Trustworthy IoT: An Evidence Collection Approach based on Smart Contracts

Claudio A. Ardagna
Università degli Studi di Milano
Crema, Italy
claudio.ardagna@unimi.it

Rasool Asal, Ernesto Damiani
EBTIC – Khalifa University
Abu Dhabi, UAE
rasool.asal@bt.com
ernesto.damiani@kustar.ac.ae

Nabil El Ioini, Claus Pahl
Free University of Bozen
Bolzano, Italy
{nelioini,claus.pahl}@unibz.it

***Abstract*—Today, Internet of Things (IoT) implements an ecosystem where a panoply of interconnected devices collect data from physical environments and supply them to processing services, on top of which cloud-based applications are built and provided to mobile end users. The undebatable advantages of smart IoT systems clash with the need of a secure and trustworthy environment. In this paper, we propose a service-based methodology based on blockchain and smart contracts for trustworthy evidence collection at the basis of a trustworthy IoT assurance evaluation. The methodology balances the provided level of trustworthiness and its performance, and is experimentally evaluated using Hyperledger fabric blockchain.**

***Keywords*-Assurance Evaluation, Blockchain, IoT.**

I. Introduction

Internet of Things (IoT) can be defined as "*the networked interconnection of everyday objects, equipped with ubiquitous intelligence*" [1]. The existence of billions of cheap and resource-constrained devices connected to the Internet introduces fundamental risks that can threaten users' life and personal sphere. A wealth of services in different domains, such as smart vehicles, smart buildings, e-health, are distributed on the basis of data collected by devices. In this context, assurance evaluation is fundamental to guarantee the correct behavior of the whole system and its devices [2]. Here, we use the term assurance in a wider sense to mean the technical judgment that a service, process, or device satisfies some properties. The implicit assumption is that data have a sufficient level of trustworthiness to create information, and in turn knowledge and wisdom [3]. This assumption is however not sound when a plethora of devices are used to collect data, and might bring to scenarios where wrong evidence results in wrong decisions and, in turn, untrusted services/applications. We argue that without an open, protocol-neutral baseline solution for IoT assurance, fundamental risks will become even worse.

Research on IoT-based systems assurance is however at an early stage, and mainly focused on defining new assurance architectures for IoT. Ardagna et al. [4] first discussed challenges in the design and development of assurance techniques for IoT, proposing a conceptual framework and architecture for IoT security assurance evaluation. Sato et al. [5] investigated the problem of trust establishment in IoT and proposed an architecture for evaluating "*area-wise trust*", where the trust level considers device identification, monitoring of device behaviors, device connection processes and protocols. Traditional assurance techniques [2] are however affected by important limitations when targeting complex IoT systems as follows.

- *Hybrid systems.* Assurance techniques do not target hybrid systems, where cloud systems at the center are connected via edge networks to smart devices at the periphery and no clear perimeters exist.
- *Untrusted (micro) providers.* Hybrid systems rely on data continuously collected by a multitude of devices, which are intrinsically unreliable and under the control of many untrusted providers.
- *Trustworthy evidence.* Traditional assurance is often driven by untrusted/unverified evidence that is accepted on the basis of the provider reputation.

The need of collecting trustworthy evidence clearly emerges in the above challenges. This need has initially dealt with in the context of forensics science by defining a systematic and reliable methodology for evidence collection and analysis [6]. Some solutions based on blockchain have been also proposed to guarantee availability, integrity, and verifiability of collected evidence [7], [8], [9].

In this paper, we fill in the above gaps by providing a novel, service-based methodology for trustworthy evidence collection at the basis of a trustworthy assurance evaluation of IoT processes and systems. Our methodology is provided at different granularities, depending on specific performance requirements, from simple trustworthy evidence collection, to trustworthy evidence aggregation, and provable evidence-based automation. It is based on blockchain and smart contract to guarantee collection of reliable evidence whose integrity is proven over time. Differently from existing solutions, our methodology links the evidence to the way in which it is collected and aggregated, or a decision based on it is taken.

The paper is organized as follows. Section II presents our reference scenario. Section III describes the architecture of our trustworthy evidence collection approach discussed in

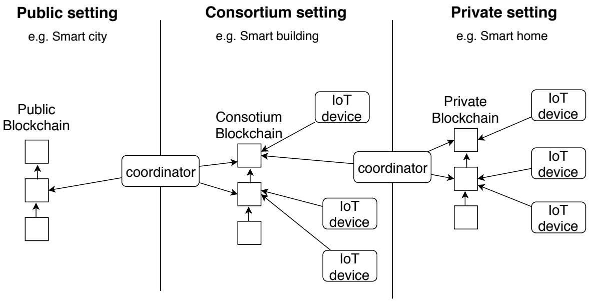


Figure 1. Evidence collection architecture

Section IV. Section V experimentally evaluates the performance of our approach. Section VI gives our conclusions.

II. Reference Scenario

Our reference scenario considers *smart multi-family residential buildings*, composed of several *smart homes*, where enhanced lighting, energy, heating, air conditioning, and physical security systems are integrated together to increase users' experience (e.g., security, comfort, convenience). Sensors are deployed at different system layers to connect people with technology, and collect precise and accurate data on the environment status; actuators and controllers use these data to manage and adapt the building according to predefined rules and policies. Assurance as a way to increase users' confidence that the smart buildings behave as expected is a fundamental requirement. Our reference scenario requires accurate and trustworthy evidence collected by local (wearable) devices that hold properties of integrity, traceability, verifiability, and privacy, to name but a few. Trustworthy evidence on local devices can then be used to produce local claims on the status of a given object/subsystem under evaluation. Such claims are at the basis of a process-wide assurance verification, where local claims are composed in process-wide global claims to verify the assurance of composite IoT-enabled processes.

III. Evidence Collection Architecture

An evidence is a piece of information at the basis of any assurance process, which can be used to trigger an action (e.g., fire alarm) or prove a state (e.g., temperature value). Evidence can come from different sources (e.g., sensors, edge nodes, users), have different granularities (e.g., single sensor reading, aggregated data, decision), and be of three types depending on where sources are deployed (e.g., private, community, public evidence).

Figure 1 depicts the high-level design of our evidence collection architecture, which includes the main services and their relationships. The architecture is mapped on three layers namely *private*, *consortium* and *public*. The proposed architecture isolates the content of each layer from the rest of the network. The coordinators are the only services that can fully access two layers at a time. Their task is to act as intermediaries between two layers, performing additional operations on the collected evidence (e.g., filtering, aggregation) before passing it to the next layer. In the following, we discuss the three layers.

1) Private layer: It is composed of private smart devices that collect atomic evidence related to private settings (e.g., a smart home) and a coordinator node that have access to such evidence. The collected evidence refers to a single unit (e.g., house, machine). Smart devices are part of a private blockchain network where they store their readings. Each device digitally signs its readings before sending them to the private blockchain.

2) Consortium layer: It is composed of *i)* one or more coordinators between the private and consortium layers, *ii)* one coordinator between the consortium and public layers, *iii)* consortium smart devices, and *iv)* a consortium blockchain. Services at consortium layer aggregate evidence coming from the private layer with additional evidence (e.g., coming from elevators, building lighting) collected at the consortium layer based on specific criteria (e.g., location, organizational boundaries). Similarly to the private layer, the consortium layer makes use of smart contracts to interact with the blockchain. The main difference between the private and consortium layers is the privacy level and the data granularity; while the private layer keeps detailed records of every device, they are protected in a private setting and only their aggregation (unless stated otherwise) is sent up to the consortium layer.

3) Public layer: It is composed of a public blockchain, mediated by a coordinator node, which shares the relevant information. The main purpose of the public layer is to act as an interface with the external world. There are many scenarios where some of the evidence collected internally might be useful or even critical to put on a public ledger. For instance, in a smart city, any power surplus generated in a local environment might be submitted for sale on a public blockchain.

IV. Trustworthy Evidence Collection and Evaluation

We propose an atomistic approach based on trustworthy evidence collection as the basis for implementing a trustworthy IoT environment, where trustworthy processes and decisions are employed. Data collected from each smart device must be first evaluated and then put into service only if a minimum amount of assurance requirements are addressed. To this aim, we use blockchain as the data repository that contains all transactions for trustworthy evidence collection and evaluation. The state of a blockchain is represented by a $k{-}v$ data store $BS{:}k{\rightarrow}v$, where k is a 35-byte key and v is an arbitrary sequence of data. In our context, v is the evidence collected by smart devices at different levels of abstraction.

The blockchain handles evidence validation and storage at various degrees using dedicated smart contracts. A smart contract is composed of a set of data structures that represent the evidence level, a set of functions that act on the evidence and define assurance requirements, and a set of emitted events. Formally, a smart contract is defined as follows.

Definition IV.1 (Smart contract SC)**.** A smart contract SC is defined as a 3-tuple $(\mathcal{D}, \mathcal{F}, \mathcal{E})$ where:

- $\mathcal{D}=\{D_p, D_g, D_d\}$ models the three contract data structures, namely, data point, aggregated evidence, decision evidence (Section IV-A);[1]
- $\mathcal{F}=\{F_1, \ldots, F_n\}$ represents the function calls acting on collected evidence (Section IV-B);
- $\mathcal{E}=\{E_1, \ldots, E_m\}$ models events emitted during contract execution.

In the following, after presenting the three different data structures, we discuss four function calls modeling different collection processes that balance evidence trustworthiness and system performance.

A. Data structures

A data point is the least amount of evidence and is defined as follows.

Definition IV.2 (Data point dp)**.** Data point dp is a 5-tuple dp=(s, n, sf, v, $\mathcal{T}$), where s is the source submitting the data point, n is the name of the data point, sf is the store function (either ES in Definition IV.5 or EC in Definition IV.6), v is the value of the data point, and $\mathcal{T}$ is the time of the data point recording. Value v can store: *i)* the specific sensor reading sr or *ii)* an error in the form "*err code–expected value–current value*" according to sf.

A set of data points can be aggregated to provide more information (i.e., aggregated evidence) as follows.

Definition IV.3 (Aggregated evidence $\mathcal{A}$)**.** Aggregated evidence $\mathcal{A}$ is a 5-tuple $\mathcal{A}$=($\mathcal{I}$, n, DA, v, $\mathcal{T}$), where $\mathcal{I}$ is the aggregation time interval, n is the name of the aggregated evidence, DA is the function data aggregation (Definition IV.7), v is the aggregated value, and $\mathcal{T}$ is the time of the evidence aggregation.

A set of data points, aggregated evidence, or a decision evidence itself can be used to produce a decision evidence, which can be enforced for process automation.

Definition IV.4 (Decision evidence $\mathcal{P}$)**.** Decision evidence $\mathcal{P}$ is a 5-tuple $\mathcal{P}$= ($\mathcal{I}$, n, TD, v, $\mathcal{T}$), where $\mathcal{I}$ is the time interval considered to generate the evidence, n is the name of the decision evidence, TD is the decision function (Definition IV.9), v is the decision value, and $\mathcal{T}$ is the time of the decision evidence.

We note that the decision function defines what action to take based on the input data.

B. Function calls

The adoption of a solution based on blockchain can however backfire: management of a blockchain is costly, while a trustworthy evidence collection and evaluation process is resource demanding. We then need to balance the level of trustworthiness we want to achieve and the performance of evidence collection/evaluation. To the aim of balancing trust and performance, we define different function calls supporting different levels of trustworthiness, which differs on the amount of computations done on chain.

1) Trustworthy evidence store: Trustworthy evidence store uses the blockchain to simply store observables collected by smart devices.

Definition IV.5 (ES)**.** Function Evidence Store ES:$sr \rightarrow dp$ takes as input a sensor reading sr and produces as output a data point dp that is stored in blockchain BS.

Example IV.1. In a smart home, sensor data can be stored following a predefined time interval or can be triggered by specific events. For instance, an energy consumption sensor can periodically send power consumption values to the blockchain, including the sensor id, the data key, and the data value (e.g., $\{Sensor123, power\_consumption, 10w/h\}$)

2) Trustworthy evidence collection: Trustworthy evidence collection extends trustworthy evidence store (Definition IV.5) by defining a function that validates the collected evidence against assurance requirements before adding it to the blockchain.

Definition IV.6 (EC)**.** Function Evidence Collection EC: $sr \times req \rightarrow dp$ takes as input a sensor reading sr and a Boolean expression of assurance requirements req, and produces as output a data point dp that is stored in the blockchain BS.

We note that an assurance requirement is a term of the form (a op v), with a an attribute, $op \in \{=, \neq, <, \leq, >, \geq\}$ a comparison operator, and v the expected value/threshold for the specific sensor reading sr. For instance, assurance requirement ($power\_consumption \leq 10w/h$) restricts the domain of valid sensor readings; assurance requirements ($timestamp - current\_datetime \leq 1hour$) and ($pending\_updates = null$) restrict the domain of valid sensor readings to those coming from updated devices and measured in the last hour. Evidence validation is done on chain and only those data readings that are successfully validated against the specified requirements are stored as valid data points in the blockchain; otherwise, an error data point with prefix "*err*" is stored, specifying the reason why the assurance requirement has not been met. This provides a higher level of trustworthiness, since it permits to track

[1] When clear from the context, we will use D_p, D_g, and D_d to denote a set of data points, aggregated evidence, and decision evidence, respectively.

all data points that have satisfied or not the assurance requirements.

Example IV.2. Following example IV.1, when the data coming from the sensor are submitted to the smart contract, assurance requirements are checked before storing the transaction. For instance, assurance requirement ($power\_consumption \leq 10w/h$) is checked every time the evidence collection function is invoked.

3) Trustworthy data aggregation: Trustworthy data aggregation supports the calculation of specific metrics directly in the blockchain. It defines a function called *data aggregation* that extends the evidence collection function (Definition IV.6) with trustworthy evidence aggregation. This approach substantially increases privacy, since it permits to hide details about the single data points, while keeping the possibility to track back data to its origin.

Definition IV.7 (DA)**.** Function data aggregation DA: $D_p \times op \rightarrow \mathcal{A}$ takes as input a set of data points D_p=$\{dp_1, dp_2, \ldots, dp_n\}$, the aggregation operator op∈$\{sum, average, min, max, count\}$, and produces as output the aggregated evidence $\mathcal{A}$=$op(D_p)$ that is stored in blockchain BS.

Example IV.3. The building power consumption is an aggregation (e.g., average, sum) of the power consumption of all smart homes plus any power consumed by the shared services (e.g., elevator, stairs lighting). To maintain a high level of privacy, while supporting full traceability of data, function data aggregation uses the data recorded by the single sensors but only exposes an aggregated version of the data.

4) Trustworthy decision: Trustworthy decision provides the highest level of trustworthiness, at high costs. The whole process from data point collection to decision making is subject to a smart contract and executed on chain. The decision making process is recorded into the blockchain to increase transparency and traceability, following a *decision smart contract* that triggers a specific decision when collected evidence satisfies a (set of) condition. We first define a decision function as follows.

Definition IV.8 (df)**.** Function df is a membership function that takes as input a set $\mathcal{D}$ of evidence and returns as output a decision, according to a set of conditions in the form (*attr op value*), with $op \in \{=, \neq, <, \leq, >, \geq\}$. It assumes the following form:

$$df(\mathcal{D}) = \begin{cases} decision1 & if \quad condition1 \\ decision2 & if \quad condition2 \\ decision3 & if \quad condition3 \end{cases}$$

The trustworthy decision function takes as input a set of evidence with the corresponding decision function, and provides as output the decision evidence.

Definition IV.9 (TD)**.** Function Trustworthy Decision TD: $\mathcal{D} \times df \rightarrow \mathcal{P}$ takes as input a set $\mathcal{D}$ of evidence (see Definition IV.1), a decision function df, and produces as output the decision evidence $\mathcal{P}$=$df(\mathcal{D})$ that is stored in blockchain BS.

Example IV.4. A decision function can reduce the house lighting when the power consumption level reaches a certain threshold. The sensor reading (i.e., current power consumption), the condition, and the decision (i.e., reduce lighting) are all stored on chain.

V. Experimental Evaluation

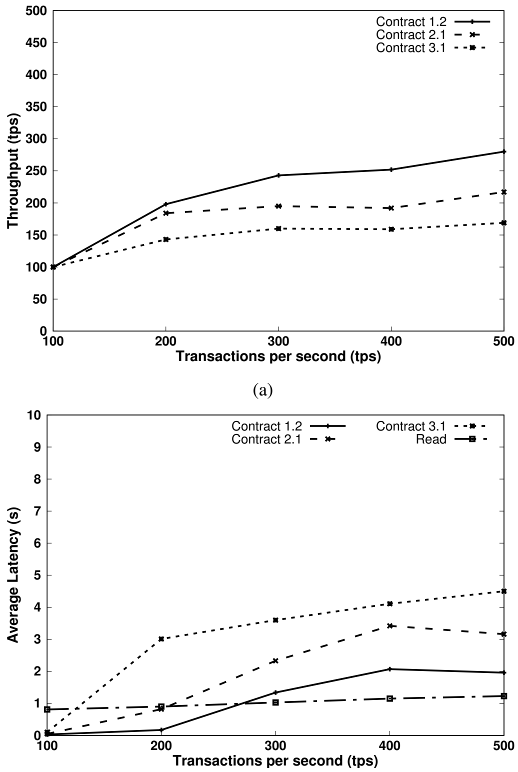
We experimentally evaluated the performance of our evidence collection approach using a virtual machine with 20 vCPUs Intel Xeon E5-2640 v4 @ 2.40GHz and 64GB RAM and Hyperledger fabric blockchain as a permissioned blockchain environment [10].

We configured a test network composed of 4 organizations, each one including two endorsing peers (they host ledgers and smart contracts) and one orderer node used to generate the new blocks of evidence in the blockchain. Each pair of organizations share one channel with the total of 3 channels, corresponding to the public, consortium, and private blockchains in Figure 1. Each channel specified relevant smart contracts (chaincode), mapping the different function calls in Section IV-B. Contract 1.1 refers to function Evidence Store ES in Definition IV.5; Contract 1.2 refers to function Evidence Collection EC in Definition IV.5; Contract 2.1 refers to function Data Aggregation DA in Definition IV.7; Contract 3.1 refers to function Trustworthy Decision TD in Definition IV.9.

We evaluated the performance of our network measuring the throughput and average latency retrieved by executing portions of our methodology (i.e., evidence validation, aggregation, decision) on chain. All the tests have been done varying the transactions per second (smart contract execution requests) in tps=100, 200, 300, 400, 500. This configuration permitted to evaluate the scalability of the network modeling an increasing number of systems interacting with it. The results discussed in this section represent the average over 5 executions of the experiments.

First, we compared the performance of Contracts 1.1 and 1.2 using transactions that trigger write operations on the blockchain. Our results show that Contract 1.2, requiring data point validation, decreases the throughput and increases the average latency with respect to Contract 1.1. The throughput is the same for tps=100, while it is decreased of 1.6% for tps=200, 2.9% for tps=300, 3.0% for tps=400, and 3.1% for tps=500. The average latency is increased in all scenarios: 11% for tps=100, 20% for tps=200, 16.6% for tps=300, 9.6% for tps=400, 12.2% for tps=500.

Figure 2 presents the transaction latency of our approach for Contracts 1.2, 2.1, 3.1. It also presents the read latency



(a)

(b)

Figure 2. Performance evaluation: (a) throughput, (b) average latency

when querying the fabric network. In both cases, we varied the number of tps to evaluate how the latency changes. The transaction latency takes in consideration the time from when a transaction is submitted till it is available in the network. The read latency instead is calculated from when a request is submitted till a reply is received [11]. We note that, independently from the selected contract, when we increase tps, a decrease of the throughput (Figure 2(a)) and an increase of the average latency (Figure 2(b)) are observed. Concerning the performance difference between different contracts, let us consider the worst scenario of tps=500 with Contract 1.2 as our baseline. With respect to the baseline: *i)* the aggregation of the evidence on chain (Contract 2.1) decreases the throughput of 22.5% and increases the average latency of 38%; *ii)* the decision on chain (Contract 2.2) decreases the throughput of 39.6% and increases the average latency of 56.4%.

To conclude, our results show that the selected level of trustworthiness (contract) introduces a not-negligible cost in terms of performance, while it has less impact on resource consumption. This suggests the importance of selecting the proper level of trustworthiness for the domain of interest to keep the overall performance under control.

VI. Conclusions

We presented a trustworthy evidence collection methodology based on blockchain and smart contracts supporting a sound IoT assurance evaluation. Trustworthy evidence collection is provided at different granularities balancing the level of trustworthiness and performance of its execution. Evidence collection ranges from simple trustworthy evidence collection, to trustworthy evidence aggregation, and provable evidence-based automation.

Acknowledgment

Research supported, in parts, by EC H2020 Project CONCORDIA GA 830927 and Università degli Studi di Milano under the program "Piano sostegno alla ricerca 2018".

References

[1] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of things," *International Journal of Communication Systems*, vol. 25, pp. 1101–1102, September 2012.

[2] C. Ardagna, R. Asal, E. Damiani, and Q. Vu, "From security to assurance in the cloud: A survey," *ACM CSUR*, vol. 48, no. 1, pp. 2:1–2:50, August 2015.

[3] J. Rowley, "The wisdom hierarchy: representations of the dikw hierarchy," *Journal of Information Science*, vol. 33, no. 2, pp. 163–180, 2007.

[4] C. Ardagna, E. Damiani, J. Schutte, and P. Stephanow, "A case for IoT security assurance," in *Internet of Everything*, B. D. Martino, K. C. Li, L. Yang, and A. Esposito, Eds. Springer, 2017.

[5] H. Sato, A. Kanai, S. Tanimoto, and T. Kobayashi, "Establishing trust in the emerging era of iot," in *Proc. of IEEE SOSE 2016*, Oxford, UK, March–April 2016.

[6] M. Irfan, H. Abbas, Y. Sun, A. Sajid, and M. Pasha, "A framework for cloud forensics evidence collection and analysis using security information and event management," *Security and Communication Networks*, vol. 9, no. 16, pp. 3790–3807, 2016.

[7] S. Bonomi, M. Casini, and C. Ciccotelli, "B-coc: A blockchain-based chain of custody for evidences management in digital forensics," *arXiv preprint arXiv:1807.10359*, 2018.

[8] J. H. Park, J. Y. Park, and E. N. Huh, "Block chain based data logging and integrity management system for cloud forensics," *Computer Science & Information Technology*, pp. 149–159, 2017.

[9] A. H. Lone and R. N. Mir, "Forensic-chain: Blockchain based digital forensics chain of custody with poc in hyperledger composer," *Digital Investigation*, 2019.

[10] N. El Ioini and C. Pahl, "Trustworthy orchestration of container based edge computing using permissioned blockchain," in *Proc. of IoTSMS 2018*, Valencia, Spain, October 2018.

[11] P. Thakkar, S. Nathan, and B. Vishwanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," *arXiv preprint arXiv:1805.11390*, 2018.