (Message Queue Telemetry Transfer) protocol and blockchain architectures is made and the key characteristics and consensus algorithms are presented. Some challenges and the Ethereum smart contracts concept of the blockchain technology are discussed.

1. Introduction

Smart home, in its essence, is a heterogeneous sensor network based on technologies such as Internet of Things (IoT). They are nowadays equipped with many devices such as smart meter, in-home displays, renewable energy sources and storage, and smart appliances such as washing machine, refrigerators, TV, oven, thermostat, HVAC (Heating Ventilation and Air Condition), lights, and plugs for electrical cars. For transparent communication between these devices it is necessary to use gateways for protocol transformations. There is a two-way communication that is utilized in the demand response, advance metering infrastructure, distributed energy generations and storage. The U.S Department of Energy presents two types of home area network architectures – utility-managed and utility and consumer managed. In the utility-managed architecture the utility monitors, controls and manages smart home appliances via his private network using utility gateway connected to this network. The utility and consumer managed architecture extends the utility-managed architecture with a common gateway/hub which acts as an intermediate device between the utility, home owners, third party service providers and the home appliances [1].

On one hand, the presence of many gateways complicates the system in the internal network of sensors and actuators. The presence of additional protocol transformations slows its functionality, and in order to reduce their number, it is necessary to select protocols that are compatible with terminal devices with limited computing capabilities. The presence of MQTT and COAP (Constrained Application Protocol) meet the requirements for devices with limited computational resources and are very convenient to use for building an internal sensor network. On the other hand, the system interface with the outside world must be compatible with different utility providers and their communication protocols. The presence of many heterogeneous devices in IoT leads to compatibility, security and discovery issues, both internally and externally. The blockchain technology, with its unified set of protocols and build-in security, could be used not only as an external link to any home-based network within the house, but also as a distributed database, that stores data contained in each transaction. That data can then be used for further analyses.

In this paper some of the issues and challenges regarding integration of blockchain technology and IoT devices are identified and a use-case implementation of a smart home application is presented.

2. Description of technologies.

2.1. Internet of Things

With the development of sensor networks, wireless mobile communication, embedded system and cloud computing, the technologies of IoT have been widely used in areas such as Smart Cities, public security, Smart homes and so on [2]. There are three essential components of IoT [3]: embedded devices – consisting of both low cost/low power devices and high-end gateways; scalable connectivity – each embedded device should be connected; cloud-based mass device management – centralized management of distributed devices.

The basic idea behind IoT is the pervasive presence around us of a variety of things or objects – such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, etc. – which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbours to reach common goals [4].

2.2. MQTT

Message Queue Telemetry Transfer (MQTT) consists of three components - subscriber, publisher and broker [5], as shown on figure 1. The publisher input messages to a specific topic, the broker dispatches these messages to the subscribed clients. Clients subscribe to particular topics relevant to them and receive each message published to those topics. The broker achieves security by authorizing the publishers and the subscribers.

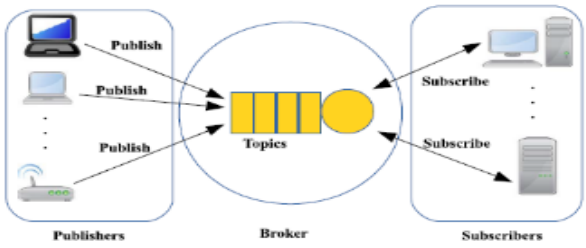

Figure 1. MQTT protocol block diagram [5]

2.3. Blockchain

Blockchain, at its core, is a peer-to-peer distributed ledger that is cryptographically secure, append-only, immutable (extremely hard to change), and can be updated only via consensus or agreement among peers [6]. There are various definitions of blockchain depending on point of view. From a business point of view a blockchain can be defined as a platform whereby peers can exchange values using transactions without the need for a central trusted arbitrator. This is a powerful concept and once readers understand it, they will realize the tsunami potential of blockchain technology. This allows blockchain to be a decentralized consensus mechanism where no single authority is in charge of the database. From a technical point of view blockchain can be thought of as a layer on the distributed peer-to-peer network running on top of the Internet, as shown in figure 2.

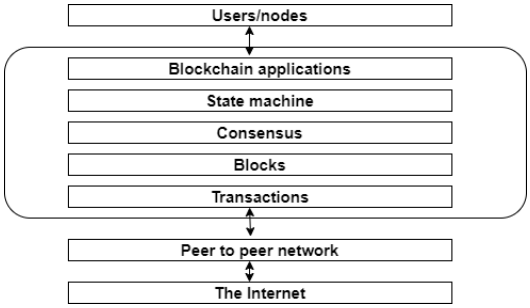

Figure 2. Blockchain as a layer on top of peer-to-peer network [6]

Each user, which participates into the blockchain network, is represented as a blockchain node that has an address, which is represented as a unique identifier. Each node can transfer values to other ones by initiating transactions inside the same network. All initiated transactions are grouped into blocks, which are validated by special nodes in the network called miner nodes. The miner nodes validate the newly created blocks by executing of consensus algorithm. All valid blocks are stored in the distributed ledger as shown in figure 3.

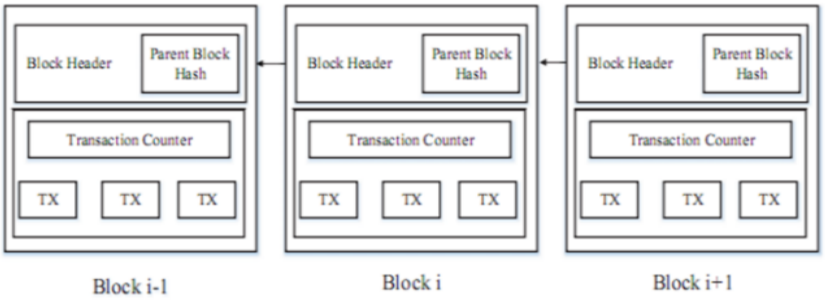

Figure 3. Distributed ledger structure [7]

The ledger is a chain of blocks, which contains all valid transactions (illustrated as TX blocks in figure 3). The first block at the chain is called genesis block. Each block is linked with previous one by Merkle Tree root hash located at his header. Being a decentralized system, blockchain systems do not need a third-party trusted authority. Instead, to guarantee the reliability and consistency of the data and transactions, blockchain adopts the decentralized consensus mechanism [8]. The essential of each consensus mechanism is to make the miner nodes to solve a puzzle to prove the truth of data. In the existing blockchain systems, there are four major consensus mechanisms: PoW (Proof of Work), PoS (Proof of Stake), PBFT (Practical Byzantine Fault Tolerance), and DPoS (Delegated Proof of Stake). The PoW consensus algorithm forces the miners to solve a computationally-intensive ease verifiable task by reaching a target value based on nonce value, hash of previous block and transaction hashes to create a new block. If the task is solved by anyone of nodes inside the network, then this node broadcasts the solution to other miner nodes and all other nodes must mutually confirm the correctness of solution. If the block is validated, other miners would append this new block to their own copy of the ledger. The PoS consensus algorithm is an energy-saving alternative of PoW. This algorithm randomly selects the node responsible to create the new block based on amount of coins that owns. It is based on fact, that the users who contain more coins are more interested the blockchain system to work correctly and these owners are more responsible to provide the ledger to contain valid blocks. The PBFT is proposed by Liskov and Castro as a replication algorithm for toleration of byzantine faults in distributed systems. In this algorithm the replicas move through a succession of configurations called views [9]. In a view one replica is the primary and the others are backups. When the primary receives a client request it starts a three-phase protocol to atomically multi-cast the request to the replicas as shown on figure 4. The three phases are pre-prepare, prepare, and commit. The pre-prepare and prepare phases are used to totally order requests sent in the same view even when the primary, which proposes the ordering of requests, is faulty. The prepare and commit phases are used to ensure that requests that commit are totally ordered across views.

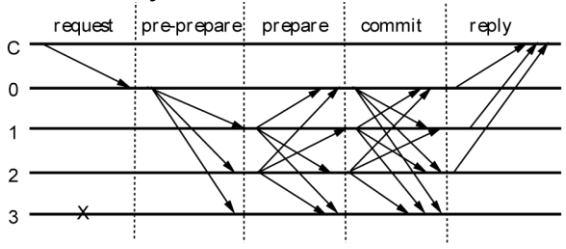

Figure 4. Practical Byzantine Fault Tolerance algorithm [9]

The DPoS consensus algorithm differs from PoS in that PoS is direct democratic while DPoS is representative democratic. In DPoS the stakeholders elect their delegates to generate and validate blocks. The parameters of the network such as block size and block intervals could be tuned by delegates. With significantly fewer nodes to validate the block, the block could be confirmed quickly, leading to the quick confirmation of transactions.

2.4. Ethereum and Smart contracts

Ethereum is an open source software based on blockchain technology, which focuses on providing a platform for building distributed blockchain applications (dApps) [10, 11]. The developer can implement, deploy and execute his application called smart contract in Ethereum network. The smart contract applications are executed by Ethereum Virtual Machine (EVM). Ethereum provides two types of accounts – user and smart contract. The user account does not execute any code. This type of account can only send messages to other accounts by creating and signing transaction using their private keys. The contract accounts always execute the code of any method defined in the deployed corresponding smart contract invoked by transaction. The Ethereum smart contracts are written in domain specific language (DSL) called Solidity. The code of the contract is compiled to byte-code and then is deployed to EVM. Figure 5 illustrates with a real world example how the smart contract acts as trusted intermediary between two users (a seller and a buyer).

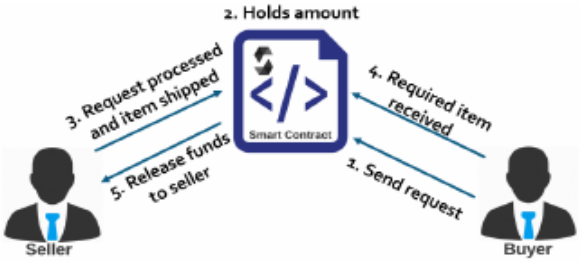

Figure 5. Smart contract execution [11]

The whole transaction which executes smart contract application completes in five steps. In the first step, the buyer sends amount of Ethers to the smart contract's address and this smart contract holds the ethers. In the second step, the smart contract notifies the seller indicating for the sent request by the buyer. In the third step, the seller checks and verifies the buyer's request and if it is valid and the amount of ethers are enough then the seller sends to the buyer the purchased item and inform the smart contract with shipment message. In the fourth step, if the buyer receives the item, then the smart contract is updated with the delivery status. In the last fifth step, after all other steps are successfully executed the hold ethers in the contract are send to the seller's account.

Each smart contract which is deployed on the blockchain network executes variety of elementary computational operations provided by EVM. The complexity of smart contract is evaluated by the number of these operations, which executes. This complexity is measured by the EVM in units named gas. The gas unit is mapped to ether costs as a fee, which is taken over by the user which initiates the request to the contract (the buyer in this case). The smart contract could contain two basic types of methods – methods which set values in their private members (set-methods) and view methods. The invocation of set-methods requires a transaction to be initiated for sending the value to be set in the corresponding private member inside the contract. The execution of this method costs the fee determined by the gas of contract. If this method is set as payable, then the amount of required ether, which will be hold by the contract, should be added to the transaction fee. The invocation of view methods does not require to initiate a transaction to view the value of private member of the contract, however its invocation does not require any fees. If we return to the example with the seller and buyer mentioned above, then the buyer requests items by the seller by invocation of payable set-method provided by the smart contract. In this situation, the buyer sends the amount of ether which will be hold by the contract for the price of items and pays the fee price of transaction determined by the gas of the contract.

3. Challenges

3.1. IoT Challenges

The challenges addressed to IoT and their middlewares are related to dynamic heterogenous resource discovery and composition, scalability, reliability, interoperability, security and privacy [12].

Resource discovery: The nature of the IoT infrastructure nullifies centralized resource registries and discovery approaches. A balance must be achieved between registry distribution and the number of registries. Fewer registries provide consistent and fast discovery of resources under normal circumstances. A problem may arise if there is large number of service discovery queries in IoT applications.

Resource management: Resource conflicts may occur in IoT applications that share resources. A conflict resolution is required to resolve conflicts in resource allocation among multiple concurrent services or applications.

Scalability: The most existing middlewares for IoT are centric around Wireless Sensor Networks (WSNs) and their network level scalability is also limited from WSNs. There is a high possibility that they will perform poorly in IoT's ultra-large-scale network. That's why it is important every component of IoT middleware to be scalable in order to achieve system-wide scalability.

Reliability: To achieve IoT middleware reliability, every component or service of a middleware needs to be easily replaceable. A better understanding of the dependency between reliability and other requirements is needed.

Interoperability: Although network interoperability is supported by most existing middlewares there are some that lack support for semantic and syntactical interoperability. There is a lack of standard in ontologies, which creates a big challenge for IoT. The best support for semantic interoperability is offered by the service-oriented approach, but the support for syntactical interoperability is limited. There is a need for better understanding of IoT services' syntax and semantics.

Security and privacy: Although all the concerns of security, privacy, and trust in all the technologies used in IoT are clearly present in the context of the IoT, they are not fully complete. Most existing middlewares' authentication-based partial security solutions are insufficient for a number of IoT applications. There is a need of research for a holistic security solution that takes care of system as well as middleware level security and privacy aspects.

3.2. Blockchain challenges

The Blockchain technology resolves partly the already mentioned challenges of IoT. In this context, knowledge of the Blockchain challenges is required. Some of them are [13]:

Storage capacity and Scalability: Storage capacity and scalability have been deeply examined in blockchain. In this technology, in Bitcoin for example, the chain is always growing at a rate of 1MB per block every 10 minutes, and there are copies stored among nodes in the network. Although only full nodes (nodes that can fully validate transactions and blocks) store the full chain, storage requirements are significant. As the size grows, nodes require more and more resources, thus reducing the system's capacity scale. In addition, an over-sized chain has negative effects on performance, for instance, it increases synchronization time for new users.

Security: This challenge requires knowledge of some of the most common attacks such as a 51% attack or a majority attack. This attack can occur if a blockchain participant is able to control more than 51% of the mining power. In this situation he can control the consensus in the network. The boom and fast evolution of mining pools, have increased the probability of this attack occurring, which in turn could compromise the integrity of Bitcoin. Another example for attack is the double-spend attack, which consists in spending the same coin twice.

Anonymity and Data Privacy: In public blockchain the ledger is public. However, all transactions are publicly visible for everyone across the blockchain network. Combined with traffic analysis, transactions could be linked back to their source IP (Internet Protocol) addresses, which points to the transactions originator. This is a big concern in privacy point of view. Various proposals have been

made to address the privacy issue since the introduction of Bitcoin. These proposals could be divided into two categories: mixing protocols and inherent anonymity.

3.3. Integration of IoT and Blockchain

Another aspect necessary to take into account is related to the IoT interactions, i.e., the communication between the underlying IoT infrastructures. When integrating blockchain, it is necessary to be decided where these interactions will take place: inside the IoTs, through blockchain, or a hybrid design involving IoT and blockchain (figure 6). These three approaches are clearly described by [13]:

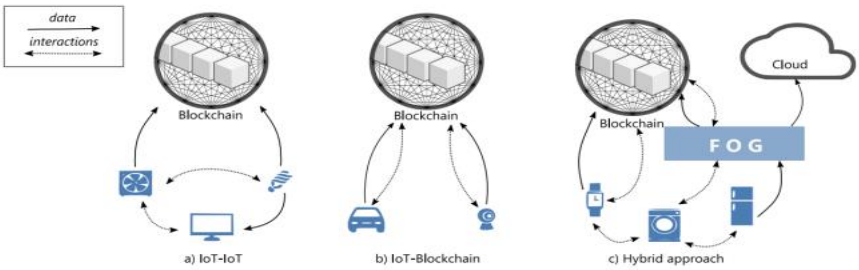

Figure 6. Blockchain IoT interactions [13]

Interactions IoT-IoT: this approach could be the fastest one in terms of latency, and security since it can work offline. The IoT devices have to be able to communicate with each other, which usually involves discovery and routing mechanisms. Only a part of the IoT data is stored in blockchain whereas the IoT interactions take place without using the blockchain (figure 6.a). This approach would be useful in scenarios with reliable IoT data where the IoT interactions are taking place with low latency.

Interactions IoT-Blockchain: in this approach all the interactions go through blockchain, enabling an immutable record of interactions. This approach ensures that all the chosen interactions are traceable as their details can be queried in the blockchain, and moreover it increases the autonomy of the IoT devices. The IoT applications that intend to trade or rent can leverage this approach to provide their services. Nevertheless, recording all the interactions in blockchain would involve an increase in bandwidth and data, which is one of the well-known challenges in the blockchain (figure 6.b). On the other hand, all IoT data associated with these transactions should also be stored in the blockchain.

Hybrid approach: lastly, a hybrid design where only part of the interactions and data take place in the blockchain and the rest are directly shared between the IoT devices. One of the challenges in this approach is choosing which interactions should go through the blockchain and providing the way to decide this in run time. A perfect orchestration of this approach would be the best way to integrate both technologies since it leverages the benefits of blockchain and the benefits of real-time IoT interactions. In this approach fog computing could come into play and even cloud computing (figure 6.c), to complement the limitations of blockchain and the IoT.

4. Use case: A smart home application

A realization of home automation network based on open source hardware is presented in [14]. In this realization CNDEP (Controller Network Data Extracting Protocol) protocol is used for interaction between the gateways for the home appliances and the control gateway. Each device in CNDEP is identified by session id. The control gateway sends command id and sensor function id to extract the sensor data or contact with the actuator. However, with this smart home implementation there are the following challenges that need to be addressed: higher scalability, interoperability and security from the middleware point of view. Although it uses a resurrecting duckling as security policy, this is not quite enough to be highly secure. This home network implementation is not highly flexible concerning the scalability because it requires a new session id to be created to add a new device in the network. This requires the whole network to be re-configured.

Considering these challenges we propose realization of an IoT-based smart home network, using the concept with two gateways and integrating IoT and Blockchain technologies. Using the IoT-IoT

approach for integrating the two technologies we propose a smart home architecture (figure 7), where each smart home manages its own private sensor network with control gateway, which could share data from this internal network and could provide access to the actuators via its smart contract deployed inside the Blockchain network. The user or utility provider could connect to the corresponding smart home and exchange data with the provided smart contract. Every participant inside the blockchain network needs to create transaction for sending data. Each transaction is digitally signed and encrypted using asymmetric security algorithms. This advantage gives secure connection between the user/utility provider and the home network. From the other hand the transactions are saved in the ledger, however the shared data are also saved at the same place and could be used for further analysis.

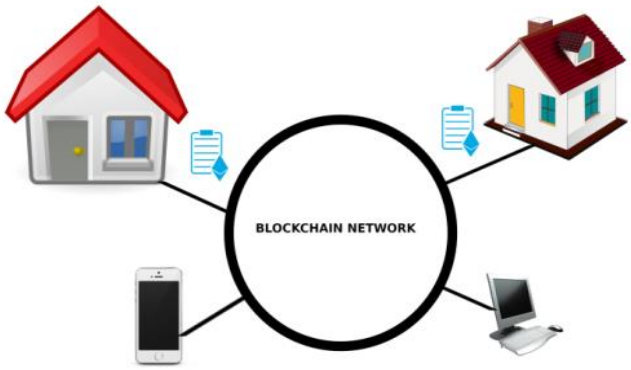

Figure 7. Smart contract execution

The smart contracts provide unified interface, which improves smart homes interoperability with the utility providers and user nomadic agents. Using blockchain network the utility provider does not need to install his own smart meter at each home and to provide own secured infrastructure. The smart home could share the sensor data and the actuators state to its own smart contract. The utility provider could connect to the same smart contract to extract the data. The smart contract also could be used as third-party software for secure connection to the home by the consumer through Internet.

In the blockchain network the participant who initiates the transaction to the smart contract pays the gas for execution of the smart contract. To decrease the number of payments of gas for invocation of the smart contract by the control gateway two approaches could be used. In the first approach, the control gateway could send transaction with all data to the smart contract inside the blockchain network if there are any changes in the sensor data. In the second approach, the consumer or utility provider could initiate a transaction for demanding of the sensor data. The control gateway extracts all the data from the sensors and actuators and creates a transaction to send these data to the blockchain network.

Our test-bed implementation of a smart home application, based on the integration of IoT and blockchain technologies is shown in figure 8.

Figure 8. Test-bed realization

The private home network is based on two-gateway architecture – gateway for each appliance inside the home (appliance gateway) and a single control gateway, which provides connection to the outside world by exchanging data with deployed smart contract at the Ethereum blockchain network. Each appliance gateway could contain sensors and actuators. They communicate with the control gateway using MQTT protocol. The control gateway acts as MQTT broker and provides specific topic for each appliance gateway for its sensors and actuators. In our test-bed implementation the appliance gateway is realized by wireless IoT module ESP8266 with connected TCN75A temperature sensor via I2C interface. This platform provides also a WiFi network interface for connection with the control gateway and sends the temperature data in degrees at the specific topic. For implementation of control gateway Olinuxino A20 is used. It has dual core A20 processor with 1GB of RAM and Ethernet interface for connection to the Internet. The control gateway has installed a go-ethereum client (geth) for connection with the blockchain, and mosquito MQTT implementation to work as MQTT broker. Olinixuno A20 platform extracts the data from the specific topic (e.g. temperature topic) and sends these data to the smart contract inside Ethereum Ropsten test blockchain network via JSON-RPC.

5. Conclusion

Every technology has its place within the IoT eco-system. The use of MQTT to build an internal network within a smart home covers a variety of devices with limited computational abilities and it is also characterized by low cost of communication. As a result of the experimental test-bed realization presented in this paper, the following conclusions are reached regarding the use of blockchain in collaboration with MQTT: the use of both technologies reduces the requirements of the control gateway to maintain a large set of protocols for communication with the internal and external networks, as well as the requirements for the use of a database to store the sensor data for further analysis. Used smart contracts provide a unified interface and can also be used as a means of distributed computing. The blockchain integration combined with smart contract feature contributes to the security and interoperability of the house with the external communication. There is a need for further research for exchanging and storage of privacy information, which requires to be visible only for limited range of participants.

6. References

[1] Hafeez A, Kandil N H, Al-Omar B, Landolsi T, and Al-Ali A R, 2014 Smart Home Area Networks Protocols within the Smart Grid Context *Journal of Communications* Vol. **9** No. 9 pp. 665-671
[2] Beng L 2009 Sensor cloud: Towards sensor-enabled cloud services (Intelligent Systems Center Nanyang Technological University)
[3] Geng W, Talwar S, Johnsson K, Himayat N and Johnson K D 2011 M2M: From mobile to embedded internet *Communications Magazine*, IEEE **49**, no. 4 pp 36-43
[4] Giusto, Iera A, Morabito G and Atzori L 2010 The Internet of Things (Springer ISBN: 978-1-4419-1673-0)
[5] Hasan H Evaluation of MQTT Protocol for E-Learning *International Journal of Computer Science & Mobile Computing* vol. **7** issue 11 pp. 57–67
[6] Bashir I 2017 Mastering blockchain (Packt publishing)
[7] Zheng Z, Xie S, Dai H, Chen X and Wang H 2017 An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends *IEEE 6th International Congress on Big Data* pp 557-564
[8] Lie X, Jiang P, Chen T, Xiapu L and Qiaoyan W 2017 A Survey on the Security of Blockchain Systems Future Generation Computer Systems (Elsevier)
[9] Liskov B and Castro M 1999 Practical Byzantine Fault Tolerance *Proceedings of the Third Symposium on Operating Systems Design and Implementation* (New Orleans, USA)
[10] Gencer A , Basu S , Eyal I , Renesse R and Sirer E 2018 Decentralization in Bitcoin and Ethereum Networks", Financial Cryptography and Data Security (http://www.arxiv.org/)
[11] Praitheeshan P, Pan L, Yuy J, Liuy J, and Doss R Security Analysis Methods on Ethereum Smart Contract Vulnerabilities — A Survey (*preprint http:/www./arxiv.org/)*

[12] Razzaque M A, Milojevic-Jevric M, Palade A, and Clarke S 2016 Middleware for Internet of Things: A Survey *IEEE Internet of things journal* vol. **3** No. 1 pp.70-95
[13] Reyna A, Martin C, Chen J, Soler E, Diaz M 2018 On blockchain and its integration with IoT. Challenges and opportunities (Future Generation Computer Systems)
[14] Petrova G, Spasov Gr, Pazhev G 2015 Realization of a Home Automation Network with Two Gateways Based on Open Source Hardware *Annual journal of electronics* pp. 116-119

Acknowledgments
This work was supported by the European Regional Development Fund within the Operational Programme “Science and Education for Smart Growth 2014 - 2020” under the Project Center of Competence "Intelligent mechatronics, eco- and energy-saving systems and technologies“BG05M2OP001-1.002-0023-C0.