(POSTER) SmartTwins: Secure and Auditable DLT-based Digital Twins for the WoT

Iakovos Pittaras and George C. Polyzos
Mobile Multimedia Laboratory
Department of Informatics, School of Information Sciences and Technology
Athens University of Economics and Business, Greece
{pittaras,polyzos}@aueb.gr

***Abstract*—Digital Twins and the Internet of Things (IoT) are two of the most prominent recent concepts and technologies. The IoT supports many applications that merge the physical with the cyber world. This highlights the need for improved security. Here, we argue that digital twins can help in securing and strengthening the IoT by using them for interacting with the actual IoT devices. While most digital twins implementations are centralized, we propose to integrate Distributed Ledger Technologies (DLTs) and digital twins into the IoT to realize decentralized, secure, available, flexible, and auditable blockchain-based IoT services for IoT devices that follow the W3C Web of Things (WoT) standards. In this work, we present the design of SmartTwin, a blockchain-based digital twin framework for which we provide two different implementations using two different blockchains, we present the design trade-offs, and we discuss future research and development directions.**

***Index Terms*—Digital Twins, Distributed Ledger Technologies, Internet of Things, Sensing, Actuation, Blockchain, Ethereum, Hyperledger Fabric, Auditability, Availability**

I. Introduction

Securing IoT services, i.e., actuation and sensing processes, requires complex security operations, such as advanced cryptographic algorithms. However, the existing security operations are not designed for the IoT, in which the IoT devices used are usually resource-constrained. So, even if some IoT devices are capable of executing these operations, more lightweight solutions that take into consideration the limitations of all the IoT devices are required. One solution towards that direction is the use of digital twins. A digital twin is the virtual replica of a physical (IoT) device, system, or asset more generally [1]. It is observed that in IoT, digital twins are mostly used for testing, monitoring, and simulating the IoT devices. However, in our work, we propose the usage of digital twins for interacting with the actual IoT devices. Users instead of interacting with the actual device, they will interact with its digital twin. Then, all valid state modifications of the virtual twin will be securely transmitted to the actual device, which will perform the required operation.

Digital twins are usually stored in a more secure network location, e.g., a Web server, than the actual IoT device, which in many cases is exposed physically to users. However, even in these cases, the digital twin is not completely secured, since it is stored in centralized servers, which constitute single points of failure. We argue that blockchains can be used to create decentralized, secure, and auditable digital twins. Blockchains, and Distributed Ledger Technologies (DLTs) more generally, can be regarded as immutable, distributed append-only ledgers of transactions distributed throughout a network of trustless nodes. A blockchain might be public or private. The most popular representatives of them are Ethereum blockchain [2] and Hyperledger Fabric [3] (for the rest of the paper, we will simply refer to it as Fabric). Each one of them has different advantages and drawbacks, especially when used in the IoT domain [4]. These two blockchains are capable of executing distributed applications, often called smart contracts. Blockchains, in general, have increased availability and auditability, hence they are a promising solution for creating digital twins.

The idea of integrating blockchains and digital twins is not new. There are many research efforts that investigate blockchain-based digital twins [5]–[7]. However, these solutions propose the usage of blockchains as a means for data sharing or data storage, while in our solution, we propose using the digital twin as an intermediary between the users and the IoT devices. As IoT devices, our solution considers devices that use the Web of Things (WoT) standards, developed by the WoT W3C working group [8]. The WoT model uses a common format for describing IoT devices, called Thing Description (TD) [9]. The TD includes metadata about the IoT devices, namely IoT device properties, actions, and events, in a machine readable format. This is what our solution exploits in order to create the digital twins of the IoT devices as smart contracts.

We summarize here our efforts for creating blockchain-based digital twins for WoT-enabled IoT devices. With our solution, we achieve to create secure, decentralized, reliable, and flexible digital twins that offer increased availability and auditability. In addition, with our solution consumers are IoT device/gateway (vendor-)agnostic. They just need to have our client application to interact with any IoT device of any manufacturer. Finally, with the proposed design, the location of the gateway does not have to be known to consumers, since they interact with it through the digital twins, securing the IoT devices/gateways even more. To show its feasibility and demonstrate its advantages in many use cases, we implement our solution in two different blockchains, Ethereum and Fabric. We argue that with these two blockchains, we cover a wide range of the the IoT use cases, those that require high performance or enhanced privacy and those that openness and

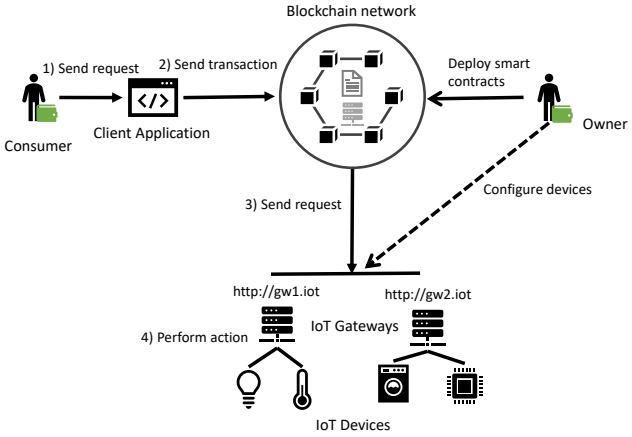

Fig. 1. An overview of the system's architecture.

full auditability is desired. Our solution can easily be adapted in any other blockchain that supports the execution of smart contracts.

II. System Overview

Our system is a typical IoT architecture, with the addition of the blockchain infrastructure. Hence, our system, depicted in Figure 1, is composed of the following entities. *Consumer(s)*, who want to interact with the provided IoT devices, an *owner*, who administrates the IoT devices and the corresponding gateways, *IoT devices* and *gateways*, and lastly, the *blockchain infrastructure*, which depending on the use case might be Ethereum or Fabric, along with the smart contracts that act as the digital twins of the IoT devices/gateways.

Consumers in order to interact with the IoT devices, they have to interact with the smart contracts that are deployed on the blockchain network. To do so, they have to own a blockchain wallet, which includes a public/private key pair used for signing transactions sent to the blockchain network. The IoT gateways, also called *servients* in the WoT standards, are software stacks that implement the WoT-specific functionality of an IoT device. The gateways include ("consume") the TD of an IoT device, or of an IoT "virtual entity", which is the composition of one or more IoT devices, and then they "expose" all the provided operations of the IoT device.

From a high level perspective, the entities in our system interact with each other as follows. Initially, the owner of the IoT devices/gateways has to physically deploy all of them and pairs them with the corresponding gateways. Then, he creates the smart contracts that act as the digital twins of the IoT devices/gateways and deploys them on the blockchain network. Depending on the blockchain used in the system, the design of the digital twins slightly changes (see the next sections). When the setup has been completed, a consumer can gain access to the provided services, hence to the IoT devices. To do so, she has to obtain permission from the owner. Then, the consumer can read the blockchain to learn all the available actions and the required parameters. From this point, a consumer can perform an IoT device access request. She sends a transaction to the smart contract-based digital twin, which includes the desired action and the appropriate parameters. The smart contract verifies the transaction, namely it checks that the requested action exists and the parameters are correct. Then, it forwards the request to the appropriate IoT gateway. Finally, the IoT gateway forwards the request to the IoT devices, which eventually perform the requested operations.

A. Ethereum-based digital twins

Ethereum is capable of executing immutable smart contracts of any complexity. However, Ethereum smart contracts have some limitations. First of all, we cannot access directly something that is off-chain from on-chain smart contracts. Furthermore, all actions that involve the invocation of a function in a smart contract incur a transaction cost, which in Ethereum is expressed as gas. Gas is the Ethereum's unit for measuring the computational and storage overhead of a transaction. In addition to gas, Ethereum introduces transaction delays, which depend on the block mining time. The average time required by an operation to be executed on the Ethereum is around 15 seconds. Finally, Ethereum is a public blockchain, meaning that everyone can read and send transactions to the blockchain. Therefore, when designing the digital twins as smart contracts in Ethereum, we should consider these limitations.

To restrict the access on IoT devices, we propose a form of access control. Consumers can gain access to the smart contract-based digital twin, by obtaining some owner-specific tokens, implemented following the ERC 20 token standard [10]. Thus, only consumers that have obtained these tokens can perform operations on IoT devices. To obtain these tokens, consumers have to communicate with the owner offline and off-chain. Furthermore, to avoid having not negligible costs, we implement in the smart contract a stripped down version of the IoT devices/gateways TD. Each action of the TD, described an IoT device/gateway, is stored in a data structure in the smart contract, called *actionsList*. This structure contains an action name, the input parameters (the type and the number of parameters), and the defined price expressed in ERC 20 tokens. This is shown in Figure 2.

A consumer to perform an operation on the IoT devices, she has to send a transaction on the smart contract-based digital twins (its address is considered known). The transaction should include the required number of tokens, the action name, and the corresponding parameters. The smart contract verifies the transaction, and if it is valid, i.e., it includes the required number of tokens, the action exists, and the parameters are correct, it transfers the tokens to the owner's address. Then, it generates an event that includes the action name and the parameters. The event is eventually caught by the IoT gateway, which "listens" the blockchain network for events. Finally, the IoT gateway forwards the requested action to the appropriate IoT device, which perform the action (actuation or sensing). A cost evaluation, as well as, a security evaluation of the Ethereum-based digital twins is presented in [11].

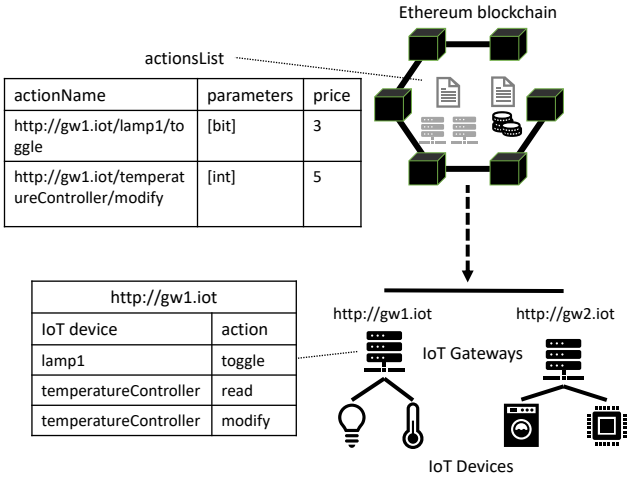

Fig. 2. Digital twin's structure on Ethereum blockchain.

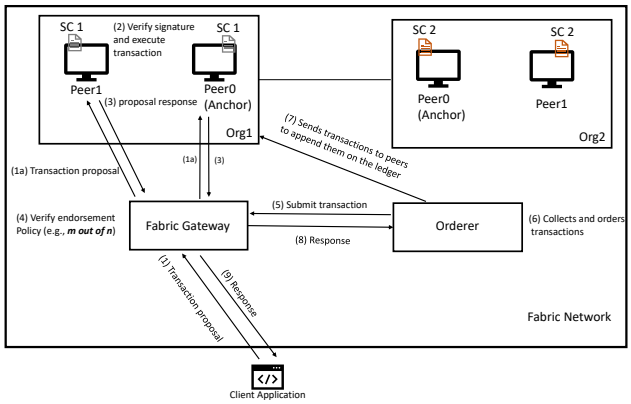

Fig. 3. Transaction flow on Hyperledger Fabric blockchain network.

B. Hyperledger Fabric-based digital twins

As we have already mentioned above, Ethereum has some limitations. So, we have also designed and implemented smart contract-based digital twins in Fabric. Fabric is a permissioned, open-source blockchain, where the membership to the network is controlled by a Membership Service Provider (MSP). It also supports the execution of smart contracts (called chaincodes), written in a general-purpose programming language. Fabric introduces a new model for transactions, called *execute-order-validate* (the flow of a transaction is shown in Figure 3). Fabric categorizes the peers into endorsing peers that execute the transactions and orderer peers that order the transactions before committing them on the ledger.

In this construction, the digital twin smart contracts include the whole TD of the IoT devices/gateways, since no cost is introduced. Furthermore, they include a function that returns the available operations of the given IoT devices, and one function that forwards the request to the appropriate IoT gateway. Thus, in order for a consumer to perform an action, she sends a transaction on the smart contract to learn all the available actions and the corresponding parameters. Then, she sends a transaction that includes the desired action and the parameters. The smart contract verifies the transaction and, if it is valid, it forwards the request directly to the appropriate IoT gateway. Finally, the request ends up on the appropriate IoT device that performs the requested operation.

Therefore, the differences between these two implementations are the following. First, the smart contracts on Fabric include the whole TD of the IoT devices, as opposed to the Ethereum smart contracts. Another difference is that in Fabric, the digital twin forwards the request directly to the IoT gateways, instead of sending it indirectly using event-based communication. Finally, Fabric introduces no cost at all and essentially no transaction delays, making it a much better fit performance and cost-wise. However, it is a permissioned blockchain and it might not be appropriate for use cases, where openness and full transparency is desired.

III. Conclusions

In this work we presented a preliminary system that combines the WoT standards and blockchain technology to create secure, available, and auditable digital twins. We have implemented and verified the feasibility of our solution using two different blockchains, Ethereum and Hyperledger Fabric. Our immediate future plans include further developing the systems in order to fully evaluate them and to provide the framework as usable, open source software.

Acknowledgment

The work reported in this paper has been funded by the Research Center of the Athens University of Economics and Business.

References

[1] B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167 653–167 671, 2019.
[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, 2014.
[3] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *EuroSys '18*, 2018.
[4] I. Pittaras, N. Fotiou, V. A. Siris, and G. C. Polyzos, "Beacons and blockchains in the mobile gaming ecosystem: A feasibility analysis," *Sensors*, vol. 21, no. 3, 2021.
[5] I. Yaqoob, K. Salah, M. Uddin, R. Jayaraman, M. Omar, and M. Imran, "Blockchain for digital twins: Recent advances and future research challenges," *IEEE Network*, vol. 34, no. 5, 2020.
[6] B. Putz, M. Dietz, P. Empl, and G. Pernul, "Ethertwin: Blockchain-based secure digital twin information management," *Information Processing & Management*, vol. 58, no. 1, p. 102425, 2021.
[7] P. Raj, "Chapter thirteen - empowering digital twins with blockchain," in *The Blockchain Technology for Secure and Smart Applications across Industry Verticals*, ser. Advances in Computers. Elsevier, 2021, vol. 121, pp. 267–283.
[8] W3C. (2017) Web of Things. https://www.w3.org/WoT/.
[9] S. Kaebish, T. Kamiya, M. McCool, V. Charpenay, and M. Kovatsch. (2020) Web of Things Thing Description. [Online]. Available: https://www.w3.org/TR/wot-thing-description/
[10] F. Vaogelsteller and V. Buterin. (2015) EIP-20: Token Standard. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20
[11] I. Pittaras, N. Fotiou, C. Karapapas, V. A. Siris, and G. C. Polyzos, "Secure, Mass Web of Things Actuation Using Smart Contracts-Based Digital Twins," in *ISCC '22*, July 2022.