systems

MDPI

Article

Characterization and Costs of Integrating Blockchain and IoT for Agri-Food Traceability Systems

Miguel Pincheira, Massimo Vecchio * and Raffaele Giaffreda

Fondazione Bruno Kessler, 38123 Trento, Italy; mpincheiracaro@fbk.eu (M.P.); rgiaffreda@fbk.eu (R.G.)
* Correspondence: mvecchio@fbk.eu

Abstract: An increasing amount of research focuses on integrating the Internet of Things and blockchain technology to address the requirements of traceability applications for Industry 4.0. However, there has been little quantitative analysis of several aspects of these new blockchain-based traceability systems. For instance, very few works have studied blockchain's impact on the resources of constrained IoT sensors. Similarly, the infrastructure costs of these blockchain-based systems are not widely understood. This paper characterizes the resources of low-cost IoT sensors and provides a monetary cost model for blockchain infrastructure to support blockchain-based traceability systems. First, we describe and implement a farm-to-fork case study using public and private blockchain networks. Then, we analyze the impact of blockchain in six different resource-limited IoT devices in terms of disk and memory footprint, processing time, and energy consumption. Next, we present an infrastructure cost model and use it to identify the costs for the public and private networks. Finally, we evaluate the traceability of a product in different scenarios. Our results showed that low-cost sensors could directly interact with both types of blockchains with minimal energy overhead. Furthermore, our cost model showed that setting a private blockchain infrastructure costs approximately the same as that managing 50 products on a public blockchain network.

Keywords: blockchain; Internet of Things; smart contracts; traceability; supply chain; costs

Citation: Pincheira, M.; Vecchio, M.; Giaffreda, R. Characterization and Costs of Integrating Blockchain and IoT for Agri-Food Traceability Systems. *Systems* **2022**, *10*, 57. https://doi.org/10.3390/systems10030057

Academic Editors: Fernando De la Prieta, Sara Rodriguez, Juan M. Corchado and Vicent Botti

Received: 15 March 2022
Accepted: 22 April 2022
Published: 25 April 2022

1. Introduction

The integration of Internet of Things (IoT) devices into blockchain-based systems is enabling new types of decentralized applications across several domains [1]. Small IoT sensors can autonomously produce and consume information about almost any process at a reduced cost and with high granularity [2]. This information later benefits from blockchain technology and its decentralized, trustless environment that enables new types of more transparent cross-organization business processes [3]. Applications for manufacturing [4], water management [5], communications [6], remote sensing [7], and mining inspections [8] are just a few examples of systems adopting the integration of blockchain and IoT.

Following this adoption trend, the agricultural domain is spawning research and development in traceability systems using IoT and blockchain. In the era of Industry 4.0, traceability systems require the precise identification of the different events (e.g., harvest, transportation, transfer of ownership) and stakeholders (e.g., producers, consumers, governments, authorities) involved in the supply chain process of agri-food products [9]. Furthermore, consumers are increasingly demanding farm-to-fork visibility, i.e., a detailed history of where their food comes from [10]. This critical attention translates into an increasing number of surveys highlighting blockchain and IoT as the main enablers of new traceability systems to cope with poorly digitalized procedures and the lack of efficient methods to collect and share information [9,11,12].

However, despite this growing interest, several challenges are associated with adopting these emerging technologies in traceability systems that are not widely understood. On the one hand, the monetary value of implementing such technological solutions is a

topic of great debate for researchers and practitioners [11]. Even if previous studies have discussed factors such as scalability and performance [13], the infrastructure costs of these systems need a more in-depth discussion. Reducing the uncertainty about these costs is a fundamental aspect of the success of the traceability system [11]. On the other hand, agriculture is one of the most restricting IoT domains where applications require power efficiency and low-cost devices to guarantee large-scale, cost-effective, and long-term installations [14]. These sensing devices are the core of the IoT system and set the foundation for blockchain-based applications. Thus, given the plethora of IoT hardware platforms, it is important to understand blockchain's impact on these constrained IoT devices to carefully design the most suitable and cost-effective IoT system under the blockchain-based traceability application.

In this paper, we provide a monetary cost model and the resource characterization for a blockchain-based traceability system that uses constrained IoT devices, extending our previous work detailed in [15]. First, we assess the blockchain impact on sensing devices by evaluating six IoT low-cost boards with restricted resources and different hardware architectures. Next, we discuss processing times, storage and memory usage, and energy consumption using two different blockchain implementations (i.e., Ethereum and Sawtooth) for the selected boards. These are critical constraints for the sensors in a traceability system, where cost-effectiveness is one of the main drivers for the adoption of traceability systems [11].

Then, we present an infrastructure cost model for both public and private blockchain networks. This model helps to decrease the uncertainty of the system regarding the expected benefits and costs [11,16]. Finally, we use our cost model to characterize the traceability system by defining a few simple parameters, which later allow estimating the cost of the application on both implementations.

Thus, the contribution of this paper is two-fold: (i) We characterize the resources and role of constrained IoT devices as the root of information for a blockchain-based application, and (ii) we provide a model for evaluating the infrastructure costs of the blockchain-based application on public and private networks using simple parameters that characterize the traceability application. We expect our device and costs evaluation and discussion to provide a practical reference that further extends current knowledge of blockchain-based traceability systems in public and private networks, particularly when using constrained sensing IoT devices. Moreover, these contributions help overcome the accessibility to blockchain technology, focusing on technical challenges and design decisions of the traceability system, which are some of the key boundaries in successfully adopting such emerging technologies for traceability applications [12,17].

The rest of this paper is structured as follows: Section 2 summarizes related works; Section 3 presents a case study of event-based traceability applying constrained IoT sensors in a blockchain-based system; Section 4 describes an infrastructure model for the proposed architecture; Section 5 provides a thorough evaluation of the presented blockchain-based architecture, characterizing the constrained devices and the cost of the infrastructure to support the traceability system; Section 7 summarizes and discuss our results and proposes potential future works.

2. Related Works

Surveys such as those conducted by Demestichas et al. [18], Corallo et al. [11], Kayikci et al. [9], Feng et al. [19], and [20] show an increasing amount of literature on the application of blockchain for agriculture traceability systems. However, few works have focused on IoT sensors and blockchain impact on these devices.

One of the first works to describe a traceability system combining blockchain and IoT was presented by Feng Tian [21]. His work proposes a combination of RFID and blockchain to enhance food safety in the Chinese agri-food market. However, his work focuses on analyzing the advantages and challenges of such systems compared to centralized systems from a social, financial, and technical perspective. The author extended his work in [22] to

focus on food safety, based on hazard analysis and critical control points (HACCP). Later on, authors of [23] analyze the integration of blockchain and IoT with existing ERP systems to create a smart agriculture ecosystem, focusing on the components and their interactions to address food safety issues.

With a more technical approach, authors of [24] described a farm-to-fork food traceability system that uses the GS1 Business Message Standard, IoT, and the Ethereum blockchain network. Similarly, authors of [10] described and implemented a blockchain-based supply chain platform that relies on IoT devices to collect information about the process. Besides describing the benefits of their system, they evaluated the performance of their proposal on two different blockchain networks, namely, Ethereum and Hyperledger Sawtooth. Likewise, authors of [25] present a blockchain-based system for the traceability of soybean. The work provides a detailed implementation description, focusing on smart contracts using the Ethereum network. Together, these studies provide important insights into the benefits of using IoT and blockchain for traceability applications. However, there has been little discussion on a blockchain impact on sensing IoT devices. Normally, these devices are the most constrained components of the entire IoT system in terms of energy and computing capabilities (i.e., processing, memory, program space), conflicting with the requirements of a blockchain-based architecture [2]. If an IoT sensing device becomes a direct actor on the blockchain system, it enables a root of trust for the sensed data, one step closer to trustworthy oracles [26], truly benefiting from blockchain properties.

Similarly, the infrastructure costs of a blockchain-based IoT traceability system have not been dealt with in-depth. The authors of [27] presented one of the first cost analyses of blockchain-based applications. Authors of [3] compared the costs of a cloud-based application against a blockchain-based one, focusing on business process execution. In the same direction, authors of [28] described a financial evaluation framework for blockchain implementations. They focus on identifying the factors influencing a blockchain-based application (i.e., cost savings, benefits). Finally, authors have adopted a more application-oriented perspective in [29]. Their work focuses on the smart contracts in the Ethereum network, presenting several metrics regarding the application. However, the authors focused only on the *gas* usage for deploying and executing smart contracts regarding the costs of the application.

Collectively, these studies have focused on application factors such as scalability, security, and performance [13]. In contrast, very little is known about the infrastructure costs of these blockchain-based applications. Reducing uncertainty about these costs and, consequently, the benefits of a blockchain-based traceability system is a fundamental factor for the success of the solution [11].

The evidence in this section suggests that, despite the increasing interest in blockchain-based IoT traceability systems, several characteristics of those applications remain unclear. On the one hand, there has been little discussion about blockchain impact on constrained IoT devices. On the other hand, far too little attention has been paid to the costs of these blockchain-based IoT applications. Thus, there is a need for studies characterizing blockchain systems, from the sensing devices to the cost of the infrastructure to support such systems.

3. Case Study: Extra Virgin Olive Oil Traceability

To properly understand and characterize the IoT devices and the costs of the traceability application, we study an event-based "farm-to-fork" scenario. Based on agri-food traceability models [30–32], we simplified the process into four stages: farming, manufacturing, transportation, and market, as shown in Figure 1. This generalized process aligns with traceability systems such as those proposed for products such as soybean [25], fresh fruits and vegetables [21], rice [33], and crops [34], to name a few.

As the agri-food product moves through each process stage, blockchain can be used to record the transfer of ownership between the stakeholders. Additionally, IoT sensors autonomously produce and register detailed information about several parameters of the

underlying process. However, the amount of data produced by these sensors can easily overflow a blockchain network. Thus, we focus on an event-based traceability application, i.e., storing a few events of high importance for each stage in the blockchain while leaving a detailed log on other components of the IoT system, such as a cloud platform or a distributed storage.

Let us consider extra virgin olive oil [30] as an agri-food product for more details of the traceability system. At the farming stage, IoT devices can monitor several parameters that might affect the quality of the olives, including air pollution, temperature, humidity, and soil conditions. These measures can be as granular as each tree on the farm. However, the event-based traceability will store only the totals (i.e., the monthly average for all trees on a sector) for a given batch. Similarly, IoT sensors can provide detailed information about production processing at the manufacturing stage but store only the totals for a single oil bottle in the blockchain. Later on, IoT sensors can provide tracking information during the transportation stage, including accurate GPS positions and environmental conditions such as temperature, humidity, and light exposure. In this case, blockchain stores only transportation anomalies that might void the delivery contract (e.g., a temperature beyond a certain threshold). Next, IoT monitors can monitor environmental parameters at the retailer storage during the market stage, while the system only records irregularities occurring before the product arrives at the final consumer. Finally, the consumer can effortlessly retrieve the entire history of the bottle of olive oil using blockchain transactions that guarantee the integrity and transparency of the data collected across the entire process.

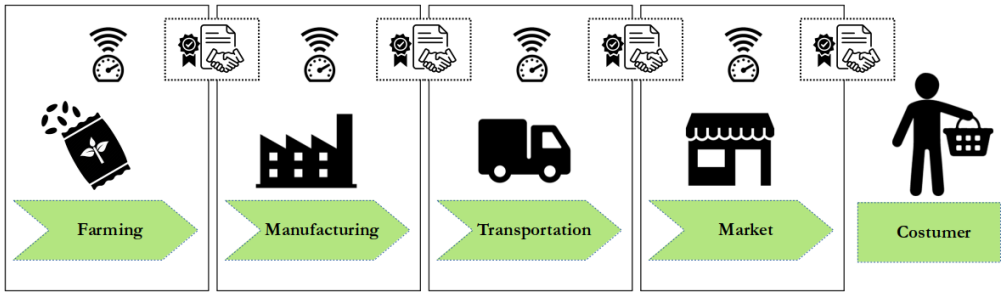


Figure 1. Simplified traceability process for a farm-to-fork scenario.

3.1. High-Level Architecture

To implement the case study, we used our architecture previously described in [5]. The architecture relies on low-cost, constrained IoT sensors and blockchain to achieve transparency, auditability, and immutability of the stored records, enabling a *trustless* environment for the interaction of several unknown actors. A key element is that sensing devices have a unique blockchain identity, enabling them as trustworthy data sources for smart contracts. Further, smart contracts provide a platform to implement the business logic for the different stages of the traceability process. Figure 2 shows a high-level view of the architecture, considering the transportation stage as an example. It is important to highlight that the proposed architecture is blockchain-agnostic. Thus, the only requirement to implement it is a blockchain platform with advanced scripting capabilities (i.e., smart contracts).

3.2. Implementation

We conceptualized the architecture for the traceability systems as a layered software comprising three modules: device, gateway, and blockchain, as shown in Figure 3. First, the device module converts the sensed values into blockchain transactions for the corresponding smart contract. Then, the gateway module acts as a simple and transparent relay component between the device and the blockchain modules, providing the required network connectivity. Finally, the blockchain module includes two types of smart contracts: twin contract and app contract. The twin contract represents the IoT devices, and the app contract implements a different process of the traceability system, such as the transfer of

ownership between stakeholders. Accordingly, app contracts can directly interact with twin contracts, considering them trustworthy oracles for the blockchain system. In the following paragraphs, we briefly describe each module and provide the implementation details.

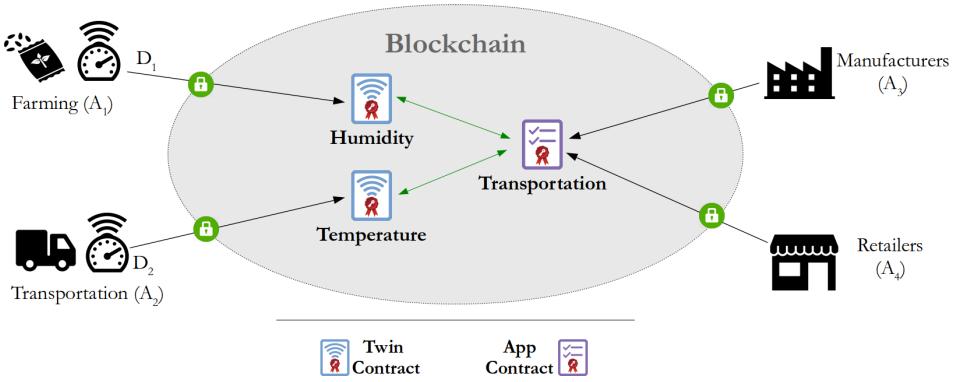


Figure 2. Architecture for the blockchain-based traceability system using transportation stage as an example.

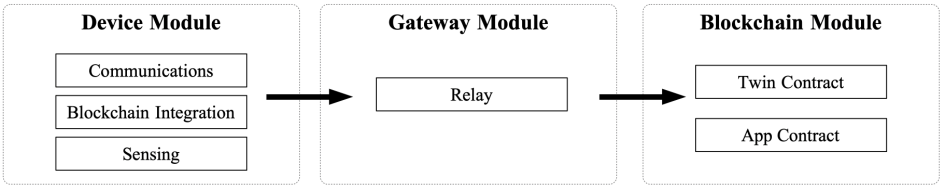


Figure 3. Conceptualized software modules for the blockchain-based traceability system.

3.2.1. Device Module

This module resides in the constrained IoT device and has three components: sensing, blockchain integration, and communications. The sensing component interacts with the physical world to precisely measure certain events. The blockchain integration component converts the sensed values into a digitally signed transaction. Finally, the communications components transmit the digitally signed measurement into the next layer of the system. We used the DHT11 temperature and humidity sensor with its open-source library for the sensing component. For the blockchain component, we used our custom constrained device library for Ethereum [2] and developed an updated version for Sawtooth . The communication component uses a serial interface over USB at 115,200 bps. The module was implemented using C language and the Arduino development framework. Thus, it favors cross-platform compatibility over code optimization.

3.2.2. Gateway Module

The module is a simple and transparent relay hosted on the gateway. The module receives transactions from the device module and forwards them to the blockchain module, making the appropriate protocol conversions. We implemented it in Python version 3.6.

3.2.3. Blockchain Module

This module provides two types of smart contracts to represent the sensors and implement the application's logic (i.e., twin contract and app contract). We tailored our design to two different blockchain platforms: Ethereum and Hyperledger Sawtooth. The selection of these particular platforms is twofold. First, we have references for permissionless (or public) and permissioned (or private) blockchain networks. Second, the selected platforms provide different levels of customization for the transactions, i.e., the records included on the distributed ledger. We implemented a twin contract for the temperature sensor and an app contract for the transportation process. The smart contracts for Ethereum were

implemented using solidity language and for version 0.6.2. The contracts for the Sawtooth platform were implemented using Python 3.6.

4. Infrastructure Costs

We consider that the blockchain-based traceability application supports the interactions I of a group of unknown actors, composed of physical entities A (e.g., providers, producers, distributors, retailers, etc.), and sensing IoT devices D (e.g., sensors at farming site, on the transport, etc.). These actors are identified only by their private/public keys, an assumption aligned with current literature [2,35]. Furthermore, the interactions of these actors are represented on the blockchain by digitally signed transactions Tx. We chose one month as the minimal time frame to evaluate the cost of the system. Thus, for a given month m, we define the costs of the traceability application $C(m)$ as the cost of the infrastructure supporting the transactions Tx.

4.1. Public Blockchain Costs

In public blockchain networks (e.g., Ethereum or Bitcoin), only the transactions that create new information (i.e., modify the state of the blockchain) have a monetary cost, typically expressed in terms of cryptocurrency used in the network. We divided this cost into two components (C_B and C_O), one for the bootstrap phase and one for the operation phase of the application. The bootstrap phase C_B (where $m = 0$) includes the price paid for the transactions needed to deploy the application's logic, which, in our case, is creating the SmartTwin (TxT) and TwinApp (TxA) contracts. The operation phase ($m > 0$) considers the price paid for transactions representing the sensors' interactions (TxI), and transactions that transfer value between actors (TxV). Thus, we can define the infrastructure cost of the application in a public blockchain $C_{pub}(m)$ as

$$C_{pub}(m) = \begin{cases} C_B = C_{TxT} + C_{TxA}, & \text{if } m = 0; \\ C_O = C_{TxI} + C_{TxV}, & m > 0. \end{cases} \quad (1)$$

The price paid for all the transactions of type i where $i \in \{T, A, I, V\}$, links the computational cost of the transaction (CC_i) to the network cryptocurrency price $P(m)$ using a processing time factor μ. To obtain the total price paid for all the transactions, we define Q_{Txi} as the total number of transactions of Txi. Thus, the following function defines the price paid for all transactions executed in a given month m.

$$C_{Txi} = CC_i \ \ \mu \ \ P(m) \ \ Q_{Txi}(m) \quad (2)$$

The computational cost CC_i is related to the operations required to complete the transactions (e.g., bytes written, variables used, etc.). Typically, simple transactions, such as a value transfer, have lower cost than transactions used for smart contracts that execute more complex logic. This computation cost translates into a cryptocurrency cost, using a processing μ. Transactions with higher cryptocurrency cost are more attractive to the nodes of the network and are typically processed faster. However, different network conditions might affect these results.

To translate this cryptocurrency cost to a monetary cost, we need the price of the cryptocurrency. Given the high volatility of this value, its accurate estimation goes beyond the scope of this paper. In our model, we map this variation using a function $P(m)$ for the price of the cryptocurrency at a given month. The user should define this function according to the blockchain used or the particular use case. The values of the function can be a fixed value (an average for all months), a value for each month, or an estimation function.

Finally, the number of transactions Q_{Txi} is defined as

$$Q_{Txi}(m) = \begin{cases} D & \text{if } i = T; \\ N_A, & \text{if } i = A; \\ N_I, & \text{if } i = I; \\ N_V, & \text{if } i = V. \end{cases} \quad (3)$$

where D is the total number of IoT devices in the system (TwinContracts); N_A is the total number of distributed applications in the system (AppContracts); N_I is the total number of IoT measurements registered on the blockchain; and N_V is the total value of transfers required in the application. For our use case, these transactions are when the item is transferred between actors (i.e., producer to transporter, transporter to retailer, etc.).

Summarizing, to estimate the infrastructure cost of a blockchain-based traceability system in a public network, we need to characterize the application by assigning values for all the parameters listed in Table 1.

Table 1. Parameters for the infrastructure cost model using a public blockchain.

Parameter	Description
A	Number of actors
D	Number of IoT devices
$P(m)$	Price of the cryptocurrency on a given month
μ	Time factor for processing transactions
CC_T	Computational cost of SmartTwin transactions
CC_A	Computational cost of TwinApp transactions
CC_I	Computational cost of interactions transactions
CC_V	Computational cost of value transactions
N_A	Total AppContracts
N_I	Total IoT measures
N_V	Total value transfers among actors

4.2. Private Blockchain

Opposite to a public blockchain, the number of transactions does not affect the infrastructure cost of a private network. In this case, the cost is the price paid for managing the nodes running the network where the transactions occur. The configuration of these nodes depends on several parameters adjusted according to expected performance (i.e., latency, throughput) and the existing trust between actors ([36,37]).

Here, we define the cost for the private infrastructure in a given month m as $C_{Pri}(m)$. Similarly to the public model, we split the cost into two phases: bootstrap ($m = 0$) and operation ($m > 0$). In the bootstrap phase, the cost is the initial investment to acquire the required number of nodes N that provide the infrastructure. The operation cost is the operational expense for running and operating the nodes.

$$C_{Pri}(m) = \begin{cases} N \ C_{node} & \text{if } m = 0; \\ N \ C_{op} & m > 0. \end{cases} \quad (4)$$

where C_{node} is the monetary cost of purchasing a new node, C_{op} is the cost of operating a node, and N is the number of nodes comprising the network. Similar to the models presented in [38], we estimate C_{op} as a percentage of C_{node}, using an operation factor F_o, such that

$$C_{op} = F_o \ C_{node} \quad (5)$$

We define that the number of nodes N is related to the number of actors A by a trust factor F_t, which is the fraction of nodes required in relation to the total number of actors. For instance, if 100 actors agree that only 30 different nodes are required to support the

infrastructure, this translates into a trust factor of 70%. Similarly, a 100% trust factor will translate into a centralized system. For simplicity, we consider a fixed number of actors A, and that one node represents one actor. Therefore, the number of nodes N is defined as

$$N = A(1 - F_t). \tag{6}$$

Summarizing, the cost of a blockchain-based traceability system on a private network requires identifying the parameters of Table 2 that characterize the application and the network where it is executed. From this table, it is clear that evaluating the costs of a blockchain system requires fewer parameters than those needed for evaluating a public system since its cost does not depend on the number of transactions or actors [39].

Table 2. Parameters for the infrastructure cost model using a private blockchain.

Parameter	Description
A	Number of actors
C_{node}	Monetary cost of a node
F_t	Trust factor among actors
F_o	Operation factor

5. Performance Evaluation

5.1. Experimental Setup

To characterize the blockchain-based traceability application from the device's perspective, we tested our cross-platform prototype on six different microcontroller boards (MCUs) from the AVR, ARM, and ESP32 architectures. A comprehensive evaluation of all possible IoT boards is beyond the scope of this paper; however, the selected pool should provide a reference for other scenarios. Table 3 presents all the boards, detailing clock speed in megahertz (Mhz), program space in kilobytes (kb), memory size in kb, model, and a reference price (updated to May 2021).

Table 3. The hardware platforms used during our performance evaluation.

Device	Model	MCU	Architecture	Clock (Mhz)	Prog. Space (kb)	Mem. Size (kb)	Price (Eur)
UNO	Arduino Uno	ATMega328P	8-Bit AVR	16	32	2	18
EVERY	Arduino nano every	ATMega4809	8-Bit AVR	20	48	6	12
L031	STM32L031	Cortex M0+	32-Bit ARM	32	32	8	10
F303	STM32F303	Cortex M4F	32-Bit ARM	72	64	12	10
L452	STM32L452	Cortex M4	32-Bit ARM	80	512	96	12
ESP32	ESP DevKit	WRover-E	32-Bit ESP32	80	1024	320	10

For the Ethereum blockchain, we used the official Geth client (version 1.10.1-stable), while for Sawtooth we used the official client (version 1.2.6). The nodes run on separated virtual machines with 4 GB of RAM, 20 GB of SSD, and 4 vCPU on an OpenStack server using a clean Linux Ubuntu installation (version 18.04). The scripts that deploy and interact with the smart contracts were implemented using Python (version 3.6) and ran on a Lenovo T490s notebook, with 16 GB of RAM, 256 SSD disk, and an Intel i7 processor at 1.90 GHz over a clean Linux Ubuntu (version 18.04). The notebook and the nodes shared the same LAN connection.

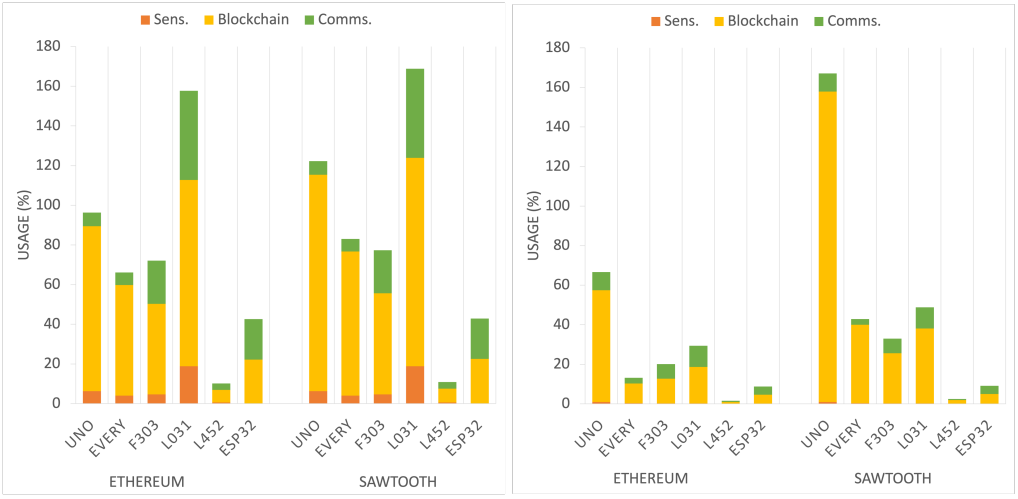
5.2. Device Module Footprint

Using the statistics provided by the compilers, we estimated the footprint of each component of the device module. The results, in terms of disk usage (program space), are shown using Table 4a (absolute values) and Figure 4a (normalized to the total available). The same approach was taken for memory usage, using Table 4b and Figure 4b. Sensing (Sens) and communications (Comms) components are the same for both implementations.

The footprint of the blockchain integration component for Ethereum is represented by Eth, and the blockchain integration component for Sawtooth is represented by Saw. Our results show that an 8-bit board with 32 kb of disk space can fit the Ethereum implementation; however, the same amount of disk space is not enough on the 32-bit boards. Results also show higher requirements for the Sawtooth implementation, which allows only four boards to run the entire device module for this platform.

Table 4. Resource usage of each component of the device module expressed in bytes.

(**a**) Disk usage					
Device	**Available**	**Sens.**	**Eth**	**Saw**	**Comms.**
UNO	32,256	2014	26,840	35,214	2202
EVERY	49,152	1956	27,428	35,735	3105
F303	65,536	3040	29,940	33,404	14,240
L031	32,768	6168	30,796	34,432	14,724
L452	524,288	3032	33,120	36,568	17,336
ESP32	1,310,720	1496	289,318	293,674	267,270
(**b**) Memory usage					
Device	**Available**	**Sens.**	**Eth**	**Saw**	**Comms.**
UNO	2048	207	1158	3214	188
EVERY	6144	196	611	2435	177
F303	12,288	928	1540	3124	908
L031	8192	896	1508	3102	876
L452	163,840	936	1560	3144	916
ESP32	327,680	13,660	14,996	16,292	13,612



(**a**) (**b**)

Figure 4. (**a**) Disk usage and (**b**) memory usage of each component of the device module with respect to the total available.

5.3. Size and Processing Times of the Blockchain Transactions

We measured the transaction size and processing time on the device module. We ran 100 experiments of sensing a value, then created a digitally signed transaction, and finally transmitted the transaction over serial protocol. Table 5 shows the average results for the 100 experiments. The results show that the processing time for creating a Sawtooth transaction is more than twice the time required for Ethereum. Similarly, the average size

of the Sawtooth transaction is almost seven times the size of the Ethereum transaction with the same sensed value.

Table 5. Average processing times for each component of the device module (expressed in ms) and transaction size (in bytes) for the blockchain transaction for each platform.

	Ethereum				Sawtooth			
Device	**Sens.**	**Block.**	**Comms.**	**Total**	**Sens.**	**Block.**	**Comms.**	**Total**
UNO	32	4245	23	4300	-	-	-	-
EVERY	32	4170	23	4225	32	8388	164	8584
F303	32	208	23	263	32	421	164	617
L031	-	-	-	-	-	-	-	-
L452	32	160	23	196	32	278	163	473
ESP32	32	83	23	139	32	146	163	341
	Avg. Transaction size: 134 bytes.				**Avg. Transaction size: 925 bytes.**			

5.4. Device Module Power Consumption

We measured each board's power consumption during the experiments described in the previous section. To this end, we used an Otii device [40], which measures energy consumption at a rate of 1000 samples/s with an accuracy of $\pm(1\% + 0.5\ \mu A)$ at 5 V.

However, estimating the energy consumption is a difficult task that depends on several factors, particularly software [41]. Thus, we implemented a simple repetitive algorithm that includes a 5 s idle state, performs a work cycle (i.e., measuring and creating the blockchain transaction), and, finally, sends it using the communications interface. Furthermore, considering that our library uses the Arduino IDE [2], we favored cross-compatibility for all architectures and did not implement any low-power mode on the boards.

In addition, we used an idle state of 5 s before each working cycle to place the power consumption in context. However, it is important to notice that no low-power mode was used for the idle state. Table 6a,b show the average power consumption for the Ethereum and Sawtooth implementation, respectively. As confirmed by the results, the processing time for creating a Sawtooth transaction is longer, and, therefore, imposes higher power requirements for the sensing IoT device.

Table 6. Average power requirements of the device module (at 5 V) for each IoT device on each implementation.

	Idle State				Working State			
Device	**Current (mA)**	**Current (var)**	**Energy (J)**	**Time (s)**	**Current (mA)**	**Current (var)**	**Energy (J)**	**Time (s)**
UNO	34.38	2.85	0.86	5.0	34.36	2.37	0.73	4.30
EVERY	29.46	0.17	0.73	5.0	30.02	0.23	0.63	4.22
F303	65.0	0.12	1.63	5.0	67.93	1.0	0.08	0.26
L452	48.79	0.0	1.27	5.0	50.15	0.08	0.04	0.19
ESP32	42.26	0.03	1.06	5.0	61.98	55.23	0.03	0.13

(**a**) Ethereum Implementation

	Idle State				Working State			
Device	**Current (mA)**	**Current (var)**	**Energy (J)**	**Time (s)**	**Current (mA)**	**Current (var)**	**Energy (J)**	**Time (s)**
EVERY	29.4	0.44	0.73	5.0	29.99	0.58	1.28	8.58
F303	64.67	0.27	1.62	5.0	67.86	1.76	0.2	0.61
L452	51.87	0.0	1.35	5.0	52.95	0.06	0.12	0.47
ESP32	42.23	0.02	1.06	5.0	59.93	43.35	0.09	0.34

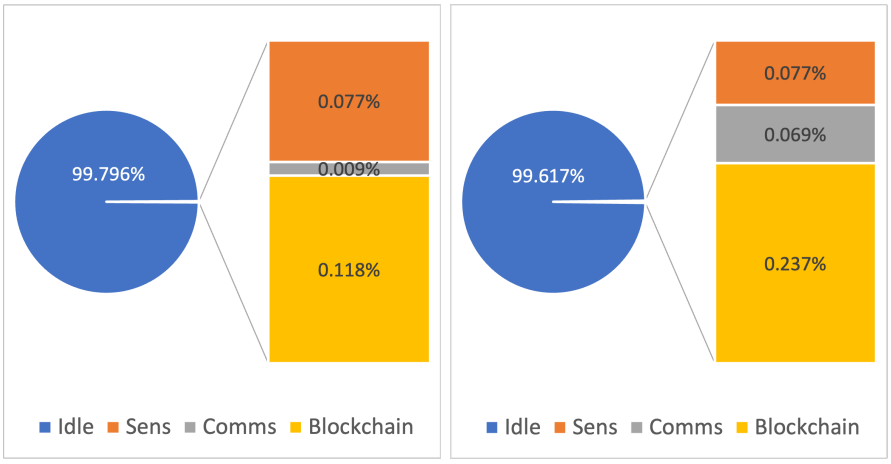
(**b**) Sawtooth Implementation

5.5. Daily Energy Budget

To better understand the significance of the power requirements for the traceability system, we evaluated blockchain's energy impact on constrained sensing devices. To this end, we considered the following simple energy budget, aligned with the models found in current literature [5,35,42].

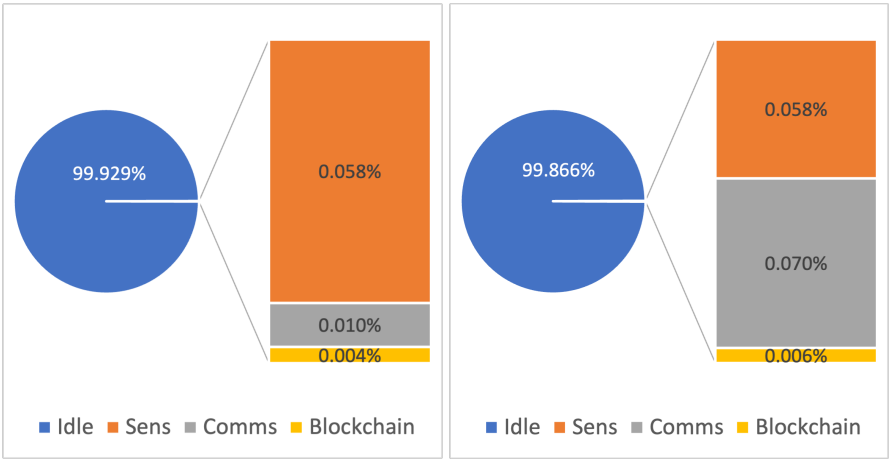
$$\mathcal{E}_{budget} = \mathcal{E}_{idle} + \mathcal{E}_{sensing} + \mathcal{E}_{communications} + \mathcal{E}_{blockchain}$$

Here, the energy budget of the IoT devices in a given timeframe $\mathcal{E}_{budget}$ is the sum of the energy required in idle state $\mathcal{E}_{idle}$, the energy required for the sensing component $\mathcal{E}_{sensing}$, the energy required for the communications component $\mathcal{E}_{communications}$, and the energy required for blockchain component $\mathcal{E}_{blockchain}$. We estimated the budget in one hour of work, using the experiments of power consumption (Table 6) and processing times (Table 5). Finally, we considered that the sensor measures the temperature every 5 min, and only one of these measurements is an event to be recorded on the blockchain. Hence, Figure 5 shows the energy budget for the board with the highest energy consumption (EVERY). Similarly, Figure 6 shows the energy budget for the board with the lower energy consumption (ESP32). Results show that, despite the power requirements of blockchain systems, the impact on the energy budget is minimal compared to the other operations of the device (e.g., sensing and communications). Furthermore, the idle state is still the most power-consuming element of the energy budget in our event-based traceability application.



(**a**) Ethereum Implementation (**b**) Sawtooth Implementation

Figure 5. Impact of blockchain operations in the energy budget for the board with the higher power requirements (EVERY).



(**a**) Ethereum Implementation (**b**) Sawtooth Implementation

Figure 6. Impact of blockchain operations in the energy budget for the board with the lower power requirements (ESP32).

6. Costs Evaluation

To estimate the infrastructure costs of the blockchain-based traceability system, for both Ethereum and Sawtooth implementations, we need to identify the parameters described in Sections 4.1 and 4.2, respectively. As described in Section 3, we consider six different actors: provider, producer, processor, distributor, retailer, and consumer. We focus only on the four most typical IoT devices (i.e., producer sensors in the field, processor sensors on the production plant, distributor sensors on the transportation, and retailer sensors on the storage). For simplicity, we evaluate the farm-to-fork of a single product (i.e., a bottle of olive oil) using a transportation contract. However, the proposed model allows one to easily evaluate more complex scenarios, choosing different parameters to characterize the application. In the following, we describe our rationale to characterize our application.

6.1. Application Characterization

The number of actors of actors A is six and the number of devices D is four. This means that we need four TwinContracts (one for each device) and one AppContract (for the transportation) which makes N_V equal to one. Since our use case considers an event-based traceability scenario, we estimated that all four sensors will register only 10 values in total (e.g., five values at the field, five during transportation, etc.). This means that N_I is 10. Similarly, we estimate that there will be four value transfers between the actors (i.e., provider to producer, producer to processor, processor to transportation, transportation to retail), rendering N_V equal to four.

Our implementation uses Ethereum; thus, the the computational cost of the transactions corresponds to the gas needed to execute a transaction, and the μ parameter on Ethereum can be directly mapped to the gas price. We defined the gas price as 60 gwei, as $\mu = 60 \times 10^{-9}$, a value currently recommended for average transaction times. Based on the implementation described in Section 3.2, Table 7 shows the gas required for creating the TwinContract (TwinSC), creating the AppContract for transportation (AppSC), registering an IoT measurement (setValue), and transferring an item (transferToken).

Table 7. Gas needed for the transactions in the Ethereum implementation.

Transaction	Required Gas
Create Sensor (TwinSC)	557,364
Create Transportation App (AppSC)	3,343,572
Register IoT Measurement (setValue)	44,468
Transfer Value (transferToken)	63,681

The price of the cryptocurrency $P(m)$ is the price of ETH (Ethereum's cryptocurrency) which is quite volatile. Thus, historic values can provide a good reference for evaluating different scenarios. We use the statics provided by Etherscan [43], and estimate a fixed Ethereum price from of USD 1087 ($P(m) = 1087$) based on the monthly average price from 2019 to 2021. Finally, for the cost of the node C_{node}, we used the minimum hardware requirements described on the official documentation for Sawtooth. This considers a CPU with 2+ cores, 4 GB+ RAM, and 320 GB SSD of free disk space. At the time of writing, this translates into a computer of USD 300. We consider the factor of operation of 40% of the node price, and that there is 0% of trust among the actors.

Table 8 summarizes all the parameters that we consider to characterize our application. We want to highlight that the parameters can be set at any other value to adapt to other situations or scenarios, based on the reader's experience or the particular case to study.

Table 8. Parameters for characterizing the traceability application.

Parameter	Description	Value
A	Number of actors	6
D	Number of IoT devices	4
$P(m)$	Price of the crypto-currency on a given month	1087 USD
μ	Time factor for processing transactions	60×10^{-9}
CC_T	Computational cost of create SmartTwin transactions	557,364
CC_A	Computational cost of create TwinApp transactions	3,343,572
CC_I	Computational cost of Interactions transactions	44,468
CC_V	Computational cost of Value transactions	63,681
N_A	Total AppContracts	1
N_I	Total IoT measures	10
N_V	Total value transfers among actors	4
C_{node}	Monetary cost of a node	300 USD
F_o	Operation factor	0.4
F_t	Trust factor among actors	0

6.2. Infrastructure Cost Comparison

Once the application is characterized, several cost evaluations can be performed. Here, we compare the total cost (bootstrap and operation) of a private Sawtooth infrastructure and the public Ethereum infrastructure. For Ethereum, we evaluate the cost for 1, 25, and 50 products, using the same bootstrap costs. Figure 7 depicts these evaluation for the four cases.

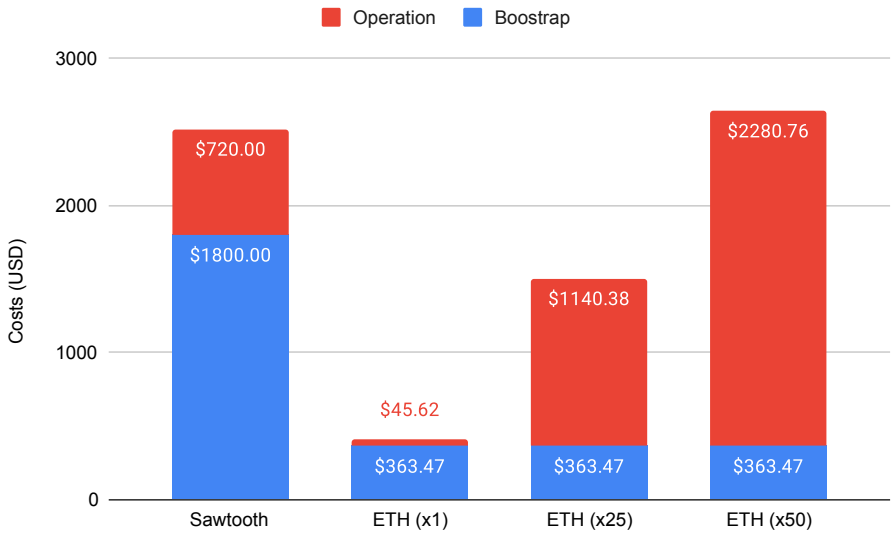


Figure 7. Infrastructure costs (bootstrap and operation phases) for the traceability application on Sawtooth and Ethereum implementations.

As Figure 7 shows, the total cost of the application is USD 409 for registering the farm-to-fork case of one product on the Ethereum network. This value is approximately 16% of the total cost of the private Sawtooth infrastructure (USD 2520). Furthermore, for the total cost of Sawtooth, we could handle the information for approximately 50 products on Ethereum (USD 2644). Although the values and rationale used to characterize the application are based on current literature, these results should be interpreted with caution. On the one hand, the conversion rate of Ethereum is quite volatile, which also influences the gas price. If the cryptocurrency price is at a low price, such as in 2020, when the average price was USD 307, and the gas price was around 30 gwei, the total cost for one product will be only USD 58. However, if the price goes up, as in 2021, when it was USD 2774 with an average gas price over 100 gwei, the total cost for one product will be USD 1740.

On the other hand, a private blockchain infrastructure provides several advantages regarding processing times and cost. For example, as reported by authors on [39], Hyperledger Sawtooth can reach a maximum throughput of about 2300 tx/s when using a similar infrastructure, as we have evaluated. Furthermore, the authors reported a throughput of 1000 tx/s before starting to decrease the system's performance. Therefore, considering that the number of transactions does not influence the infrastructure cost, the current evaluation could handle even processing 1000 products per second within the same infrastructure cost. However, the infrastructure cost will increase when adding more nodes, as depicted in Figure 8. Typically, this increment of nodes happens when more untrusted actors enter the system and require participation in the network's consensus algorithm [1,44]. Furthermore, permissionless blockchain networks based on PoW consensus are the most secure platform for developing decentralized applications [45]. Ethereum is estimated to have almost 8000 active nodes [46]. This network size provides high levels of redundancy and fault-tolerance, increasing the availability of a blockchain-based traceability system and providing an open platform for integrating several unknown actors to increase the system value.

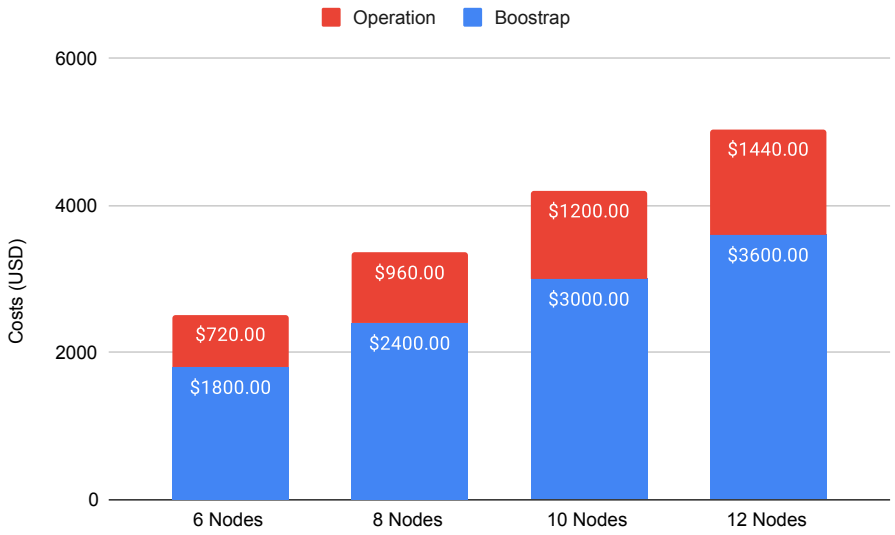


Figure 8. Infrastructure costs (bootstrap and operation phases) for the traceability application on Sawtooth considering different number of nodes.

6.3. Blockchain Processing Times

We evaluated the transaction processing time for executing the setValue() operation as it is the most frequent transaction in the system. To avoid the monetary cost of using the Live Ethereum network, we use the Ropsten test network. This network uses the same proof-of-work consensus algorithm and provides similar time response to the live Ethereum network, but without the associated monetary cost. We sent one transaction approximately each 30 min over a period of one week, for a total of 336 transactions, using the gas price of 60 Gwei. Figure 9 shows the distribution of the processing times for the transactions. The average processing time was 17 s, which is within the current limit of the network. Furthermore, just a couple of transactions took more than one minute to process. As a reference, in the private Sawtooth network, the average processing time for the transaction was less than 1 s.

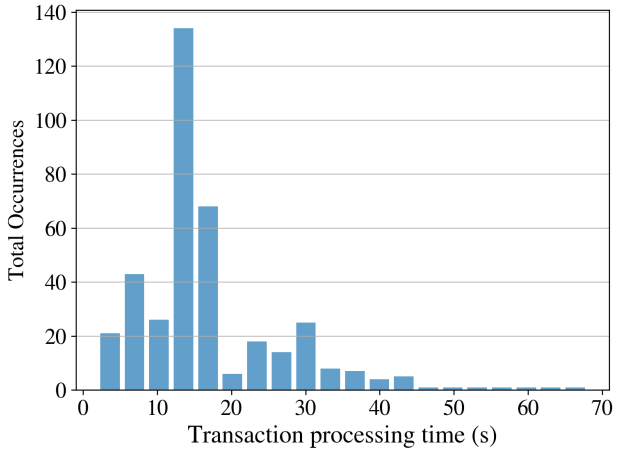


Figure 9. Distribution of the transaction processing times in the Ethereum Ropsten network.

7. Discussion and Future Works

This paper described, characterized, and evaluated a blockchain-based IoT traceability application using public and private networks. First, we evaluated the impact of blockchain operations on constrained sensing devices. To this end, we described an event-based traceability scenario and implemented the system using two different blockchain platforms.

Then, we benchmarked six off-the-shelf IoT boards with a price lower than USD 20 and with very limited memory and program space. Finally, we evaluated the cost of the infrastructure to support the traceability system by defining a cost model based on a series of parameters that characterize the system.

Our experiments showed that, for constrained sensing devices, the Ethereum blockchain requires fewer resources than the Sawtooth blockchain. For example, the Ethereum implementation of the device module can work on an 8-bit board with only 2 kb of memory and 32 kb of disk space. However, these are the minimum requirements, as other device functionalities (such as different communications technologies or sensors) might need additional space or memory. Nevertheless, other IoT boards within the same price range provide more resources to include additional functionalities. Conversely, the implementation for the Sawtooth blockchain has a bigger memory and disk space footprint, as the Sawtooth protocol creates a batch of transactions rather than single transactions, as in the case of the Ethereum protocol. This difference increases the processing times and uses an encoding scheme that adds additional overhead to the transaction size, making it six times larger than an Ethereum counterpart.

The Sawtooth protocol's overhead also impacts constrained devices regarding the energy budget. For example, in one hour of work, measuring every 5 min, in the board with the higher power requirements, blockchain operations for Sawtooth accounted for 0.24% of the total energy budget, while Ethereum accounted only for 0.12%. However, in the context of the entire energy budget, the requirements of both implementations are minimal, as the idle state accounts for more than 99% of the total requirement.

The findings of this study support the idea that even constrained IoT devices can directly interact with blockchain networks such as Ethereum and Sawtooth. For a traceability system, this means that low-cost IoT sensors can function as a direct actor on the blockchain system, enabling a root of trust for the sensed data and becoming one step closer to trustworthy oracles for the entire system.

Regarding costs, the proposed model easily characterizes the application and evaluates the infrastructure in different scenarios. As a result, the model clarifies the expected benefits and costs, which benefit the potential adoption of the traceability system. For example, using the historical prices of the cryptocurrency, our evaluation showed that the cost of handling seven products on Ethereum (USD 2510) is almost equal to the cost of setting up and managing the private Sawtooth network with six nodes (USD 2520). Although the values and rationale used to characterize the application are considering current literature, these results should be interpreted with caution. Even if a private blockchain infrastructure provides several advantages regarding processing times and cost, including additional unknown actors might change the cost and performance of the network. Furthermore, permissionless blockchain networks based on PoW consensus are currently the most secure platform for developing decentralized applications and they provide an open platform for integrating several unknown actors to increase the system value.

These findings add to a growing body of literature on the combination of IoT devices and blockchain systems, particularly highlighting the role of low-cost, constrained IoT devices in traceability applications. Furthermore, our study provides an empirical reference for the characterization and cost evaluation of blockchain-based traceability systems that use IoT devices.

Future works include migrating the software library for the device module to other blockchain platforms. Similarly, benchmarking the devices with other communications technologies, such as NB-IoT, LoraWan, or 5G, will provide additional insights for other possible traceability scenarios. Finally, another interesting research path is characterizing and evaluating traceability systems combining private and public networks.

Author Contributions: All co-authors contributed equally to the preparation of this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request due to project restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ali, M.S.; Vecchio, M.; Pincheira, M.; Dolui, K.; Antonelli, F.; Rehmani, M.H. Applications of Blockchains in the Internet of Things: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* **2018**, *21*, 1676–1717. [CrossRef]
2. Pincheira, M.; Vecchio, M. Towards Trusted Data on Decentralized IoT Applications: Integrating Blockchain in Constrained Devices. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
3. Rimba, P.; Tran, A.B.; Weber, I.; Staples, M.; Ponomarev, A.; Xu, X. Quantifying the Cost of Distrust: Comparing Blockchain and Cloud Services for Business Process Execution. *Inf. Syst. Front.* **2020**, *22*, 489–507. [CrossRef]
4. Mistry, I.; Tanwar, S.; Tyagi, S.; Kumar, N. Blockchain for 5G-enabled IoT for industrial automation: A systematic review, solutions, and challenges. *Mech. Syst. Signal Process.* **2020**, *135*, 106382. [CrossRef]
5. Pincheira, M.; Vecchio, M.; Giaffreda, R.; Kanhere, S.S. Exploiting constrained IoT devices in a trustless blockchain-based water management system. In Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, ON, Canada, 2–6 May 2020; pp. 1–7.
6. Zhang, K.; Zhu, Y.; Maharjan, S.; Zhang, Y. Edge Intelligence and Blockchain Empowered 5G Beyond for the Industrial Internet of Things. *IEEE Netw.* **2019**, *33*, 12–19. [CrossRef]
7. Pincheira, M.; Donini, E.; Giaffreda, R.; Vecchio, M. A Blockchain-Based Approach To Enable Remote Sensing Trusted Data. In Proceedings of the 2020 IEEE Latin American GRSS ISPRS Remote Sensing Conference (LAGIRS), Santiago, Chile, 21–26 March 2020; pp. 652–657. [CrossRef]
8. Pincheira, M.; Antonini, M.; Vecchio, M. Integrating the IoT and Blockchain Technology for the Next Generation of Mining Inspection Systems. *Sensors* **2022**, *22*, 899 . [CrossRef]
9. Kayikci, Y.; Subramanian, N.; Dora, M.; Bhatia, M.S. Food supply chain in the era of Industry 4.0: Blockchain technology implementation opportunities and impediments from the perspective of people, process, performance, and technology. *Prod. Plan. Control* **2020**, *33*, 301–321. [CrossRef]
10. Caro, M.P.; Ali, M.S.; Vecchio, M.; Giaffreda, R. Blockchain-based traceability in Agri-Food supply chain management: A practical implementation. In Proceedings of the 2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany), Tuscany, Italy, 8–9 May 2018; pp. 1–4.
11. Corallo, A.; Latino, M.E.; Menegoli, M.; Striani, F. What factors impact on technological traceability systems diffusion in the agrifood industry? An Italian survey. *J. Rural. Stud.* **2020**, *75*, 30–47. [CrossRef]
12. Behnke, K.; Janssen, M. Boundary conditions for traceability in food supply chains using blockchain technology. *Int. J. Inf. Manag.* **2020**, *52*, 101969. [CrossRef]
13. Destefanis, G.; Marchesi, M.; Ortu, M.; Tonelli, R.; Bracciali, A.; Hierons, R. Smart contracts vulnerabilities: A call for blockchain software engineering? In Proceedings of the 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), Campobasso, Italy, 20 March 2018; pp. 19–25.
14. Khutsoane, O.; Isong, B.; Abu-Mahfouz, A.M. IoT devices and applications based on LoRa/LoRaWAN. In Proceedings of the IECON 2017—43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November 2017; pp. 6107–6112.
15. Pincheira, M.; Vecchio, M.; Giaffreda, R. Benchmarking Constrained IoT Devices in Blockchain-Based Agri-Food Traceability Applications. In *International Congress on Blockchain and Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 212–221.
16. Vacca, A.; Di Sorbo, A.; Visaggio, C.A.; Canfora, G. A systematic literature review of blockchain and smart contract development: Techniques, tools, and open challenges. *J. Syst. Softw.* **2021**, *174*, 110891. [CrossRef]
17. Kamilaris, A.; Fonts, A.; Prenafeta-Boldύ, F.X. The rise of blockchain technology in agriculture and food supply chains. *Trends Food Sci. Technol.* **2019**, *91*, 640–652. [CrossRef]
18. Demestichas, K.; Peppes, N.; Alexakis, T.; Adamopoulou, E. Blockchain in Agriculture Traceability Systems: A Review. *Appl. Sci.* **2020**, *10*, 4113. [CrossRef]
19. Feng, H.; Wang, X.; Duan, Y.; Zhang, J.; Zhang, X. Applying blockchain technology to improve agri-food traceability: A review of development methods, benefits and challenges. *J. Clean. Prod.* **2020**, *260*, 121031. [CrossRef]
20. Sunny, J.; Undralla, N.; Pillai, V.M. Supply chain transparency through blockchain-based traceability: An overview with demonstration. *Comput. Ind. Eng.* **2020**, *150*, 106895. [CrossRef]
21. Tian, F. An agri-food supply chain traceability system for China based on RFID & blockchain technology. In Proceedings of the 2016 13th International Conference on Service Systems and Service Management (ICSSSM), Kunming, China, 24–26 June 2016; pp. 1–6. [CrossRef]

22. Tian, F. A supply chain traceability system for food safety based on HACCP, blockchain and Internet of Things. In Proceedings of the 2017 International Conference on Service Systems and Service Management, Dalian, China, 16–18 June 2017 pp. 1–6.
23. Lin, J.; Shen, Z.; Zhang, A.; Chai, Y. Blockchain and IoT Based Food Traceability for Smart Agriculture. In Proceedings of the 3rd International Conference on Crowd Science and Engineering, Singapore, 28–31 July 2018, ICCSE'18 ; Association for Computing Machinery: New York, NY, USA, 2018. [CrossRef]
24. Kim, M.; Hilton, B.; Burks, Z.; Reyes, J. Integrating Blockchain, Smart Contract-Tokens, and IoT to Design a Food Traceability Solution. In Proceedings of the 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 1–3 November 2018; pp. 335–340. [CrossRef]
25. Salah, K.; Nizamuddin, N.; Jayaraman, R.; Omar, M. Blockchain-Based Soybean Traceability in Agricultural Supply Chain. *IEEE Access* **2019**, *7*, 73295–73305. [CrossRef]
26. Heiss, J.; Eberhardt, J.; Tai, S. From oracles to trustworthy data on-chaining systems. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019.
27. Croman, K.; Decker, C.; Eyal, I.; Gencer, A.E.; Juels, A.; Kosba, A.; Miller, A.; Saxena, P.; Shi, E.; Gün Sirer, E.; et al. On Scaling Decentralized Blockchains. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 106–125.
28. Demir, M.; Turetken, O.; Ferworn, A. A Financial Evaluation Framework for Blockchain Implementations. In Proceedings of the 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 17–19 October 2019; pp. 0715–0722. [CrossRef]
29. Wu, K.; Ma, Y.; Huang, G.; Liu, X. A first look at blockchain-based decentralized applications. *Software Pract. Exp.* **2021**, *51*, 2033–2050. [CrossRef]
30. Zanoni, B. Extra-virgin olive oil traceability. In *The Extra-Virgin Olive Oil Handbook*; John Wiley and Sons: Hoboken, NJ, USA, 2014; pp. 245–250.
31. Accorsi, R.; Bortolini, M.; Baruffaldi, G.; Pilati, F.; Ferrari, E. Internet-of-things paradigm in food supply chains control and management. *Procedia Manuf.* **2017**, *11*, 889–895. [CrossRef]
32. Verdouw, C.N.; Wolfert, J.; Beulens, A.; Rialland, A. Virtualization of food supply chains with the internet of things. *J. Food Eng.* **2016**, *176*, 128–136. [CrossRef]
33. Kumar, M.V.; Iyengar, N. A framework for Blockchain technology in rice supply chain management. *Adv. Sci. Technol. Lett* **2017**, *146*, 125–130.
34. Umamaheswari, S.; Sreeram, S.; Kritika, N.; Prasanth, D.J. Biot: blockchain based IoT for agriculture. In Proceedings of the 2019 11th International Conference on Advanced Computing (ICoAC), Chennai, India, 18–20 December 2019; pp. 324–327.
35. Pincheira, M.; Vecchio, M.; Giaffreda, R.; Kanhere, S.S. Cost-effective IoT devices as trustworthy data sources for a blockchain-based water management system in precision agriculture. *Comput. Electron. Agric.* **2021**, *180*, 105889. [CrossRef]
36. Leal, F.; Chis, A.E.; González-Vélez, H. Performance Evaluation of Private Ethereum Networks. *SN Comput. Sci.* **2020**, *1*, 1–17. [CrossRef]
37. Schäffer, M.; Di Angelo, M.; Salzer, G. Performance and scalability of private Ethereum blockchains. In *International Conference on Business Process Management*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 103–118.
38. Bibi, S.; Katsaros, D.; Bozanis, P. Business Application Acquisition: On-Premise or SaaS-Based Solutions? *IEEE Software* **2012**, *29*, 86–93. [CrossRef]
39. Ampel, B.; Patton, M.; Chen, H. Performance Modeling of Hyperledger Sawtooth Blockchain. In Proceedings of the 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), Shenzhen, China, 1–3 July 2019; pp. 59–61. [CrossRef]
40. Qoitech. Power Analyzer Qoitech. 2022. Available online: https://www.qoitech.com/ (accessed on 14 January 2022).
41. Georgiou, K.; Xavier-de Souza, S.; Eder, K. The IoT Energy Challenge: A Software Perspective. *IEEE Embed. Syst. Lett.* **2018**, *10*, 53–56. [CrossRef]
42. Bouguera, T.; Diouris, J.F.; Chaillout, J.J.; Jaouadi, R.; Andrieux, G. Energy consumption model for sensor nodes based on LoRa and LoRaWAN. *Sensors* **2018**, *18*, 2104. [CrossRef] [PubMed]
43. etherescan.io. Ether Daily Price (USD) Chart. 2022. Available online: https://etherscan.io/chart/etherprice (accessed on 14th January 2022).
44. Moschou, K.; Theodouli, A.; Terzi, S.; Votis, K.; Tzovaras, D.; Karamitros, D.; Diamantopoulos, S. Performance Evaluation of different Hyperledger Sawtooth transaction processors for Blockchain log storage with varying workloads. In Proceedings of the 2020 IEEE International Conference on Blockchain (Blockchain), Rhodes Island, Greece, 2–6 November 2020; pp. 476–481. [CrossRef]
45. Conoscenti, M.; Vetrò, A.; De Martin, J.C. Blockchain for the Internet of Things: A systematic literature review. In Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, Morocco, 29 November–2 December 2016; pp. 1–6. [CrossRef]
46. Antonopoulos, A.M.; Wood, G. *Mastering Ethereum: Building Smart Contracts and Dapps*; O'Reilly Media: Sebastopol, CA, USA, 2018.