Received 26 April 2016; | Revised 6 June 2016; | Accepted 6 June 2016

DOI: xxx/xxxx

INTERNET TECHNOLOGY LETTERS

Blockchain as a Service: securing bartering functionalities in the H2020 symbIoTe framework

Pietro Tedeschi[1,2] | Giuseppe Piro*[1,2] | Jose Antonio Sanchez Murillo[5] | Nemanja Ignjatov[4] | Michał Pilc[3] | Kaspar Lebloch[4] | Gennaro Boggia[1,2]

[1]Department of Electrical and Information Engineering (DEI), Politecnico di Bari, Bari, Italy, Email: {name.surname}@poliba.it

[2]CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni Parma, Italy

[3]PSNC, Poznań Supercomputing and Networking Center, IBCh PAS, Poland, Poznań, Email: {name.surname}@man.poznan.pl

[4]University of Vienna, Cooperative Systems Research Group, Vienna, Austria, Email: {name.surname}@univie.ac.at

[5]Atos, Madrid, Spain, Email: jose.sanchezm@atos.com

Abstract

Blockchain is emerging as a promising technology able to support transparent, secure, and immutable transactions traceability in decentralized networks. Its usage in many application domains, including the Internet of Things, is gaining the attention of even more researchers and industries worldwide. In line with current research interests, the work presented in this letter has been carried out in the context of the European H2020 symbIoTe project. Among its main features, the symbIoTe framework offers bartering functionalities across a federation of Internet of Things platforms. This letter extends the baseline implementation of bartering functionalities and formulates a novel methodology that properly integrates and takes advantages from the Blockchain technology. Even if the proposed approach is general, the main facets characterizing the conceived approach are illustrated through a fictional use case envisaging the provisioning of Intelligent Transportation System and air pollution services in a Smart City.

KEYWORDS:
IoT, Blockchain, Bartering, Smart City

1 | INTRODUCTION

Recently, the *Blockchain* technology attracted a growing interest in finance, telecommunications, and Information Technology domains. It represents a distributed ledger of immutable information, stored in a list of blocks that are fully replicated in logical entities forming a peer-to-peer network[1]. A block contains one or more transactions, its own cryptographic hash value, the hash of the previous block, and a timestamp. Each block is validated and added to the ledger according to a mining process, which implements a specific consensus protocol, such as Proof-of-Work (PoW), Proof-of-Stake (PoS), Proof-of-Burn (PoB), Proof-of-Authority (PoA), Byzantine Fault Tolerant (BFT) and Federated Byzantine Agreement (FBA)[2]. The resulting chain is immutable (i.e., blocks cannot be withdrawn) because the tasks needed to modify a block stored in the past, update the whole chain, and share the new list of blocks among all the copies of the ledger in the network are extremely complex and require an huge computational power. At the same time, the chain is also resilient against double-spending and Sybil attacks[3]. The potential of Blockchain can be further enhanced with *smart contracts*. In particular, a smart contract is generally used to record the terms of an agreement between two actors in a distributed ledger, in a self-verifying, self-executing and tamper resistant manner[1]. Once compiled into a virtual machine bytecode, it is published (and validated) as a transaction. Moreover, its execution can be triggered by means of another transaction, sent and validated across the Blockchain.

Blockchain is widely considered a key enabling technology for advanced services. For instance, by capitalizing on these promising properties, researchers and industries are trying to integrate the Blockchain technology into the Internet of Things (IoT) context[4]. Some interesting results achieved so far refer to security functionalities (e.g., authentication, access control, and intrusion detection), lightweight implementations, and shared economy applications[5 6 7 8].

With the aim of significantly extending the current state of the art in this exciting research area, this letter investigates the possibility to efficiently use Blockchain technology and smart contracts for designing advanced functionalities initially conceived by the European H2020 symbIoTe project[9]. Specifically, symbIoTe targets the definition of a federation of IoT platforms where implementing resource sharing and bartering functionalities in a flexible, unified, and secure way. Indeed, starting from the baseline solutions developed by the project, this letter formulates a novel methodology that see Blockchain technology and smart contracts as crucial technical components enabling bartering functionalities, while guaranteeing an immutable trustworthiness of enabled services. Even if the proposed approach is general, the main facets characterizing the conceived approach are illustrated through a fictional use case envisaging the provisioning of Intelligent Transportation System and air pollution services in a Smart City.

The remainder of this paper is organized as follows: Section 2 presents a review of the state of the art and provides an overview of the European H2020 symbIoTe project; Section 3 describes the symbIoTe procedure conceived within the European H2020 symbIoTe project and discusses its novel implementation based on both Blockchain technology and smart contracts; Section 4 summarizes the conclusions of the work and draws future research activities.

2 | LITERATURE REVIEW

2.1 | Securing IoT using Blockchain

Securing operations represents a keystone requirement for the IoT. Therefore, Blockchain is seen as a possible way to improve IoT network security, mainly in the area of identity management, access control, authentication and authorization, and intrusion detection. First of all, Identity Management (IdM) systems have been initially considered as reference mechanisms for authenticating and authorizing users within a network. Unfortunately, they do not scale in scenarios with a high number of IoT devices. However, new solutions emerged so far rely on immutability and cryptographic strength of the Blockchain technology for securely storing users' and devices' identities. For instance, a Blockchain-based Identity Framework enabling an identity self-management within a given IoT platform is discussed in[5]. Moreover, a lightweight consensus mechanism leveraging on a distributed scheme to maintain Blockchain security and privacy, while satisfying typical IoT requirements expressed in terms of communication latencies and resource usage, is presented in[6].

Web of Trust represents a novel initiative to create decentralized Public Key Infrastructure (PKI) based on Blockchain and provides models of self-sovereign identity that use X.509 certificates by storing the public key into the Blockchain. While digitally signing each transaction to push into the Blockchain, any entity is able to prove its identity, thus leading to an enhanced automation of IdM and authentication services in the IoT.

Sovrin[1] extends the aforementioned approach by allowing to connect to the Blockchain additional information related to the end users. Specifically, sensible and private information are stored within a so called *off-chain* for preserving users' privacy. Blockchain, instead, just contains pointers to where these user data may be retrieved. A permissioned Blockchain technology is used in[7] for managing access control and key management functionalities. More in general, the work presented in[8] claims that Blockchain could provide a Global Unique Identifier and a set of asymmetric key pair to each IoT device. Other contributions use Blockchain for different security services. For instance, a detection and prevention system for the IoT is presented in[10], where Blockchain is strongly advocated for building intrusion event datasets. Additionally, case studies for Blockchain-based security maintenance exemplified by Smart Home IoT platforms were described in[5] and[6].

The consensus approach built on top of the PoW algorithm produces a significant computational overhead. It brings an inapplicability of Blockchain to most of the IoT devices with limited storage and processing capabilities, mostly when considering that fully replicated Blockchain should be stored onto devices. The simpler way to use Blockchain in the IoT context is discussed in[5]. Here, the database is replicated into a single device having enough processing power to mine, process, and store blocks. Otherwise, different data structures and lightweight consensus mechanisms (like those based on the Byzantine Generals problem) should be taken into account[11]. The block-less Blockchain represents a valid solution in this direction. In general, it

[1] https://sovrin.org/

requires that miners should have a partial replication of Blockchain's contents. Moreover, complex consensus mechanisms are not required any more.

IOTA platform [2] is a representative implementation of a block-less Blockchain, where nodes namely *Tangle*, are not required to reach a consensus for storing valid transactions into the ledger, but they only need to run a tip selection algorithm for deciding which transaction should be orphaned in case of conflicts. *Hashgraph*[3] is an alternative lightweight Blockchain implementation that offers high scalability that intends to provide a new form of distributed consensus to address the inefficiency due the PoW.

IoT systems are capable of sensing information about user and environment and transmitting them into the public Internet. Blockchain can provide highly automated means for agreeing on parameters for information exchange, like Quality of Service (QoS), Service Level Agreement (SLA), Vouchers, etc. Therefore, by incorporating immutable, backward-traceable reputation systems it can improve current mechanisms for Bartering, tracking of goods, and more in general supporting shared economy strategies. For example, [12] and [13] use Blockchain for trading sensor data of IoT devices and other goods, by using different approaches to negotiate and gain access to the sensor data. The work [12] proposes to use keys to access sensor data and multi-signed transactions as a means for Bitcoin exchange with commodities. On the other hand, [13] describes a model where data are being purchased directly from sensors, which represents a highly automated mechanism for the exchange of goods.

2.2 | An overview on the European H2020 symbIoTe project

symbIoTe [9] is an H2020 project funded by European Commission that aims to improve the interoperability between different IoT platforms. It provides a solution to federate IoT platforms that will be able to share resources between them, granting access to data of sensors, actuators and virtual services to users of any platform of the federation. These resources could be shared through bartering functionality that represents a procedure that supports the exchange of goods or services between parties belonging to different, but federated, IoT platform, where no economic transaction is involved. Here, vouchers subsume the SLA (including the type of goods) and timing details. The two parties publish SLAs that describe the resources they want to exchange. Thus *bartering* is designed for a user that tries to access a resource in another's platform (where the user is not registered in) defined in the federated ecosystem.

3 | BLOCKCHAIN IN SYMBIOTE

In order to mitigate single points of failure, keep track of interactions among the nodes and execute transactions and agreements automatically during the bartering procedure, this contribution envisages the possibility to implement bartering functionalities through Blockchain and smart contracts. The main facets characterizing the conceived approach are illustrated through a fictional use case envisaging the provisioning of Intelligent Transportation System and air pollution services in a Smart City.

3.1 | Example use case

Sensors Inc. is a fictional company based in Spain with several deployments of environmental sensors across different cities. With these deployments and agreements of collaboration between several municipalities, they have built a smart routing application that drivers can use to avoid traffic jams and at the same time collaborate by reducing pollution in highly polluted zones of big cities. Madrid is one of those cities close to industrial zones where pollution can be a severe problem in the dry days. To avoid that, Madrid municipality gets an agreement with Sensors Inc. to promote their smart routing application. In this deal, Madrid gains the possibility of getting pollution data from sensors all around the city by Sensors Inc. and the latter gains the possibility of getting traffic information from sensors deployed by the city in key zones. Let P_A and P_B be Sensors Inc.'s platform Madrid's platform, respectively. Let A_A the smart routing application (native to and registered in platform P_A), and A_B the Madrid's pollution maps application (native to and registered in platform P_B). Application A_A will be granted access to resources in the foreign platform P_B if platform P_A grants access in the future to another application A_B. Without loss of generality, it is possible to assume that for every access platform P_B grants to an application A_A, platform P_A should grant just one access to resources for applications of type A_B. But, further and more complex interactions can be defined for valuable or expensive resources, like 1 to N accesses, unlimited access during a period of time and so forth.

[2] https://www.iota.org/
[3] https://www.hederahashgraph.com/

3.2 | Baseline approach implemented in symbIoTe

In the baseline symbIoTe framework, the accountability of access mechanisms is implemented through the concept of *coupons*. Each time an application A_A tries to access a resource in a foreign platform P_B, the following process takes place. As an initial state, neither platform P_A or platform P_B are in the possession of valid coupons from any other platform. Therefore, first, application A_A request access to a traffic sensor in platform P_B. Second, platform P_B communicates with platform P_A and asks for a coupon. Since platform P_A does not have a valid coupon from platform P_B, transmitted from previous interactions, it generates one coupon C_A. This coupon is a promise to platform P_B that, when presented, it will grant access to one of platform P_A resources. Once generated, it sends this information to symbIoTe[9] framework for accountability and validation and then returns this coupon to platform P_B. Third, when platform P_B receives this coupon, it validates it again by means of symbIoTe framework. If valid, then it stores it for future usage. Finally, it grants access to the resource to application A_A. Now let us suppose that an application A_B tries to access platform P_A. First, platform P_A communicates with platform P_B and asks for a valid coupon. Second, platform P_B already has a coupon C_A from a previous interaction so it sends it to platform P_A. Third, P_A validates in symbIoTe framework that this coupon has not been used. On success, it marks the coupon as used, informs the symbIoTe framework of this consumption and grants access to application A_B. When this cycle completes, a *bartering* transaction is finally realized, meaning that application A_A got access to one resource in platform P_B in exchange for application A_B getting data from a resource in platform P_A.

3.3 | Advanced solution based on Blockchain and smart contract

The baseline approach described before, there is a central authority (that is the core entity of the symbIoTe framework) managing the validity of different coupons that are generated. This means that this validation process needs to be running somewhere in either party's premises or a trustworthy third-party one. The use case considered in this letter, instead, envisages that:

- Madrid region has more than four million vehicles registered, and being the country's capital, most of them will pass at one point by the city. It means that at any point, hundreds of thousands of users might be accessing traffic or pollution data, so the system validating coupons will need to support this kind of request flow;
- Even if both platforms trust each other, having one of them hosting the coupon validation system means that it can alter it to benefit its interests by returning valid messages when its own platform validates invalid coupons. Having a third party hosting it does not solve the problem since it can turn malicious too.

Based on these premises, a decentralized, neutral, and robust solution able to validate transactions between two parties and enforce them when needed is highly required. Also, Blockchain and smart contracts appear as excellent solutions to solve the problem. With reference to a generic X entity, let $\{PK_X, SK_X\}$ be the public and the secret key, respectively. The contract $C(P_A, P_B)$ signed between platform P_A and platform P_B is formalized as in the following: platform P_A guarantees to platform P_B that, when one of its application A_B comes with this coupon in the future, it will get access to resources available in platform P_A. This contract might contain also information about the conditions in which the access will be granted (several times or unlimited during a time window), optional expiration date, etc. To guarantee its integrity, contracts are cryptographically signed by its issuing platforms (i.e. the contact $C(P_A, P_B)$ is signed by both platform P_A and platform P_B). Upon generation, the contract will be sent as a multi-signature transaction T_C and stored in the Blockchain, who will automatically provide integrity validation. Specifically, T_C contains:

$$T_C = [T_{ID}, D, CB, S, ts]\sigma \tag{1}$$

where T_{ID} is the transaction ID, D corresponds to the smart contract address and it will be empty in order to trigger the procedure for the smart contract creation, CB is the smart contract byte-code, $S = H(PK_X)$ defines the sender address where $H()$ is a generic hashing function, $\sigma = E(H(T_{ID}, D, CB, S, ts), SK_{P_A}, SK_{P_B})$ represents the transaction signatures, where $E()$ is a generic digital signature algorithm, and ts is the timestamp introduced to make the system resistant to replay attacks. Nevertheless, when a contract is called, its status change will be stored in the Blockchain as well. The issuer can then access the status history. Since each change is stored and validated, it can validate the integrity of the operation, checking how many times it has been used, what's the contract's status and usages left or if it has expired. With this solution, when application A_A wants to access to resources available in platform P_B, the following message exchange is implemented (see Figure 1 a):

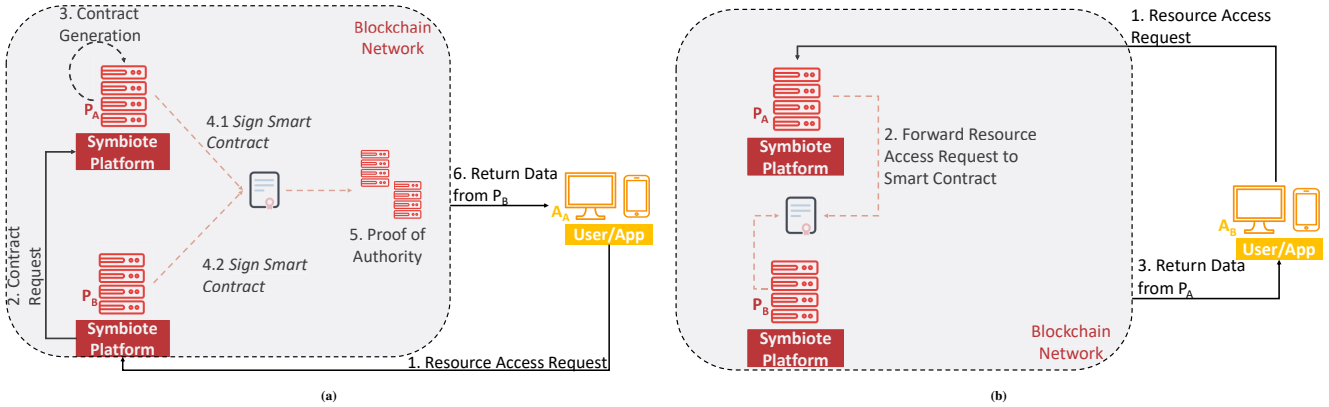

FIGURE 1 Use case: **(a)** Smart Contract definition and Resource Access by A_A in P_B; **(b)** Resource Access by A_B in P_A.

- Application A_A sends an access resource request to the platform P_B through a transaction $T_{AR} = [T_{ID}, H(PK_{P_B}), R, ts]\sigma$, where T_{ID} is the transaction ID, $H(PK_{P_B})$ is the P_B platform address, R is the resource name, ts defines the timestamp and $\sigma = E(H(T_{ID}, H(PK_{P_B}), R, ts), SK_{A_A})$ represents the transaction signature;
- Platform P_B checks if there already exists a contract signed with the platform P_A in the Blockchain. If not, platform P_B sends to platform P_A a transaction contract request $T_{CR} = [T_{ID}, H(PK_{P_A}), C, ts]\sigma$ for a valid contract. Since the initial conditions are the same as in the previous use case, platform P_A does not have a valid contract established with platform P_B so it creates one contract $C(P_A, P_B)$;
- Then, the contract $C(P_A, P_B)$ is sent with a multi-signature transaction T_C by platform P_A towards Blockchain;
- A Blockchain node selected with Proof of Authority will check that this transaction is valid; if the transaction is valid, the smart contract will receive an address D. In the case the procedure ends successfully, a resource of platform P_B is granted to application A_A.

Similarly, when application A_B wants to access to resources available in platform P_A, the following message exchange is implemented (see Figure 1 b):

- Application A_B sends an access resource request to the platform P_A, through the transaction $T_{AR} = [T_{ID}, H(PK_{P_A}), R, ts]\sigma$;
- Since platform P_B has already established a smart contract C with platform P_A from a previous interaction, it will forward the resource access request A_B's application A_B to the smart contract C with a transaction $T_D = [T_{ID}, D, H(PK_{A_A}), R, ts]\sigma$, where firstly we have the smart contract address D, the application address A_A, the resource requested R, the current timestamp ts and finally the platform P_A transaction signature $\sigma = E(H(T_{ID}, D, H(PK_{A_A}), R, ts), SK_{P_A})$;
- In the case the signature is valid, it records in the Blockchain that platform P_B is using this contract $C(P_A, P_B)$. Since the contract states that any application from platform P_B using that contract will get access to resources in platform P_A, the contract is automatically enforced and the access is granted for platform P_B.
- The contract is then fulfilled and resources in the platform P_A will be given to the application A_B.

4 | CONCLUSIONS AND FUTURE WORKS

This letter proposes a novel approach for implementing bartering services envisaged by the European H2020 symbIoTe project by means of the Blockchain technology. Specifically, the conceived approach allows federated platforms to share resources, based on smart contracts. The devised approach permits to achieve transparency of the transactions between the nodes, trustworthiness

of the involved entities, the immutability of the data written on Blockchain, decentralization regarding the consensus mechanism and an high level of security and information integrity in transactions based on cryptographic signing procedures. Future research activities include the investigation of additional use cases and the evaluation of performances through simulation tools. The proposed approach could be implemented by using a well-known blockchain platform (like Ethereum) and by developing new Application Program Interface able to integrate our functionalities in small and large-scale scenarios.

5 | ACKNOWLEDGEMENTS

This work was framed in the context of the project SymbIoTe, which receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 688156.

References

1. Christidis K., Devetsikiotis M.. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access.* 2016;4:2292-2303.
2. Bach L. M., Mihaljevic B., Zagar M.. Comparative analysis of blockchain consensus algorithms. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* 2018;:1545-1550.
3. Zyskind G., Nathan O., Pentland A. '.. Decentralizing Privacy: Using Blockchain to Protect Personal Data. *2015 IEEE Security and Privacy Workshops.* 2015;:180-184.
4. Reyna Ana, MartÃŋn Cristian, Chen Jaime, Soler Enrique, DÃŋaz Manuel. On blockchain and its integration with IoT. Challenges and opportunities. *Future Generation Computer Systems.* 2018;88:173 - 190.
5. Dorri A., Kanhere S. S., Jurdak R., Gauravaram P.. Blockchain for IoT security and privacy: The case study of a smart home. *2017 IEEE International Conference on Pervasive Computing and Communications Workshops.* 2017;:618-623.
6. Zhu X., Badr Y., Pacheco J., Hariri S.. Autonomic Identity Framework for the Internet of Things. *2017 International Conference on Cloud and Autonomic Computing (ICCAC).* 2017;:69-79.
7. Kravitz D. W., Cooper J.. Securing user identity and transactions symbiotically: IoT meets blockchain. *2017 Global Internet of Things Summit (GIoTS).* 2017;:1-6.
8. Khan Minhaj Ahmad, Salah Khaled. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems.* 2018;82:395 - 411.
9. European Union's Horizon 2020. symbIoTe - symbiosis of smart objects across IoT environments, https://www.symbiote-h2020.eu/(accessed July 30, 2018).
10. Banerjee Mandrita, Lee Junghee, Choo Kim-Kwang Raymond. A blockchain future to Internet of Things security: A position paper. *Digital Communications and Networks.* 2017;Open Access.
11. L. Lamport M. Pease. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst..* 1982;4(3):382–401.
12. Zhang Y., Wen J.. An IoT electric business model based on the protocol of bitcoin. *2015 18th International Conference on Intelligence in Next Generation Networks.* 2015;:184-191.
13. Wörner Dominic, Bomhard Thomas. When Your Sensor Earns Money: Exchanging Data for Cash with Bitcoin. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication.* 2014;:295–298.

How to cite this article: Pietro Tedeschi, Giuseppe Piro, Jose Antonio Sanchez Murillo, Nemanja Ignjatov, Michał Pilc, Kaspar Lebloch, and Gennaro Boggia (2018), Blockchain as a Service: securing bartering functionalities in the H2020 symbIoTe framework, *Internet Technology Letters*, *2018;00:1–6*.