Secure and Efficient Web of Things Digital Twins using Permissioned Blockchains

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*—The growing popularity and features of digital twins have made them key elements in Industry 4.0 and the Internet of Things (IoT). In this paper, we propose the use of digital twins as an indirection mechanism, instead of accessing directly the actual IoT devices, in order to offer secure sensing and actuation in IoT systems through Distributed Ledger Technologies (DLTs), which can enhance the security and transparency of such digital twins. In particular, we leverage advances in Web of Things (WoT) and permissioned blockchains to build flexible and secure digital twins. Specifically, we build smart contract-based digital twins of IoT devices that follow W3C's WoT standards. This architecture offers decentralization, immutability, auditability, and enhanced availability and reliability.**

***Index Terms*—Digital Twins, Internet of Things (IoT), Distributed Ledger Technologiess (DLTs), Hyperledger Fabric, interoperability, decentralization, immutability, auditability, availability, reliability**

I. Introduction

The Internet of Things (IoT) is an ecosystem of connected physical devices, which collect, share, and act on data. The IoT merges the cyber with the real world and improves the quality of our lives by providing a multitude of (real-world) services. However, the IoT departs from the traditional system architectures. First of all, there are many different types of devices and many different manufacturers, who develop and use different and often competing protocols, but also different business entities with their own and often competing goals and priorities. This diversity leads us far from the goal of one IoT. To deal with the aforementioned limitations, we argue that we first can take advantage of the Web of Things (WoT). In particular, the WoT W3C working group [1] develops an interoperable IoT architecture by using well-know Web technologies, such as HTTP(s) and RESTful APIs. The WoT standards come with a lot of benefits and address the problems of fragmentation and lack of interoperability in IoT.

In addition, many IoT devices are less powerful than typical computers, hence they cannot perform complex security operations. Furthermore, in many cases, devices are physically exposed (e.g., to potential attackers), making them even more vulnerable. Thus, a new architectural pattern has been observed, where users do not interact directly with the IoT devices, but with a more powerful gateway. However, in these cases, other problems can arise, since the gateway constitutes a single point of failure [2]. In this paper, to secure the IoT devices, as well as, the corresponding gateways, we propose the communication between the users and the IoT devices/gateways to be mediated by the digital twin of the IoT gateway. Users instead of interacting with the actual device, will be interacting with its digital twin, a virtual representation of the IoT device [3]. All valid state modifications of the virtual twin will be securely transmitted to the actual device, which eventually will perform the requested operations.

The digital twins are usually operating in a more powerful and secure network location than the actual devices, such as a Web server or the Cloud. However, digital twins can also suffer from network outages, if they are implemented in a centralized manner. In this work, we are exploring how Distributed Ledger Technologies (DLTs) can be used to create secure and reliable digital twins. In particular, we use the Hyperledger Fabric blockchain [4] and its support for smart contracts to design and implement smart contract-based digital twins of IoT devices that use the WoT standards. We chose to use a private blockchain rather than a public one, in order to avoid issues regarding the performance of our system, such as increased transaction delays, scalability issues, and high costs. Additionally, permissionless blockchains may be inappropriate for use cases, such as a smart home, due to their public nature.

In this paper, we propose an IoT system that takes advantage of smart contract-based digital twins, for the use case of a smart home. Our solution achieves the following. Initially, our solution enhances security and availability by removing the need for a trusted centralized party, which stores the digital twin, as we implement it as a smart contract. Moreover, it improves interoperability of IoT by adopting the WoT architecture. Finally, our solution makes the users oblivious to IoT devices and device vendor-agnostic, since users interact with the digital twin, not the actual devices. Thus, they do not need to know anything other than the provided actions, about the actual devices. Furthermore, users do not have to deploy different software for different IoT devices, which is a common case.

The remainder of this paper is organized as follows. In Section II, we present background information and technologies used in our work, as well as the related work in the area. In Section III, we introduce our blockchain-based WoT

architecture, its design, and its implementation. In Section IV, we qualitative evaluate our system and we present and discuss its advantages and its drawbacks. Finally, in Section V, we conclude our paper and we discuss some future extensions and improvements.

II. Background and Related Work

A. Web of Things

The Web of Things (WoT) architecture [5] attempts to structure well-known Web protocols and tools for connecting IoT devices to the Web. In the WoT architecture communication model, IoT devices are made available through REST-based APIs, which can be used by *consumers* to access device *properties*, to trigger device *actions*, as well as to receive device-generated *events*. In order to improve the interoperability and usability of IoT platforms, the WoT model uses a common format for describing IoT devices referred to as the *Thing Description (TD)* [6]. The TD is machine-readable and includes metadata about the IoT device (such as its ID, a title, and security definitions), as well as IoT device *properties*, *actions*, and *events* that can be accessed or invoked through Web *links* and Web *forms*.

Listing 1 provides an example of a WoT TD for a smart lamp, which is encoded using JSON-LD. The smart lamp in this example exposes a property named `status`, which returns the current status of the lamp, and an action named `toggle`, which accepts as input a boolean and instructs the lamp to switch on/off. Additionally, this TD includes information about how this IoT device can be accessed (i.e., via an HTTP request to the specified URI). In particular, to switch on the lamp, a POST request should be sent to https://example.com/things/lamp1/toggle. Similarly, in order to read the current status of the lamp, a GET request should be sent to https://example.com/things/lamp1/status.

```
{“@context”: “https://www.w3.org/2019/wot/td/v1”,
 “id”: “lamp1”,
 “title’’: “My lamp’’
 “securityDefinitions’’: {...}
 “security’’: [...]
 “properties”: {
   “status”: {
    “type”: “string”,
    “forms”: [{“href”: “https://example.com/things/
          lamp1/status”}] } },
 “actions”: {
   “toggle”: {
        “type”:“boolean’’
        “forms”: [{“href”: “https://example.com/things
             /lamp1/toggle”}] } },
 “events”: {...} }
```

Listing 1. WoT Thing Description for smart lamp.

B. Distributed Ledger Technologies

A blockchain is an append-only ledger of transactions distributed throughout a network of trustless nodes. Hence, they are also referred to as DLTs. Each block of the blockchain contains a list of validated transactions organized in a Merkle Tree. Transactions are validated by several network nodes and are added in the ledger upon consensus (usually with a Byzantine Fault Tolerant protocol [7]). A blockchain may be public [8] or private [4], even though finer distinctions can be made in some cases. In private, permissioned blockchains, only nodes with the right credentials can join the network, observe the blockchain, and alter its state.

A popular implementation of a private blockchain is Hyperledger Fabric [4] (for simplicity, from now on, we will refer to it as Fabric). Fabric is a private, permissioned, open-source blockchain, where the membership to the network is controlled. Fabric, like other popular blockchains, supports the execution of distributed applications, written in a general-purpose programming language, called smart contracts (or chaincodes). A smart contract is executed simultaneously and in parallel by several special nodes, called endorsing peers. Fabric introduces a new model for transactions, called execute-order-validate, in addition to other blockchains that follow the order-execute model. Thus, in Fabric, the transaction flow is composed of three steps. Initially, the transaction is executed, then it is ordered by the consensus protocol, and finally it is validated against an endorsement policy (e.g., n out of m endorsing peers should execute the smart contract and produce the same output) before committing it to the ledger. The flow of a transaction in Fabric is shown in Fig. 1 on the bottom right corner.

C. Related Work

Many research efforts investigate blockchain-based digital twins. Yaqoob et al. [9] present some potential use cases, architectures, and technologies that can enhance digital twins to be more effective in real-life industrial problems. They propose the integration of a blockchain into digital twins by suggesting that it can be used as storage for digital twin's data. Khan et al. [10] propose a framework, called spiral digital twin framework, which uses a blockchain to securely and reliably store the digital twin data. As the blockchain, they propose the use of a new blockchain, called twinchain, which addresses the issues of transaction delays, which are significant for many public blockchains, such as Ethereum. More recent efforts [11]–[13] are trying to address the problem of sharing digital twin data. Putz et al. [11] propose EtherTwin, a Decentralized Application (DApp) that facilitates digital twin information and data sharing among multiple parties, without the need for trusted third parties. The work presented by Dietz et al. [12] tries to address the same problem. The authors examine how DLTs can be used to provide secure information sharing for data generated by digital twins.

All these efforts highlight the advantages of integrating DLTs and digital twins. However, these works use the blockchain as a means for digital twin data sharing, while we are using the blockchain to implement the actual digital twin (as a Dapp) to secure and ruggedize the physical IoT devices and gateways.

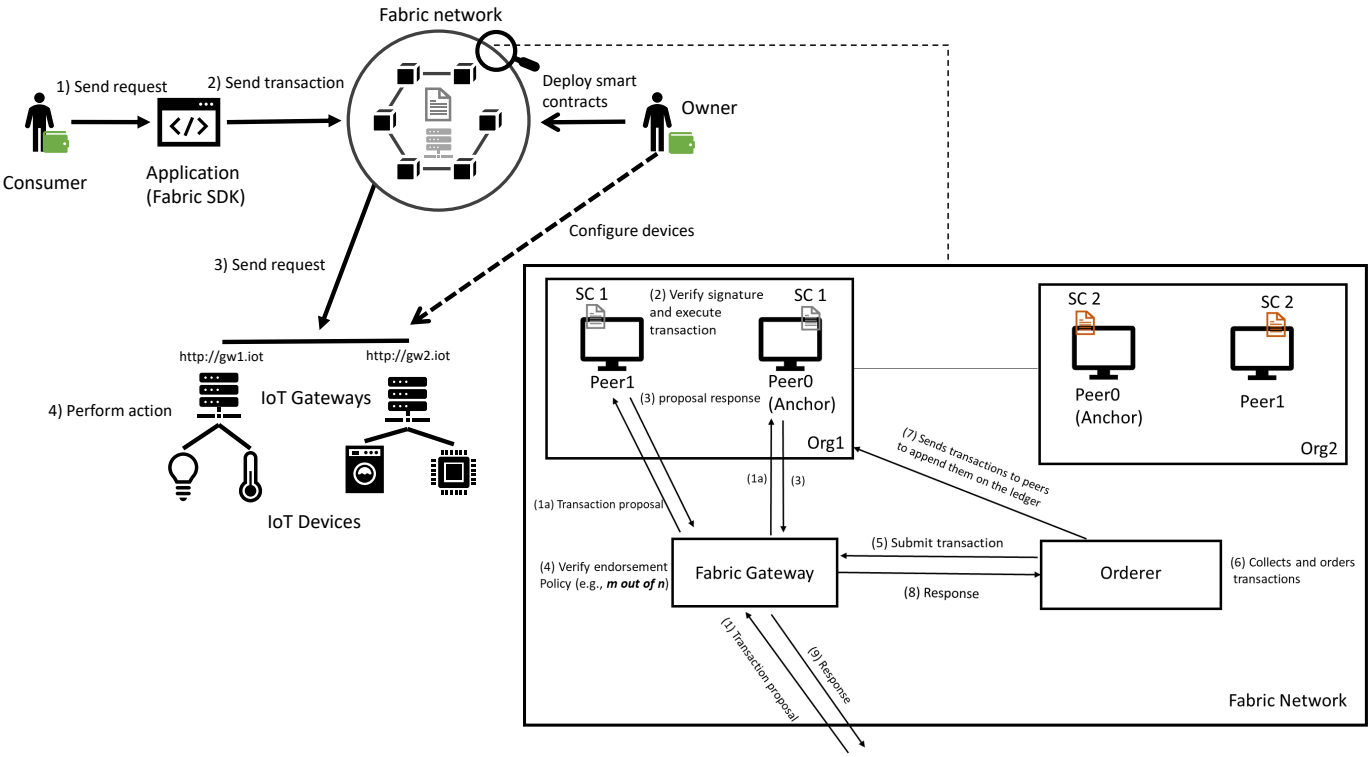

Fig. 1. An overview of the system's architecture.

III. System Overview

We now present the design of the considered architecture, illustrated in Fig. 1. The system is composed of the following entities: the *owner* of the IoT devices/gateways, some *consumers*, who want to interact with the IoT devices, the *Fabric network*, where the digital twins are deployed, and lastly the *IoT devices and the corresponding gateways*. From a high level perspective, these entities interact with each other as follows. Initially, the owner configures and physically deploys the IoT devices/gateways and make them available to the consumers through their digital twins residing on the Fabric network. To do so, he develops the smart contracts that act as the digital twins and deploys them on the Fabric. Then, consumers in order to perform an action (e.g., an actuation process) send a transaction to the smart contract, which forwards the request to the IoT gateways. The gateway will eventually perform the requested action. The whole process is implemented using the following two phases.

A. Setup phase

During the setup phase, the owner has to configure the IoT devices and the IoT gateways, and physically deploy them. The IoT gateways (in the WoT model, gateways are also called *servients*) "consume" the TD of the IoT devices and "expose" their actions and properties. The IoT devices/-gateways are identified by URIs (see Listing 1). Next, the owner has to set up and bring up the Fabric blockchain network. The Fabric network is composed of organizations, which are composed of (endorsing) peers, an orderer, who orders the transactions, and a Certificate Authority (CA), which controls the membership on the network. In our use case, the Fabric network has only one organization. Then, he has to develop and deploy the smart contracts that represent the digital twins that correspond to the physically deployed IoT devices/gateway, on the Fabric network. The owner can create one digital twin per IoT device, or he can create one digital twin per gateway or even one digital twin for all of his gateways, since the WoT standard defines that all gateways together can compose a WoT "virtual entity", which has one TD that includes all the actions, properties, and events of all the IoT devices/gateways. The smart contract includes the TD of the corresponding IoT device(s)/gateway(s), one function that returns the available operations of the IoT devices, and one function that forwards the request to the appropriate gateway. We should note here that the function that returns information about the IoT devices/gateways does not return the URIs of these devices/gateways. So, consumers can interact with them only through their digital twins, as they do not know their endpoint.

In order for a consumer to be able to interact with the digital twins, she has to be a member of the Fabric's network. In Fabric, membership to the network is controlled by a Membership Service Provider (MSP), which is a trusted authority. The MSP implementation in Fabric follows the PKI model. Namely, it uses X.509 certificates issued by Certificate Authorities (CAs). Thus, the consumer to be able to interact with the digital twins, she has to obtain an identity, namely a X.509 certificate. To do so, she has to communicate with the admin of the network,

who in our case is the owner, and the corresponding CA to obtain the certificate.

B. IoT device access phase

After the completion of the setup phase, the consumer is able to request and perform an action on the provided IoT devices. Initially, the consumer sends a transaction on the smart contract-based digital twin to learn all the available operations that are offered. Then, she sends a transaction that includes the action she wants to invoke. The transaction is sent to the appropriate smart contract, which verifies that the transaction is valid. In particular, it checks if the requested action is included in the TD and if it includes the correct parameters. If the transaction is valid, then the smart contract forwards the request to the appropriate gateway. Finally, the request ends up on the appropriate IoT device, which performs the requested action.

C. Implementation

We developed a proof of concept implementation of the presented system. Our IoT gateway is based on Eclipse's Thingweb[1], which is a Node.js implementation of the WoT model. For our proof of concept implementation, we emulated one IoT device, a smart lamp. The smart contract that acts as the digital twin of the smart lamp was implemented using JavaScript[2]. Finally, our client application, which is responsible for acquiring the certificate, and interacting with the smart contract on Fabric through the Fabric gateway, implements the Fabric SDK and it was also implemented using JavaScript.

IV. Discussion

Our proposed system has some intriguing (security-related) properties. With the use of the digital twins, we make consumers oblivious to the actual IoT devices. The consumers do not need to know anything specific about the IoT devices, rather than the actions they provide, which they learn from the digital twin that includes the TD of the IoT devices. To interact with an IoT device, a consumer has to send a transaction to the blockchain instead of communicating directly with the IoT device/gateway, which would require consumers to be aware of the vendor's (or other) specific protocols. Thus, we allow consumers to be IoT device/gateway vendor/protocol-agnostic. With the proposed design, users do not have to deploy and use different client software for their IoT devices, which is the common case. They just need to deploy one client application, which implements the Fabric SDK and interacts with the smart contracts, to interact with any IoT device of any manufacturer. Moreover, new IoT devices can easily be added (or others removed) from the system by just updating the smart contract or deploying a new one, without the need for changing anything to consumer applications. Furthermore, since the consumers interact with the digital twin and not the actual IoT gateway, the location (or address) of the gateway does not have to be known. In that way, we are securing the gateway even more.

In addition, our system inherits all the properties of the blockchain. Blockchains offer reliability, decentralization, and increased availability by design, since there is no single point of failure. Every transaction and its output is immutable recorded on the ledger. This also enhances our system with increased auditability, since every interaction with the IoT devices is recorded on the ledger, and thus we can audit it at any time.

Regarding the blockchain technology, we chose to use Hyperledger Fabric for a variety of reasons. First of all, Fabric can serve many different use cases. In our system, we use it for the use case of a smart home. For this reason, we built it with just one organization. However, if the use case is a big smart business building, where there are many employees working in and for different organizations, then Fabric can be used with two or more organizations. Each organization would have their digital twins, without the other organizations knowing anything about them. In Fabric, we can even have smart contracts within an organization that cannot be accessed by some peers of the same organization, through the use of private channels, a term and technology introduced by Fabric. This is something that cannot be achieved with the use of the Ethereum blockchain, which is public and everything recorded (even the smart contracts) on the ledger can be accessed by anyone. Furthermore, Fabric presents better performance than Ethereum, which introduces at a minimum 15 seconds delay, or other public blockchains, as we have shown in our previous work [14]. In fact, recent findings [15] show that Fabric can scale up to 20000 transactions per second. Ethereum, also introduces some monetary cost, which may not be negligible. Moreover, with Ethereum (monetary) costs, it would be too costly to store the actual TD of an IoT device/gateway on an Ethereum smart contract. Finally, Ethereum smart contracts cannot communicate with the "outside" world (directly), so they cannot forward the request directly to the IoT gateway. Thus, if we were using the Ethereum blockchain, instead of Fabric, a different, more complicated design would have to be adopted.

V. Conclusions and Future Work

In this paper we presented a system that implements digital twins for IoT devices that follow the WoT standard. In particular, we leveraged the Hyperledger Fabric blockchain to build the digital twins of WoT/IoT devices or gateways. Our solution secures the real IoT devices/gateways by allowing consumers to interact with them only through their digital twins. Furthermore, by implementing the digital twins as smart contracts, we achieve decentralization, flexibility, auditability, reliability, and availability.

In our work, we store the TD of the IoT devices in the smart contract. However, an interesting extension to our system would be to fully implement the WoT servients (IoT gateways) as smart contracts. Furthermore, it is in our immediate plans to fully experiment with the presented system in order to evaluate

[1] http://www.thingweb.io/

[2] https://github.com/mmlab-aueb/DLT-DigitalTwins

it, in terms of performance, as well as in terms of security by presenting a threat model.

ACKNOWLEDGMENT

This work has originated with H2020 project SOFIE (Secure Open Federation for Internet Everywhere, Grant # 779984) and has since been supported in part by the Research Center of the Athens University of Economics and Business.

REFERENCES

[1] W3C. (2017) Web of Things. https://www.w3.org/WoT/.

[2] Noor Al-Sibai. (2021) Amazon Outage Shuts Down IoT Vacuums, Doorbells, Fridges, Even Home Locks. [Online]. Available: https://futurism.com/amazon-outage-iot

[3] 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.

[4] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18. Association for Computing Machinery, 2018.

[5] M. Kovatsch and R. Matsukura and M. Lagally and T. Kawaguchi and K. Toumura and K. Kajimoto. (2020) Web of Things Architecture. [Online]. Available: https://www.w3.org/TR/wot-architecture/

[6] S. Kaebish and T. kamiya and M. McCool and V. Charpenay and M. Kovatsch. (2020) Web of Things Thing Description. [Online]. Available: https://www.w3.org/TR/wot-thing-description/

[7] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

[8] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, 2014.

[9] 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.

[10] A. Khan, F. Shahid, C. Maple, A. Ahmad, and G. Jeon, "Toward smart manufacturing using spiral digital twin framework and twinchain," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1359–1366, 2022.

[11] 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.

[12] M. Dietz, B. Putz, and G. Pernul, "A Distributed Ledger Approach to Digital Twin Secure Data Sharing," in *33th IFIP Annual Conference on Data and Applications Security and Privacy (DBSec)*, vol. LNCS-11559, Jul. 2019, pp. 281–300.

[13] W. Shen, T. Hu, C. Zhang, and S. Ma, "Secure sharing of big digital twin data for smart manufacturing based on blockchain," *Journal of Manufacturing Systems*, vol. 61, pp. 338–350, 2021.

[14] 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.

[15] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, "FastFabric: Scaling hyperledger fabric to 20 000 transactions per second," *International Journal of Network Management*, vol. 30, no. 5, September 2020.