Future Generation Computer Systems 105 (2020) 432–445

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A Blockchain Tokenizer for Industrial IOT trustless applications ☆

Daniele Mazzei [a,*], Giacomo Baldi [c], Gualtiero Fantoni [b], Gabriele Montelisciani [c], Antonio Pitasi [c], Laura Ricci [a], Lorenzo Rizzello [c]

[a] Department of Computer Science, University of Pisa, Italy
[b] Department of Civil and Industrial Engineering, University of Pisa, Italy
[c] Zerynth, USA

ARTICLE INFO

Article history:
Received 3 July 2019
Received in revised form 2 December 2019
Accepted 12 December 2019
Available online 16 December 2019

Keywords:
Industrial IOT
Blockchain
Ethereum
Smart contract
Supply chain
Industry 4.0

ABSTRACT

The Blockchain is a novel technology with a wide range of potential industrial applications. Despite a vast range of tests, prototypes, and proof of concepts implemented in the last years, the industrial use of Blockchain technology is still in the early stages. Enabling the interaction of industrial Internet of Things (IOT) platforms with Blockchain might be challenging because standards are still missing in both these technologies. Moreover, integrating productive assets with distributed data exchange and storage technologies is a kind of activity that needs to take into account various aspects, in particular: interoperability, portability, scalability, and security that need to be guaranteed by design.

This paper describes the implementation of a portable, platform-agnostic and secure Blockchain Tokenizer for Industrial IOT trustless applications. The Industrial Blockchain Tokenizer (IBT) is based on an industrial data acquisition unit able to gather data from both modern and legacy machines while also interfacing directly with sensors. Acquired data can be processed locally enabling an edge filtering paradigm and then sent to any Blockchain platform. The system has been designed, implemented and then tested on two supply chain scenarios. Tests demonstrated the system capability to act as a bridge between industrial assets and Blockchain platforms enabling the generation of immutable and trust-less "digital twins" for industrial IOT applications.

1. Introduction

The Blockchain is a novel technology with a wide range of potential industrial applications [1]. Inflated by the FinTech industry, then considered the holy grail of modern payment handling, BLOCKCHAIN is trying to move from FinTech to industrial applications [2,3]. Various research activities demonstrated the potential benefits of integrating Blockchain in business. Precision farming [4], robotic swarms for ocean bed exploration [5], smart grid for energy production [6], various healthcare platforms [7–9] and many other demonstrators have been implemented.

On the Information Technology (IT) side, various proof of concepts have been also built: Blockchain powered decentralised architectures for the management of Artificial Intelligence (AI) algorithms [1], protection of medical data by means of AI and Bitcoin [10], platform for deepfake videos identification [11], AI and Machine Learning (ML) algorithms marketplace [12], monetisation of IOT data [13], and many other [1,14].

However, despite the number of demos, prototypes and proof of concepts built in the last years, according to Deloitte's "Breaking Blockchain open" 2018 report [15], *the real industrial use of the Blockchain technology is still a chimaera.*

In this paper, the design, development, and test of a device aimed at connecting any industrial machine to any Blockchain platform are presented. This work aims to build an enabling technology for the diffusion of Blockchain in industrial applications thanks to the lowering of the development, integration and maintenance efforts of a Blockchain-powered Industrial Internet of Things (I-IOT) solution.

The paper is organised as follows: Firstly, Section 2 introduces the concept of IOT and analyses the academic activity and market trends on Blockchain-powered IOT solutions. Section 2.1 focuses on the Industrial IOT technology introducing the requirements and constraints used in Section 3 for the definition of a design guideline. Section 4 describes the design and development of the Blockchain tokenizer while the tests on two industrial supply-chain use cases are described in Sections 5.1 and 5.2. The reasons behind the choice of supply-chain as a testing domain are described in Section 5.

☆ Demo videos available on https://www.youtube.com/playlist?list=PLTTabSBmQk8s_eMRPCwZfvOIUKwu58yjP.

* Corresponding author.
E-mail address: daniele.mazzei@unipi.it (D. Mazzei).

https://doi.org/10.1016/j.future.2019.12.020
0167-739X/

2. Blockchain and IOT

The Internet of Things refers to the network of numerous physical objects provided with Internet connection [16]. Such devices continuously acquire information about the surrounding environment and exploit their connection to autonomously communicate with each other and with cloud servers using data aggregation units (gateway or edge-server).

The growing market of low costs micro-controllers (MCU) and micro-processors (MPU), their small footprint, their reduced cost and the availability of on-board antennas permit nowadays to make *smart* almost any of our classical devices (e.g. light bulbs, doors). For this reason, the amount of data produced by IOT devices is expected to grow exponentially in the following years, together with the associated market. Gartner forecasts [17] that the IOT market will quadruple in size, growing from $900 billion in 2014 to $4.3 trillion by 2024, with more than 30 billion connected objects.

The core concept at the basis of IOT and Industrial IOT technologies is the "Digital Twin" [18,19]. Grieves [20,21] firstly presented the concept of the digital twin as *"a digital replica of a living or non-living physical entity"*. By bridging the physical and the virtual world, data is transmitted seamlessly allowing the virtual entity to exist simultaneously with the physical entity. In industrial applications, this allows creating virtual models of physical assets, objects and goods to simulate real object's behaviour in real-world environments and applications [22].

The digital twin is composed of three components: the physical entities in the physical world, the virtual models in the virtual world, and the connected data that tie the two worlds [20]. Therefore, to enable the generation of a digital twin, it is necessary to interface physical assets with the digital world using sensors and interfaces able to acquire and convert real values in streams of data.

The growing of the IOT market also leads to another big challenge. The continuous gathering of personal and business-sensitive data by IOT objects and their visibility from the Internet put the problem of security as a prime concern. In this context, security has not only to be intended as the avoidance of physical or propriety damage (e.g., caused by the hacking of a smart alarm system) but it also has to be referred to the privacy issue [23].

On its "IOT signals 2019" report [24], Microsoft reported that 85% of respondents are in "IOT adoption" phase, and three-fourths of these have IOT projects in planning. Among IOT adopters, 88% believe IOT is critical to business success believing in a 30% ROI (Return of Investment) in two years. Nearly all IOT adopters (97%) have declared security concerns when implementing IOT while 38% of IOT adopters cite complexity and technical challenges in using IOT in industrial scenarios as a barrier to furthering adoption. Moreover, it is essential to focus that nearly one-third of projects (30%) implemented by respondents fail in the proof-of-concept stage, often because the implementation is expensive to scale (declared by 32%) or bottom-line benefits are unclear (28%).

Looking at this data, it is clear that a distributed trust technology, ensuring scalability, privacy, and reliability, for IOT platforms could be a cornerstone for the growth of a secure Industrial IOT environment [25,26].

In order to understand if and how the scientific community feels these needs, a corpus composed of Scopus retrieved papers' titles, and abstracts have been created and then analysed using Topic Modelling technique [27]. A query on Scopus[1] for `Blockchain AND (IOT OR Internet of Things)`, with a publication date within 2015 and 2020 has been executed resulting 1304 papers (see Fig. 1).

The result of the analysis are summarised in Tables 1 and 2.

Table 1
Most used words in "Blockchain AND IOT" papers' title.

1. Security	9. Energy
2. Technology	10. Supply
3. Privacy	11. Application
4. Management	12. Device
5. Architecture	13. Contract
6. Trust	14. Authentication
7. Control	15. Storage
8. Model	16. Software

Table 2
Labels extracted by the topic modelling algorithms.

1. Security and protection	6. Scalability
2. Privacy	7. Digital storage and cloud
3. Network architecture	8. Embedded devices requirements
4. Consensus and miners	9. Smart home and intelligent building
5. Communication protocol	10. Business and data market

Security, privacy and network architecture are the main reasons why scientists are working on the application of Blockchain technology to IOT and Industrial IOT architectures [28,29]. This demonstrates that the academic community correctly perceived current market needs and it is trying to address these challenges by porting Blockchain on the IOT world.

For example, in [30], authors present an architecture that integrates Software Defined Networking (SDN), Blockchain and fog computing. The SDN monitors and analyses traffic data in the IOT network to detect attacks, while Blockchain delivers decentralised attack detection. Fog supports attack detection and mitigation at the edge node. [31–33] present an approach for performing collaborative Deep Learning (DL) at the device level to overcome privacy leak and obtain enough data employing the Blockchain to ensure the confidentiality and integrity of DL in IOT.

2.1. Industrial IOT

In business, the final goal is to optimise production and increase efficiency. For this reason, many companies and factories are trying to monitor and analyse production facilities collecting data from industrial machines. Industrial IOT is the enabling technology of this new phenomenon known as *sensor-driven business* [34,35].

Industrial Internet of Things can radically change the industry as we know nowadays, but side effects cannot be underestimated. Security, trust and tampering will become major problems in the fourth and fifth industrial revolutions, and it is not trivial to solve them in a context where low powered and resource-constrained devices are ubiquitous. This lead to *trust* issues [36] that *Blockchain systems* are trying to deal with.

In the industrial context, security is addressed by the CIA triad (Confidentiality, Integrity, Authentication) [37] and by the enforcement of communication channels and storage. On the other hand, enforcing security by centralising the control and management of the system could lead to trust issues that arise when data is controlled by a single agent among several ones belonging to the same consortium or market [38].

While security solutions are already widely available and slowly (especially when it comes to constrained IOT devices) adopted [39,40], the extensive use of cloud-based centralised IOT platforms makes the building of a trusted environment very challenging. A trusted environment is a requirement for trustless applications: the use of the adjective trustless aims to underline how the trust should not be required towards a single element of the system to assure trust properties of the whole application.

[1] Query executed on September 9th 2019.

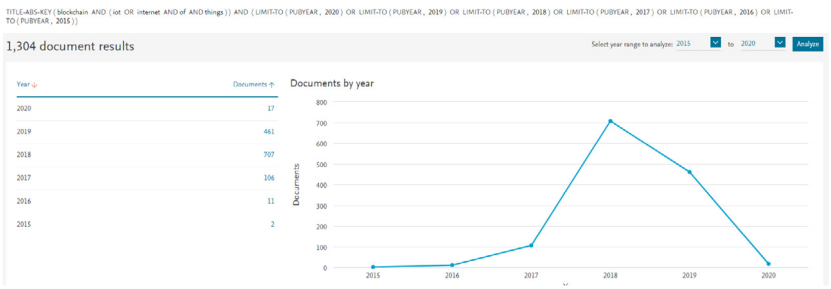

Fig. 1. Blockchain AND (IOT OR Internet of Things) query results on Scopus (9-9-2019).

Distributed ledgers [41] could represent a valuable resource in solving trust-related concerns since, by data sharing and consensus mechanisms enforced by cryptography, they create trusted setups in a trustless fashion [3,26,42].

Furthermore the latest trend of BaaS (Blockchain as a Service) platforms seem to offer the right compromise between the ease of use of cloud services and truly trusted setups, favouring the adoption of Blockchain solutions built by the more significant IT and cloud vendors, for example: AWS [43], Microsoft [44], IBM [45] and Oracle [46].

Looking at the scientific literature, we can assert that academics identified Blockchain as one of the most promising technologies to solve security and trust problems while guaranteeing scalability, privacy, and reliability in Industrial IOT applications [25,47,48].

Various pilots aimed at using Blockchain as IOT extending technology for industrial applications have been done. In [49] an authentication mechanism based on Blockchain-enabled fog nodes has been implemented using Ethereum smart contracts for the enforcement of IOT devices security. In [50] a decentralised Proof of Delivery (POD) solution for digital assets has been developed using Ethereum smart contracts to provide immutable and tamper-proof logs, accountability, and traceability thanks also to the integration with the InterPlanetary File System (IPFS) [51].

2.2. From IOT digital twin to Blockchain token

In the previous section, the IOT Digital Twin concept has been introduced like an in-silico copy of real-world assets. A similar concept is also present in the Blockchain world: the **"Token"**. A Token is the digital representation of an asset available on the Blockchain [52].

A Token is not necessarily related to a Cryptocurrency like Bitcoin or Ether and can be considered as a sort of security document. Technically, a Token is implemented by an algorithm defined in a *Smart Contract* on a Blockchain. For example, Ethereum [53,54] is a platform that can be used to create any arbitrary smart contract, including those that represent digital assets called **Ethereum tokens** [55]. Ethereum tokens can represent anything from a physical object like gold (Digix[2]) to a native currency used to pay transaction fees (Golem[3]). Tokens can be used for a variety of purposes such as paying to access a network or, like in our use-cases, tracking the status and modifying the value of goods and parcels.

Like in the case of the IOT Digital Twin, the link between a Token and its corresponding physical asset is initially purely fictitious. The Token contains the asset model that is populated with data by the algorithms implemented on the smart contract and thus firmly anchored. The algorithm of the smart contract guarantees the uniqueness of the data since it does not allow copies, substitution and could also limit the number of Tokens available on the Blockchain. **Tokenization** of real-world assets, is a trend that generated much interest over the past year [56]. Following the same paradigm, in this work, Ethereum tokens are used as a technology for the implementation of the digital twin for a trust-less Industrial IOT platform.

Following the approach taken in [57], we decided to test our technology in the context of the supply chain for manufacturing (The testing domain is described in Section 5). In Sections 5.1 and 5.2 two use cases are reported showing how Ethereum smart contract tokens have been used as digital twins of physical products (e.i. an item produced by an industrial machine and a parcel endowed with environmental parameters sensors).

3. Challenges and design guideline

Linking Blockchain platforms with industrial assets using IOT devices might lead to technological and cost obstacles. The biggest challenges to face for a Blockchain solution for Industrial applications are:

- interoperability
- portability
- scalability
- security

The following section reports an analysis of how these challenges emerge. The analysis is organised as a guideline for the design of a possible technological solution. The guideline is divided into sections and produces various Design Principles indicated with a capital DP, followed by a unique numeric ID.

3.1. Interoperability

The world of Industrial IOT devices is heterogeneous. PLCs have been the fundamental computational unit in factories for decades: equipped with proprietary, highly specialised software, they cannot be easily extended to support custom features nor replaced.

Embedded systems running bare-metal software, Real-Time Operating System (RTOS) or Linux based distribution are typically used as sensing and actuation nodes. Nodes are then connected to IOT gateways where data are aggregated and forwarded to cloud services or company servers.

Together with sensors node, nowadays, a wide range of automatic guided vehicle (AGV) and robotic arms are employed in factories and warehouses. This robotic systems typically run the new de-facto standard ROS [58,59] framework, a GNU/Linux based OS tailored for robotic applications.

In such a composite environment, any of these elements might need to be capable of tracking its actions on a Blockchain platform. For this reason, it is necessary to design an interoperable

2 https://www.dgx.io/.

3 https://golem.network/.

solution. In this context, interoperability means the possibility of connecting any industrial device to any kind of Blockchain able to support Tokens and Smart Contracts.

In software architecture design, this requirement is typically addressed by introducing an abstraction layer. Therefore, the designed solution must include an abstraction layer able to bridge any industrial apparatus with different smart contract enabled Blockchains **(DP1: Interoperability)**.

3.2. Portability

In Industrial contexts, each device and apparatus have to be strictly dedicated to the accomplishment of specific tasks. It is unacceptable to require the modification of industrial apparatus software in order to allow these units to perform Blockchain operations. Moreover, Blockchain operations and transactions can require specific protocols while being computational intense, thread blocking and time-consuming. All these Blockchain required features are almost impossible to enable in legacy industrial apparatus but also in most modern industrial machines.

For this reason, it is necessary to design a portable system aimed at decoupling Blockchain-related operations from apparatus functionalities and capabilities. In this context, portability is defined as the capability of the system to be plugged to any Industrial apparatus or device (also with constrained computational capabilities) enabling the execution of Blockchain-related operations using simple commands and with a fire-and-forget approach **(DP2: Portability)**.

3.3. Scalability

Scalability is defined as the capability of a system to properly handle a growing amount of work, to keep a stable level of efficiency in response to increased input. For a business, to cope with increased volume of orders without collapsing or limiting requests from clients. In this context, scalability is not addressed as a property of the industrial apparatus, but as the property of a generic factory to add Blockchain capabilities to its production line seamlessly.

In this context, scalability is addressed as a property of the industrial IOT platform that can be added to a generic industrial process seamlessly while guaranteeing the future capability of growing without requiring important intervention **(DP3: Scalability)**. It is essential to highlight that solution scalability cannot be addressed by increasing the computational load on the industrial apparatus because this will conflict with previously defined DR2.

Since Blockchain applications in the Industrial IOT world are still being defined, having a flexible solution to provide access to arbitrary smart contract enabled Blockchain platforms guaranteeing scalability represents a key advantage.

3.4. Security

A trusted system shall not compromise the security properties of the connected industrial apparatus. Transactions might be sent without encryption but signed to guarantee Integrity and Authentication, like what happens in the Ethereum public network [60]. Other Blockchain platforms, primarily the ones providing the possibility of deploying permissioned consortium instances, also allow to establish private channels to assure transactions confidentiality (i.e. Hyperledger Fabric).

In any case, a private key is needed to sign transactions and precautions must be taken for its secure storage and against its theft. Therefore, a reliable cryptographic stack is required for the execution of the security operations.

Two design principles for security can be derived: **(DP4: Industrial grade security)**: protect private keys at least following recommended Industrial IOT security standards; **(DP5: Built-in security)**: provide a built-in reliable cryptographic stack.

4. Building the Tokenizer

Five design principles for a technological solution aimed at bridging the gap between Blockchain and Industrial IOT have been derived from the analysis carried in Section 3. In this section, the Industrial Blockchain Tokenizer (IBT) development process is described starting from the selection of the hardware components and then discussing the software architecture design and implementation.

4.1. Hardware

The defined guideline oriented the authors towards the use of an external hardware solution aimed at guaranteeing the complete decoupling of Blockchain operations from the industrial apparatus routines and functions **(DP2: Portability)**. It would be possible, in modern machines controlled by an industrial PC to use a pure software solution to be installed utilising modern application containerisation techniques. However, it would be impossible to guarantee that failure and crashes of the containerised software will not affect the entire system operations. Moreover, the containerised software will use the computational resources of the host machine violating **DR3: Scalability**.

On the contrary, dedicated hardware peripheral would require commands from the industrial machine while returning responses through a standard interface. This condition makes mandatory the implementation of a standard, legacy-machine supportable, lightweight and scalable communication interface for the IBT **(DP1: Interoperability)**. The dedicated hardware should also be capable of taking care on-board of needed calculations, assuring reusability in different industrial contexts and scenarios while also guaranteeing scalability in term of the number of supported functions and target Blockchains **(DP3: Scalability)**.

The use of a dedicated hardware solution could reduce the scalability of the systems increasing linearly the cost required for the connection of new machines to the Blockchain service. However, due to the low price reached nowadays by IOT and I-IOT devices, authors believe that the above-described advantages widely overcome this limitation.

Finally, the IBT should seamlessly enable the tokenization of both industrial machines and goods **(DP3: Scalability)**. For this reason, the device needs to be endowed with digital interfaces for the connection with machines equipped with digital controllers but also with analogic ports aimed at acquiring data from sensors placed directly on goods, parcels and products in general.

An analysis of commercially available Industrial IOT gateway has been conducted. Nowadays the "Gateway" term is widely used in the IOT context making it difficult to identify a specific product category. The 4ZeroBox by TOI srl[4] has been chosen as a hardware unit for the development of the IBT. The 4ZeroBox is an IOT gateway specifically designed for retrofit applications with a focus on portability, interoperability and scalability.

In particular, **DP1: Interoperability** comports the need for supporting different communication protocols for the interfacing with industrial machines, sensors and Blockchain services. For the connection with industrial machines, the 4ZeroBox is equipped with the most popular industrial communication interfaces (RS232, RS485, Ethernet and CAN) and ready to use software

[4] www.thingsoninternet.it.

library for the most used protocols are also available (ModBus, EtherCat, CANBus, Siemens S7). The system also has 13 analogic inputs that enable the connection of 4–20 mA, 0–10 V, resistive, and current-coil sensors. On the Blockchain side, the IBG can communicate with remote servers through the 4ZeroBox built-in Ethernet and WiFi interfaces. The gateway supports the most common over TCP/IP protocols (HTTP, HTTPS, MQTT, MQTTS and other) for which software libraries are available. If needed, it is also possible to install a GSM expansion board.

A solution reliable for scalability **(DR3: Scalability)** need to be easy to install, fast to program, and easy to manage. The 4ZeroBox is easily programmable in C and Python languages thanks to the Zerynth platform[5] [61–64] (software details are discussed in Section 4.2). Moreover, well-documented programming SDK and toolchain are also available together with a mass programming toolkit and a certificate generation toolchain **(DR3: Scalability)**. Finally, the 4ZeroBox can be also linked to a dedicated cloud-based device manager from which it is possible to monitor the state of field-deployed devices and efficiently perform Firmware Over The Air (FOTA) updates and Remote Procedures Calling (RPC).

Furthermore, the 4ZeroBox is equipped with an onboard crypto chip (Microchip ATECC508A[6]), making it compliant with the most advanced I-IOT security standards. The ATECC508A crypto chip also supports Elliptic Curve cryptography [65] which is very common in the Blockchain world thus making the 4ZeroBox compliant with design requirements **(DP4: Industrial grade security)** and **(DP5: Built-in security)**. Section 4.2 details the use of ATECC508A crypto chip.

Regarding DP4 and DP5, it is also necessary to consider anti-tampering countermeasures. Tampering detection/response techniques are extensively documented [66,67] while tampering detection mechanisms are categorised into four groups: switches, sensors, circuitry and electronic [66] depending on the kind of tampering attack they aim at protecting. In general, the use of the analogic and digital I/O ports available on the 4Zerobox allows the implementation of anti-tampering mechanisms aimed at activating key erasure procedures when detecting unauthorised access [68]. However, these anti-tampering mechanisms are here identified, and their enforceability took into consideration as a requirement but not implemented in the IBT prototype described in this work.

It is important to highlight that most of the IOT gateways available on the market are based on Single Board Computer (SBC) endowed with Micro-Processor (MPU) that typically run custom Linux distributions. On the contrary, for security and power consumption reasons, we preferred to use a Micro-Controller (MCU) based solution where a minimal firmware runs on top of a Real-Time OS (RTOS). Due to the minimal nature of MCU firmware and to the architecture of real-time operating systems for MCU, MCU based embedded devices are less prone to cyber-attacks compared to mainstream computer systems like RaspberryPi and similar MCU based SBCs [69]. However, as the Stuxnet worm [70] proves, MCUs are not entirely safe from attacks but are still considered by the automotive and avionics industries safer than Linux based systems [71]. Another aspect that supports the use of MCU for critical applications is the recent release on the market of MCUs with support for hardware-implemented TrustZone memory. This secure memory is dedicated to the storage of application binaries and cannot be read from outside or modified if marked as "locked" [72]. Therefore, once a device has been programmed for a specific application and declared as locked, its firmware cannot be changed anymore, preventing software manipulation. Regarding power-consumption, MCUs are typically designed for low-power applications. Thus, they can be used in battery-operated scenarios for a very long time. Moreover, MCU hardware and RTOS architectures are designed for frequent hibernation allowing an easier design of low-power firmware for battery operated scenarios.

4.2. Software

Requirements R1, R2, R3 have been partially addressed by choosing a dedicated external unit aimed at taking care of all the Blockchain related operations using a generic interface with the industrial machine. Therefore, the IBT hardware will expose to industrial machines a simple command protocol that has to be accessible over a variety of interfaces enabling the execution of fire-and-forget Blockchain operations. Fig. 2 reports the IBT firmware architecture. The drawings highlight the various software elements implemented on the IBT firmware. Blue blocks represent software modules that are executed on the IBT MCU while in green is reported the cryptographic stack that is executed on the dedicated IBT crypto chip (see Section 4.1 for more detail). Blocks related to cloud or external servers services are reported in red. The device is connected to industrial machines equipped with digital interfaces through one of the communication interfaces supported by the 4ZeroBox. The *communication interface driver* takes care of handling the connection. The driver sends and receives data from the *command parser* module that can be considered as the communication abstraction layer of the IBT introduced in Section 3 and required by **(DP1: Interoperability)** and **(DP2: Portability)**. When using the IBT system for the monitoring of goods, parcels or legacy machines that do not support digital communication interfaces, a set of sensors can be plugged to these assets making the IBT able to acquire data related to the asset status. Data gathered by the sensors are acquired and filtered locally on the 4ZeroBox MCU using a dedicated software module. Data gathered by the command parser or by the sensors reading module are sent to the Blockchain layer where all the functions required for the enabling of Blockchain operations are implemented. The Blockchain layer communicates with the external crypto chip where the cryptographic stack handles all the required cryptographic operations. The IBT is connected with the external Blockchain server through a networking module that manages the Ethernet or Wifi connection. The networking module uses the external crypto chip for the management of the networking security keys and certificates.

4.2.1. Software Development Kit (SDK)

Development phase started with the choice of an SDK to program the IBT. The official 4ZeroBox SDK is Zerynth [61], a platform that allows programming in Python MCUs for IOT applications. Zerynth is a Python Virtual Machines (VM) that executes a bytecode compiled from Python by an external toolchain (Zerynth Studio). The Zerynth VM runs on top of a real-time operating system supporting multi-threaded programming. The Zerynth VM executes each Python thread as a dedicated thread of the RTOS making Zerynth allowing the building of multithreaded module-based software architecture.

Moreover, the Zerynth choice has also been motivated by the portable and high-level nature of the Zerynth Virtual Machine. A firmware written with Zerynth in hybrid C/Python can run virtually on any 32-bit microcontroller **(DP2: Portability)** giving us the possibility to migrate to other MCU based solutions different from the 4ZeroBox.

Zerynth SDK supports advanced networking and has a built-in cryptographic stack integrated seamlessly with hardware crypto elements like the ATECC508A for which libraries are available.

5 www.zerynth.com.

6 https://www.microchip.com/wwwproducts/en/ATECC508A.

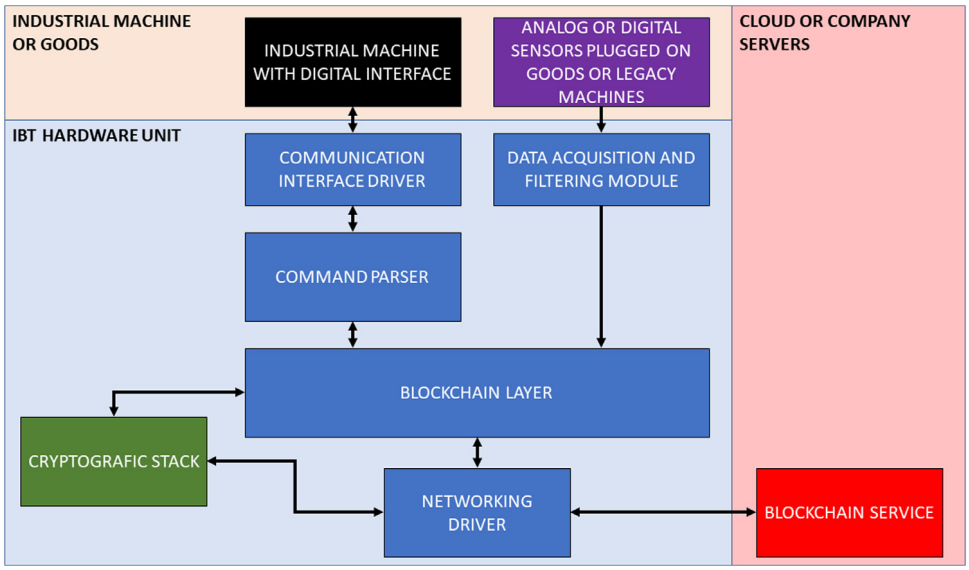

Fig. 2. Firmware architecture overview. Blue blocks represents software module that are executed on the IBT MCU while in green is reported the cryptographic stack that is executed on the dedicated IBT cryptochip. Blocks running on external servers are reported in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The Zerynth SDK supports all the communication interfaces available on the 4ZeroBox providing also high-level protocol libraries that have been used for the implementation of the interface drivers.

4.2.2. Hayes command set extended

For the building of the command parser, a command protocol has been defined and implemented.

A popular command set, used for decades in telecommunications and networking, is the Hayes protocol, also known as AT command set [73]. It has been introduced and used extensively with *modems*, `modulator/demodulator` devices in the classical sense, taking digital information and *modulating* it over different channels, and returning digital information *demodulated* from the external channel (typically analog channels). In the years, the concept of *modem* has gained full acceptance, even indicating communication devices used to link computers to other computers or networks. In this broader sense, the IBT could be considered as a sort of Blockchain modem. A modem that creates the link between an industrial asset and a Blockchain server producing the asset digital twin using the Blockchain token technology.

Hayes command set is composed of basic AT commands, command formats and result codes. It has been extended for custom solutions, the most popular of which is the one adopted for Global System for Mobile Communications (GSM) modems [74]. In this work, a custom set of AT commands for implementing Blockchain functionalities has been developed, making the IBT able to act as a Blockchain modem for industrial assets.

Table 3 reports the set of AT commands implemented.

In this work, AT commands have been specifically targeted to a Blockchain of the Ethereum family. Our choice is motivated by the fact that, currently, Ethereum is one of the few Blockchain offering a mature technology for the development of smart contracts. Other Blockchains, like Cardano[7] or Algorand,[8] are currently developing their smart contract system. It is worth noticing that our system will be able to include an interface with these smart contracts with minimal changes since this requires only a modification of the command parser module. (More details on the reasons behind the choice of Ethereum as a testing platform are reported in the following section.)

Table 3
Implemented Hayes command.

Command	Description
$AT + CADD =< contract\_address >$ $AT + CADD =?$	Set the address of a Contract Account. Get currently used contract address.
$AT + CGAS =< function\_name >$ $, < gas\_price >, < gas\_limit >$	Specify *gas_price* and *gas_limit* for function *function_name*
$AT + CGAS =< function\_name >$ $, ?, ?$	Get *gas_price* and *gas_limit* for *function_name*
$AT + CTX =< function\_name >, <$ $value >, < arg1 >, < arg2 >$ $, \ldots, < argN >$	Send a transaction to chosen Contract Address, calling *function_name* with a list of arguments and a value in ETH (can be [none]). $arg1, arg2, \ldots, argN$ are optional.
$AT + CCALL =< function\_name >$ $, < rx\_bits >, < rx\_type >, <$ $arg1 >, < arg2 >, \ldots, < argN >$	Call *function_name* not modifying the blockchain (not needing a paid transaction), to retrieve Contract info (e.g. variables state). Currently a single return value is supported and *rx_bits* is used to specify value length in bits, while *rx_type* specifies value type. Possible *rx_type* are *STRING*, *INTEGER*. $arg1, arg2, \ldots, argN$ are optional.
$AT + ETXCNT$	Get transactions count for current account.
$AT + ERPC =< rpc\_address >$	Set the node exposing the JSON ac{RPC} server to interact to.
$AT + ERPC =?$	Get the currently used address of the node.

4.2.3. Blockchain platform and interface

The Blockchain layer reported in Fig. 2 has been developed as an abstraction layer aimed at connecting the IBT with any Blockchain platform that supports smart contracts and tokens. For the implementation of the use case demo, various Blockchain platforms (IOTA, Hyperledger and Ethereum) have been analysed. IOTA is a crypto-currency emerged in 2015 intending to solve one

[7] https://www.cardano.org.
[8] https://www.algorand.com/.

of the Blockchain's main problems: **scalability**. Blockchain technologies like Bitcoin cannot expand quickly and will not be able to process high-speed transactions. IOTA addresses these issues with a new technology called *Tangle*. The tangle is a specialised database in which participants have to confirm two other transactions if they want to execute one. With this algorithm, IOTA can theoretically process an infinite amount of transactions and grows its network strength if more participants join [75]. Unfortunately, IOTA does not natively support smart contracts and, at the current stage, various discussion on the need of extending the platform with this feature is active on the IOTA community and within the board of directors [76]. Hyperledger is a consortium that incubates and promotes a range of business Blockchain technologies, including distributed ledger frameworks, smart contract engines, client libraries, graphical interfaces, utility libraries and sample applications. Different company and groups created their projects as customisation of the Hyperledger modular architecture. However, at the current stage, a general-purpose public project based on Hyperladger is not yet available [77]. On the other side, Ethereum [60] is considered the most prominent framework for smart contracts and its capitalisation reached 1 billion dollars since its launch in July 2015 [78]. Ethereum is available as public Blockchain; it has a free test network and can also be installed locally for test and development purposes. Moreover, a dedicated Zerynth library for interfacing with the Ethereum Blockchain is already available.[9] For this reason, Ethereum has been chosen as a platform for the implementation of the use cases.

However, it is important to highlight that several security vulnerabilities have been discovered in Ethereum smart contracts. These vulnerabilities have been identified with hands-on development experiences [79] and static analysis of the contracts available on the Ethereum Blockchain [80]. Ethereum has also suffered real attacks on smart contracts. The most successful attack managed to steal $60M from a contract [78]. This lead to the publication of various works aimed at mitigating vulnerabilities of Ethereum smart contract. In [51] an off-chain secure download phase based on the Interplanetary filesystem has been added to the Ethereum network obtaining proof of delivery mechanism.

The ATECC508A crypto element is one of the firsts crypto chip to integrate ECC (Elliptic Curve Cryptography) algorithms, but unfortunately, at the current stage, it only supports **secp256r1** curve while for signing Ethereum transaction the **secp256k1** [81] curve is required. For this reason, in this work, the crypto chip is used for the generation of the private key only while the transaction is signed on the IBT microcontroller using the Zerynth Ethereum library. IBT private key is generated using the ATECC508A crypto chip integrated high-quality FIPS 800-90 [82] Random Number Generator (RNG). The generated key is then stored in the crypto chip TrustZone memory and accessed by the Blockchain layer running on the microcontroller for the signing of Ethereum transactions.

Moreover, the ATECC580A features hardware monotonic counters that can be useful to further add security guarantees on the Ethereum transaction nonces. Authors are aware that, at the current stage, standard Ethereum implementation does not support permissioned Blockchain deployment that is attracting much attention in the industrial world. However, various parallel projects are addressing this issue by creating open-source Ethereum forks like the J.P. Morgan Quorum [83], that extends standard Ethereum with data privacy and consortium instances.

4.2.4. Smart contract

A general smart contract has been developed using Solidity,[10] the official Ethereum programming language. The contract acts as a storage for acquired data and enables the creation of the digital twin by producing the Blockchain token (limitation of this choice are discussed in 6). The contract has built-in rules that can be used according to the industrial scenarios for updating a shipment value or generating production alerts and alarms based on the tracked data and history.

The contract has been developed using the *Truffle suite*. *Truffle* includes a toolchain to compile Solidity code, deploy it on a Blockchain, run unit tests, and implements an interactive console to use the smart contract methods. The *Ganache* tool has been used for running locally an Ethereum debug Blockchain with auto mining capabilities.

Once deployed on a Blockchain, the public address of the deployed contract must be embedded in the IBT Blockchain layer firmware module together with its ABI (Application Binary Interface) which contains the signatures of the methods exposed. The contract does not call other contracts and has been built with simplicity in mind to avoid some common security pitfalls that may be used by an attacker to make unauthorised transactions [78]. The smart contract takes advantage of *events* for a more accessible development of front-end monitoring applications.

An arbitrary number of asset types can be configured in the contracts for the enabling of a different ruleset. In more details, a tracked item (e.i a parcel or any item produced by an industrial machine) is a *struct* composed of: a sender (Ethereum address), a receiver (Ethereum address) and (optionally) an initial value (integer, in Wei). The readings from the IBT *communication interface driver* or *data acquisition and filtering* modules are encoded in a single byte array in order to avoid hitting the maximum stack size of the EVM (Ethereum Virtual Machine).

The smart contract can be seen as a digitally signed certificate for the correct production or handling of the goods and as a receipt for the payment (when required). The contract also acts as a middle man for the payments enabling the receiver to pay with its Ethereum account.

The smart contract becomes an asset **(a token)** that includes a copy of the parcel/item data **(digital twin)** and also a digitalised story of the changes applied to the item value and properties by the rules defined in the contract **(a shipping certificate)**.

The following functions have been implemented on the contract:

- *create a good.* This function is called by the merchandise producer, specifying the consumer address, the current position, and the address of the next agent in charge of the transportation;
- *take a good.* This function is called by the next agent in charge of the transportation, also specifying the desired next taker in the chain;
- *give a good.* This function is called by the current parcel carrier when transferring good ownership to the desired taker;
- *retrieve the list of carriers for good.* This function returns the updated list of carriers so that, for example, after taking a good, an agent can be sure of transferred ownership by checking the last element of the list;
- *retrieve the list of transfer timestamps for good.* This function returns a list of transfer timestamps to check when each transfer, of a specific good, happened;
- *update good position.* This function is called by the current carrier to update transported good location.

[9] https://docs.zerynth.com/latest/official/lib.Blockchain.Ethereum/docs/index.html.

[10] https://github.com/Ethereum/solidity.

- *create shipment.* This function registers new shipment metadata (ID, sender address, receiver address, initial value) in the contract.
- *update shipment.* This function stores new data received from device sensors and update goods value accordingly.
- *get shipment data.* This function returns all the registered information, including the current value and the history of data logged from sensors.
- *complete shipment.* This function marks a shipment as complete and delivered. This function requires the caller to pay for the shipment. The amount is the last computed value.

Transactions can be executed by IBT authorised devices or by APPs associated with the operator's identity.

Moreover, the smart contract has been bundled with some unit tests ran within the Truffle suite before actually deploying it to a private Ethereum test Blockchain. The main functions of the contract have been tested against some expected results and different *fake* item generations have been produced simulating incoming data from the IBT in order to verify the smart contract functionalities and rule execution.

4.2.5. User interface and dashboard

A user interface generation backend has also been realised to report data published on the Blockchain by the IBT.

The interface has been built using Web3js[11] for calling smart contract methods from javascript, React[12] and Leaflet[13] with OpenStreetMap[14] for the development of the UI components including a tracking map when required by the use-case.

The backend allows the generation of applications that can be executed in any web browser, allowing the use of smartphone and tablet as monitoring devices. As a provider for the Ethereum RPC (Remote Procedure Call) interface Metamask[15] has been selected as the easiest way to call a smart contract method from inside a web-browser. Metamask can be installed as an extension for the most popular browsers: *Google Chrome*, *Mozilla Firefox*, and *Opera*, replacing the built-in Web3js engine and providing an improved user experience.

5. Test: Blockchain for distributed supply chain management

With big industrial and technological entities working together to achieve production-ready solutions, the Supply Chain field seems the best candidate for broad adoption of Blockchains. As defined by Swaminathan et al. in [84] a supply chain can be defined as *a network of autonomous or semi-autonomous business entities collectively responsible for procurement, manufacturing and distribution activities associated with one or more families of products*.

Thus, a supply chain can be considered as a multi-agent network. Therefore, the performance of each entity depends on the performance of others, on their operations and their coordination activities.

The need to track in real-time shipping goods among many distributors, both concerning goods conditions and passages of ownership, has forced the Supply Chain Management (SCM) sector to approach digital tracking solutions [85] which are slowly replacing paper-based transport bills [86]. However, Blockchain–SCM integration is still in its infancy [87] and the recent query `((Blockchain)AND(IOT OR (internet of things)) AND (logistics OR (supply chain)))` on Scopus produced only 40 results.[16]

On the other hand, data cannot be collected by the single entities participating in the chain and stored in private databases, without trust issues arising. Therefore, distributed ledgers aimed at creating immutable and shared records of every transaction associated with physical assets can enforce a new level of trust in the sector.

Following IBM analysis [88], Blockchain solutions would have a critical impact on three different trust-related areas:

- **Visibility**: Blockchains creates visibility in where things are, but also traceability in where things were. Even companies that have mastered visibility within their organisation find it difficult, today, to handle the massive amount of documents produced from dozens of vendors involved in the supply process. A seminal example is the wheat supply chain where everything needs to be tracked and reported at every passage, *from the farm to the shipping container to the factory floor to the loaf of bread on a shelf*. Thanks to improved visibility, managers could act and plan accordingly to the goods state in real-time. Another advantage carried by greater visibility is the possibility of reducing the absolute number of disputes and the time needed to resolve them. Eliminating the need for intermediaries aimed at creating trust among vendors would also lower the costs;
- **Optimisation**: optimisation comes as a consequence of real-time visibility. Being aware ahead of time of a supplier's shipment state would allow the organisation to take internal actions accordingly: steps to reshuffle internal inventory or complete the shipment from a different supplier in case of partial shipment, for example. An iterative optimisation process which becomes possible only thanks to data availability and reliability. Lowering the trust barrier would make it easier for companies to switch partners, making the market more dynamic;
- **demand**: with data about customer purchases available to every participant in the production and distribution network, every entity would be updated on real market demands. *Democratic* demand data could eradicate one of the most intractable challenges of the supply chain: the **bullwhip effect** [89] (Fig. 3), where orders are inflated as they move from final customer to manufacturer due for example to stock unit purchase.

For these reasons, we decided to focus on the Supply Chain domain where a good goes from a warehouse and reaches the final user after having passed through actors and processes that can modify its value. For making the explanation more precise and closer to real scenarios, two different use cases have been implemented: (1) internal logistic and (2) external logistic. (Videos of the use cases are available on https://www.youtube.com/playlist?list=PLTTabSBmQk8s_eMRPCwZfvOIUKwu58yjP.)

5.1. Use case 1: Robotized warehouse and transportation

A robotised warehouse application has been set up to test IBT capabilities on a production site (internal logistic). The internal logistic setup includes two different robots and a human operator, representing a plant where goods are produced and moved from a first production site to a second customisation line employing robotic arms, AGVs (Automatic Guided Vehicle) and human operators. This test aims to reproduce a critical step in supply chains: **the passage of ownership.**

11 https://web3js.readthedocs.io/en/1.0/.
12 https://reactjs.org/.
13 https://leafletjs.com/.
14 https://www.openstreetmap.org.
15 https://metamask.io/.

16 Query executed on July 1st 2019.

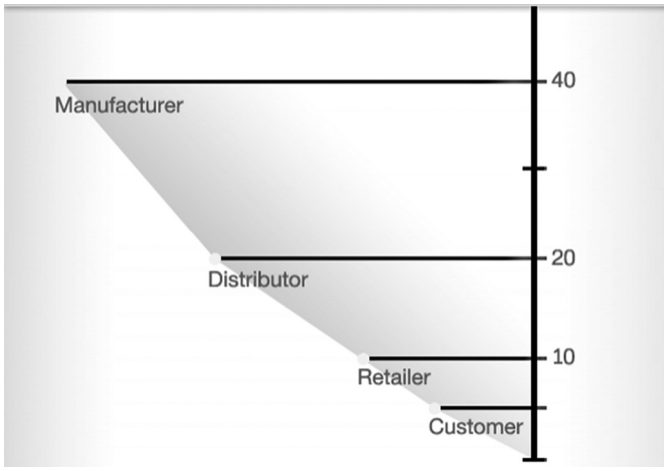

Fig. 3. Order inflation from customer to manufacturer [89].

In the internal logistic scenario, the good (the teddy bear in Fig. 4) is initially produced (and owned) by a KUKA manufacturing robotic arm (Kuka-LWR 4+[17]) ideally placed at the production plant. The good is then transferred to an autonomous vehicle (iRobot Roomba[18]) which must transport it to the customisation plant and, during transportation, update position tracking in real-time. At the second plant, a human operator takes ownership of the good to apply needed customisations and make it ready for final distribution. In this context, the Kuka arm can be identified as goods producer, the iRobot Roomba as goods carrier and the human operator as goods consumer.

(Demo video: https://youtu.be/58ZccdBBKoA)

5.1.1. Workflow and execution

Fig. 5 reports a detailed description of the internal logistic use case workflow from the perspective of each agent (columns). Blocks represent the agent's states that are reported in red when interaction with the Blockchain smart contract is required.

From time t_0 to time t_f, good ownership (blue circle in workflow blocks) passes from producer (Kuka arm) to consumer (human operator).

Reported workflow highlights the need of an identification mechanism to let each agent recognise the transaction partner and authorise the transfer process: as a consequence, each agent has a unique ID that is also printed on a QR code to be carried and to be visible for testing purposes.

In this scenario, the operator (henceforth O) is able to monitor, through a simple dashboard (henceforth D), all transportation phases here described:

– t_0 **(good creation):** Kuka arm (henceforth K) creates a good instance at the production site, the operator detects this action on the D which reports created goods ID and current carrier. iRobot (henceforth R) is also able, thanks to plugged IBT, to detect good creation and starts moving to the production site for taking the item.

– t_1 **(first identification):** R arrives at production site, identifies K thanks to its QR code and calls `GoodsTracker take` function for good 26. K completes the identification phase too and starts the physical transfer process.

– t_2 **(good transfer):** K completes the physical transfer phase and calls `GoodsTracker give` function, authorising the transfer of ownership performed by `GoodsTracker` Smart Contract function on the Blockchain platform.

– t_3 **(ownership transferred):** R detects ownership transfer and starts moving towards O site. D shows a change in ownership. Furthermore, the transfer table is updated with a transfer id, human-readable timestamp of the transfer and names of involved giver and taker.

– t_4 **(intermediate position update):** R stops at an intermediate position to update goods location coordinates: the change is immediately visible for O through D.

– t_5 **(second identification):** R arrives at O site and updates good location coordinates for the second time. It is immediately identified by O webcam and recognises O thanks to showed QR code, authorising the release of the goods calling `GoodsTracker give` function. A button, notifying carrier arrival, appears on D.

– t_6 **(good accepted by consumer):** O physically takes the good from R carrier and clicks on the button, triggering a `GoodsTracker take` function call. `GoodsTracker` contract performs the passage of ownership, which is shown on D.

Fig. 6 shows the user interface customised for this specific use-case. The interface allows real-time monitoring of the production and transportation of the goods while also enabling the human operator to interface with the Blockchain system accepting incoming goods.

5.2. Use case 2: Shipment condition monitoring and goods value real-time modification

In this use case, the IBT has been used as a sort of Blockchain-powered smart sensor for shipment state monitoring.

The IBT has been integrated for the purpose with two environmental parameter sensors and a GSM module in the 4ZeroBox (see Fig. 7). The 4ZeroBox has two expansion sockets based on the *MikroBUS* standard. For this use case the sockets have been used for the plugging of the following sensors:

- *BME280*, an integrated environmental unit capable of detecting humidity, pressure, and temperature;
- *LEA*-6S, a high performance GPS positioning chip;
- *ADXL345*, a 3-axis high-resolution accelerometer with ultra-low power consumption.

all the sensors mounted on the expansion boards communicate with the IBT microcontroller via I^2C protocol handled by the Zerynth dedicated libraries already available.

(Demo videos: https://youtu.be/KyCgE_xNwuY and https://youtu.be/kIjS6R5yIW4)

5.2.1. Workflow and execution

In Fig. 8, the external logistic workflow is illustrated from the point of view of the device, the sender, and the receiver of the shipment. As in case 1, each column represents an agent, red blocks indicate where that interaction with the Blockchain is required while blue circles indicate the ownership of the goods.

Also, in this scenario, the operator (O) can monitor, through the dashboard (D) all transportation phases here described:

– t_0 **(parcel creation):** The Operator creates a parcel using the dashboard (D) to insert the sender and receiver addresses together with the initial value of the parcel.

– t_1 **(IBT configuration):** The IBT device is notified by the backend on the creation of a new parcel to be tracked. Sensors data stored on the device buffer are cleaned.

— t_2 **(New Shipment Creation):** The IBT calls the Blockchain function for the creation of a new shipment passing the data of the parcel and the initial value of the sensor reads.

– t_3 **(Sensor Data Stream):** IBT device streams the acquired data to the backend with a frequency defined in the firmware. This frequency can also be exposed as a parameter in order to

17 Kuka LightWeight Robot 4+ https://www.kukakore.com/.

18 https://www.irobot.com/about-irobot/stem/create-2.

Fig. 4. The Kuka Arm releasing the produced item (teddy bear) on the iRobot Roomba autonomous vehicle. In overlay the operator monitoring dashboard.

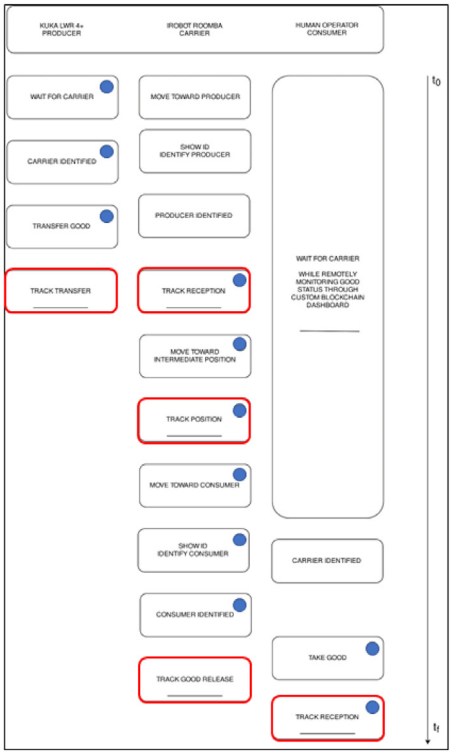

Fig. 5. Internal logistic workflow. Red blocks indicate where an interaction with the Ethereum smart contract is activated while blue circles indicate at each stage who is the agent owning the good. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

allow easy customisation of the parcel without requiring device firmware modification.

– t_4 **(Shipment Status Monitoring):** Both the sender and the receiver can monitor in real-time using the dashboard the status of the parcel visualising the current GPS position, the value in ETH, and the sensors data.

– t_5 **(Shipment delivered):** The receiver declares the parcel arrived and accepts to pay the final value in ETH in order to take the parcel and close the shipment. Events from the dashboard are sent to the backend. The smart contract is executed for the last time and permanently stored on the Blockchain.

– t_6 **(Device Stop and Reset):** The backend notifies the device that the shipment at which it was associated is completed. The sensors data stream is terminated, and the device set to the idle status.

Fig. 9 shows the user interface customised for this specific use-case. The interface allows real-time monitoring of the shipment position and status. Sensors data are reported in real-time together with the smart contract calculated parcel ETH value. The interface also allows the human operator to interact with the Blockchain system by accepting the incoming parcel, thus transferring (paying) the reported amount of ETH to the parcel sender address.

6. Costs and sustainability analysis

This work shall be considered a Proof of Concept (POC) of how Blockchain technology can be used in industrial IOT contexts for the monitoring of assets and goods in trust-less scenarios. In this preliminary version of the IBT, the smart contract permanently stores every data received from the robot/sensors/assets on the Blockchain. This is a big limitation leading to cost and scalability problems. In the reported use cases, the contract has been executed by a local Ethereum node, and the total gas cost for the shipment of the use case 2 has been calculated as reported in Fig. 10. The cost of the shipment is linearly dependent by the amount of data pushed from the IBT to the node. For example, sending a packet of data every 10 min for a week would consume about 150'000'000 gas units, which in June 2019 are worth $121.

This scalability and cost limitation can be reduced by changing the smart contract structure. For instance, having the possibility to write code to be executed directly on the IBT microcontroller, it is possible to move from a pushing strategy to an event-based paradigm where sensors data are analysed locally, and events only are sent to the Blockchain. This simple modification will dramatically drop the shipment cost of use case 2 to a few gas units without infringing the guideline and specification discussed above.

Another critical point is the number of transactions that a device execute for storing all the desired information and events. A possible solution is to classify locally on the IBT firmware events as critical/non-critical. In this case, non-critical events will be buffered on the device memory and sent in a single package for the execution of a new transaction only when a critical event is detected or at the closing of the shipment. The events classification strategy can be balanced according to the property and value of the goods in order to reach a sustainable shipment tracking cost.

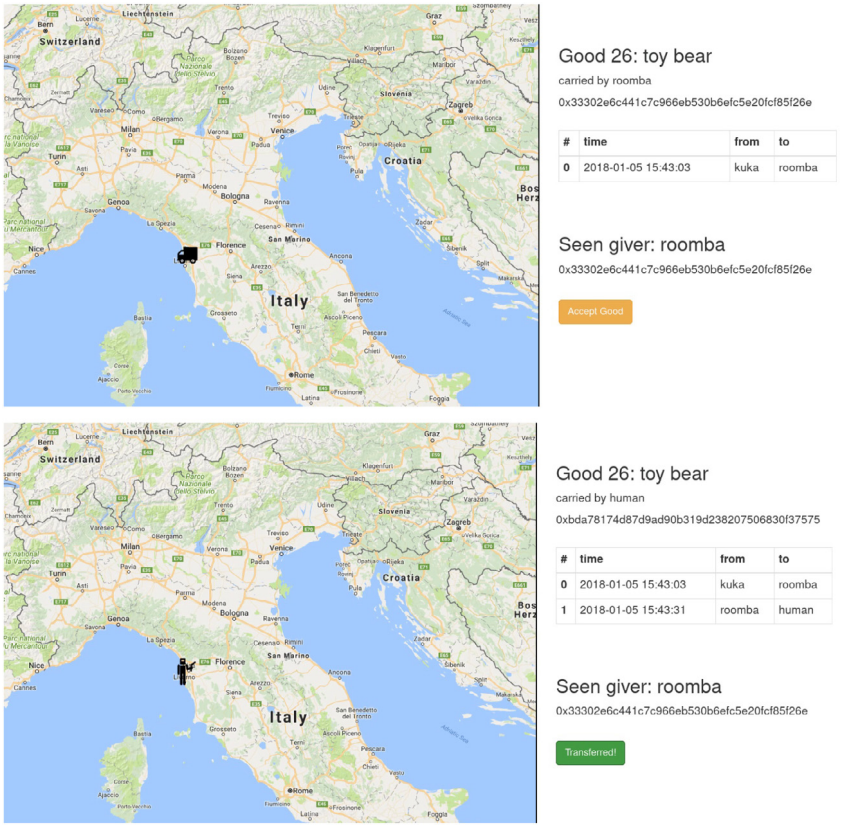

Fig. 6. Human operator dashboard to monitor goods production and accept incoming goods.

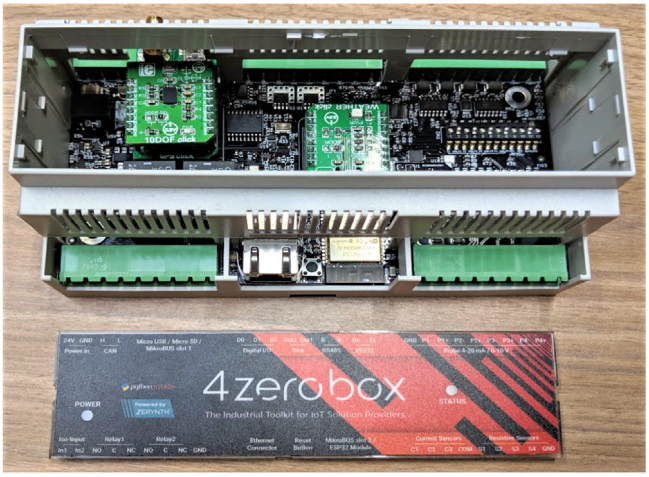

Fig. 7. 4ZeroBox expanded with GPS, accelerometer and temperature sensors via mikrobus click units (green boards).

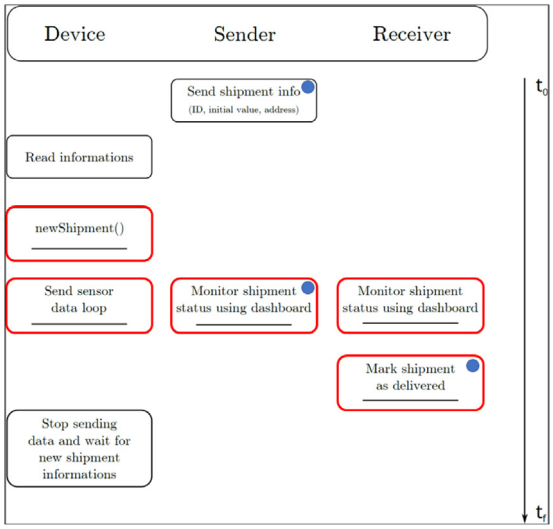

Fig. 8. External logistic use case workflow. Red blocks indicate where an interaction with the Ethereum smart contract is activated while blue circles indicate at each stage who is the agent owning the good. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Introducing this modification does not affect the scalability of the system. Rules and parameters can be written on a smart contract and then transferred as configuration to the IBT when a new parcel or good is created.

7. Conclusion

In this paper, a solution aimed at interfacing industrial assets and apparatus with Blockchains for industrial trustless applications has been designed, implemented and tested. The proposed solution is completely agnostic, allowing the interaction of modern and legacy machines with public, on-premises or as a service Blockchains. Moreover, the proposed solution can be easily customised to become a standalone Blockchain-powered tracking system. In the design process, a guideline has been defined and then used for the driving of the device hardware and software selection and implementation.

The system has been designed to act as a bridge between the Industrial IOT, and the Blockchain world enabling the tokenisation of industrial assets, and the creation of Blockchain-powered

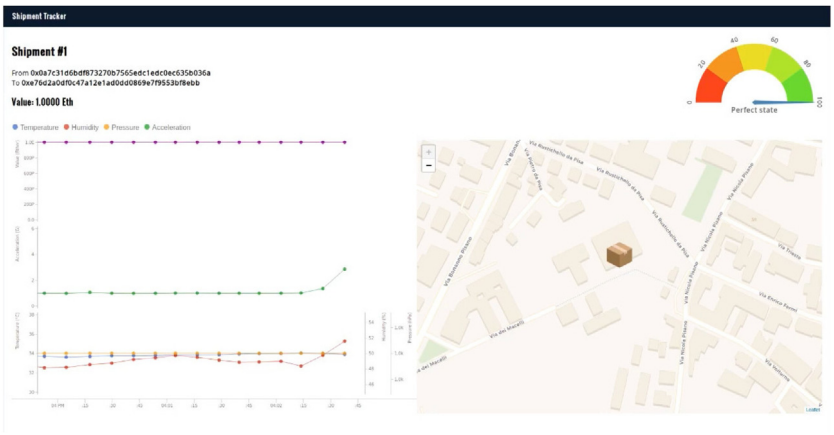

Fig. 9. The shipment monitor interface with real-time report of the parcel value in ETH.

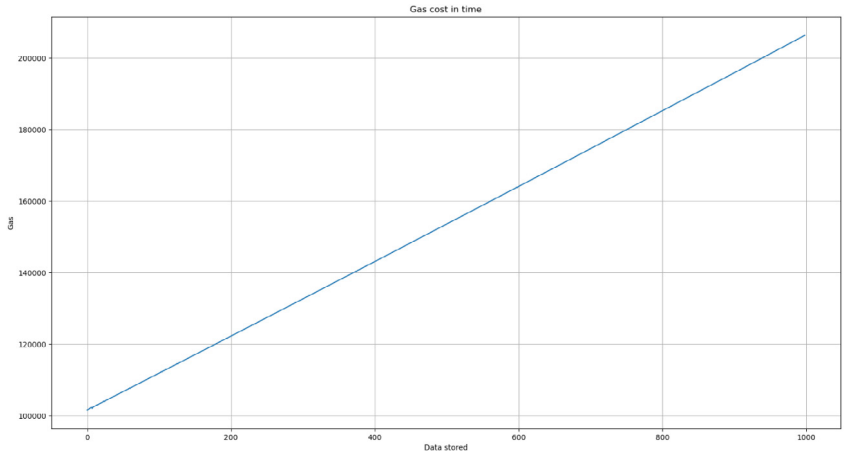

Fig. 10. GAS usage of a shipment tracked with the device customised for use case 2.

digital twins. The IBT can be considered as a reference design for the building of Blockchain-powered end-to-end Industrial IOT solutions that cover the entire data flow going from the machine or parcel parameter tracking to the operator monitoring dashboard. The implemented cases allowed testing the IBT plugged to different industrial apparatus and sensors. The system can add Blockchain functionalities to modern industrial scenarios where robotised machines are available but also to external scenarios where parcels and goods are far from industrial apparatus and so, difficult to be tracked and monitored. The integration process turned out almost effortless thanks to the choice of commercially available and well-documented hardware and software solutions. Moreover, the definition of an ad-hoc Hayes command set made the interfacing with industrial machines effortless. Human-centred design is mandatory in the innovation of industrial scenarios and processes. For this reason, a dedicated dashboard generation engine has also been developed as a component of the IBT system.

In the presented use cases all the computing effort has been executed by the IBT computational unit, both concerning cryptographic functions and networking, allowing the Industrial IOT assets and sensors to focus on its primary tasks so that also industrial security and scalability requirements are satisfied.

The box has also proven to be reliable for repeated use, but, for being declared production-ready, it should undergo batteries of tests that are beyond the scope of this work, same considerations are valid for industrial certifications.

The crypto element embedded on the 4ZeroBox has been used to store private keys and sign Ethereum transactions fully accomplishing the security defined requirements. The GAS estimation for the execution of the smart contract has been calculated, and various cost reduction strategies have been identified.

Future development is needed to extend the Blockchain layer adding other Blockchain platforms. Anyhow, this passage will be entirely painless for the user thanks to the implemented abstraction layers.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Daniele Mazzei, Giacomo Baldi, Gualtiero Fantoni and Gabriele Montelisciani are co-founders of the Zerynth company. Lorenzo Rizzello and Antonio Pitasi are employees of the Zerynth company. Laura Ricci has no relationship with Zerynth.

Zerynth contributed with its employees and assets to the work reported in this paper.

References

[1] K. Salah, M.H.U. Rehman, N. Nizamuddin, A. Al-Fuqaha, Blockchain for ai: review and open research challenges, IEEE Access 7 (2019) 10127–10149.
[2] D. Tapscott, A. Tapscott, Blockchain Revolution: How the Technology behind Bitcoin Is Changing Money, Business, and the World, Penguin, 2016.

[3] A. Reyna, C. Martín, J. Chen, E. Soler, M. Díaz, On blockchain and its integration with iot, challenges and opportunities, Future Gener. Comput. Syst. 88 (2018) 173–190, http://dx.doi.org/10.1016/j.future.2018.05.046, URL http://www.sciencedirect.com/science/article/pii/S0167739X17329205.

[4] J. Lin, Z. Shen, A. Zhang, Y. Chai, Blockchain and iot based food traceability for smart agriculture, in: Proceedings of the 3rd International Conference on Crowd Science and Engineering, ACM, 2018, p. 3.

[5] E.C. Ferrer, The blockchain: a new framework for robotic swarm systems, in: Proceedings of the Future Technologies Conference, Springer, 2018, pp. 1037–1058.

[6] M. Mylrea, S.N.G. Gourisetti, Blockchain for smart grid resilience: Exchanging distributed energy at speed, scale and security, in: 2017 Resilience Week (RWS), IEEE, 2017, pp. 18–23.

[7] M. Hölbl, M. Kompara, A. Kamišalić, L. Nemec Zlatolas, A systematic review of the use of blockchain in healthcare, Symmetry 10 (10) (2018) 470.

[8] K. Peterson, R. Deeduvanu, P. Kanjamala, K. Boles, A blockchain-based approach to health information exchange networks, 2016.

[9] P. Mamoshina, L. Ojomoko, Y. Yanovich, A. Ostrovski, A. Botezatu, P. Prikhodko, E. Izumchenko, A. Aliper, K. Romantsov, A. Zhebrak, et al., Converging blockchain and next-generation artificial intelligence technologies to decentralize and accelerate biomedical research and healthcare, Oncotarget 9 (5) (2018) 5665.

[10] A. Maxmen, Ai researchers embrace bitcoin technology to share medical data, Nature 555 (7696) (2018).

[11] H.R. Hasan, K. Salah, Combating deepfake videos using blockchain and smart contracts, IEEE Access 7 (2019) 41596–41606.

[12] Singularitynet, 2018, URL https://singularitynet.io/.

[13] A. Suliman, Z. Husain, M. Abououf, M. Alblooshi, K. Salah, Monetization of iot data using smart contracts, IET Netw. 8 (1) (2018) 32–37.

[14] M.A. Khan, K. Salah, Iot security: Review, blockchain solutions, and open challenges, Future Gener. Comput. Syst. 82 (2018) 395–411.

[15] Breaking blockchain open: Deloitte's 2018 global blockchain survey, 2018, URL https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-fsi-2018-global-blockchain-survey-report.pdf.

[16] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Comput. Netw. 54 (15) (2010) 2787–2805, http://dx.doi.org/10.1016/j.comnet.2010.05.010, URL http://www.sciencedirect.com/science/article/pii/S1389128610001568.

[17] V. Liu, Business benefits of the internet of things: A gartner trend insight report, 2019, URL https://www.gartner.com/en/doc/3806366-business-benefits-of-the-internet-of-things-a-gartner-trend-insight-report.

[18] E. Negri, L. Fumagalli, M. Macchi, A review of the roles of digital twin in cps-based production systems, Procedia Manuf. 11 (2017) 939–948.

[19] Q. Qi, F. Tao, Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison, IEEE Access 6 (2018) 3585–3593.

[20] M. Grieves, Digital twin: Manufacturing excellence through virtual factory replication, White paper, 2014, pp. 1–7.

[21] A. El Saddik, Digital twins: The convergence of multimedia technologies, IEEE MultiMedia 25 (2) (2018) 87–92, http://dx.doi.org/10.1109/MMUL.2018.023121167.

[22] J. Hochhalter, W.P. Leser, J.A. Newman, V.K. Gupta, V. Yamakov, S.R. Cornell, S.A. Willard, G. Heber, Coupling damage-sensing particles to the digitial twin concept.

[23] R. Roman, J. Zhou, J. Lopez, On the features and challenges of security and privacy in distributed internet of things, Comput. Netw. 57 (10) (2013) 2266–2279, http://dx.doi.org/10.1016/j.comnet.2012.12.018, towards a Science of Cyber Security Security and Identity Architecture for the Future Internet. URL http://www.sciencedirect.com/science/article/pii/S1389128613000054.

[24] Iot signals: Summary of research learnings, 2019, URL https://azure.microsoft.com/mediah{and}ler/files/resourcefiles/iot-signals/IoT-Signals-Microsoft-072019.pdf.

[25] A. Panarello, N. Tapas, G. Merlino, F. Longo, A. Puliafito, Blockchain and iot integration: A systematic survey, in: Sensors, 2018.

[26] N.M. Kumar, P.K. Mallick, Blockchain technology for security issues and challenges in iot, Procedia Comput. Sci. 132 (2018) 1815–1823, http://dx.doi.org/10.1016/j.procs.2018.05.140, international Conference on Computational Intelligence and Data Science. URL http://www.sciencedirect.com/science/article/pii/S187705091830872X.

[27] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.

[28] S. Huckle, R. Bhattacharya, M. White, N. Beloff, Internet.of. things, Internet of things blockchain and shared economy applications, Procedia Comput. Sci. 98 (2016) 461–466.

[29] R. Shrestha, S. Kim, Integration of iot with blockchain and homomorphic encryption: Challenging issues and opportunities, Role Blockchain Technol. IoT Appl. 115 (2019) 293.

[30] S. Rathore, B.W. Kwon, J.H. Park, Blockseciotnet: Blockchain-based decentralized security architecture for iot network, J. Netw. Comput. Appl. 143 (2019) 167–177.

[31] S. Rathore, Y. Pan, J.H. Park, Blockdeepnet: A blockchain-based secure deep learning for iot network, Sustainability 11 (14) (2019) 3974.

[32] S. Rathore, J.H. Park, Semi-supervised learning based distributed attack detection framework for iot, Appl. Soft Comput. 72 (2018) 79–89.

[33] S. Rathore, P.K. Sharma, A.K. Sangaiah, J.J. Park, A hesitant fuzzy based security approach for fog and mobile-edge computing, IEEE Access 6 (2017) 688–701.

[34] D. Mourtzis, E. Vlachou, N. Milas, Industrial big data as a result of iot adoption in manufacturing, Procedia Cirp 55 (2016) 290–295.

[35] C. Perera, C.H. Liu, S. Jayawardena, The emerging internet of things marketplace from an industrial perspective: A survey, IEEE Trans. Emerg. Top. Comput. 3 (4) (2015) 585–598.

[36] O. Salman, I. Elhajj, A. Kayssi, A. Chehab, An architecture for the internet of things with decentralized data and centralized control, in: Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of, IEEE, 2015, pp. 1–8.

[37] D.B. Parker, Information security in a nutshell, Inf. Syst. Secur. 6 (1) (1997) 14–19.

[38] R. von Solms, J. van Niekerk, From information security to cyber security, Comput. Secur. 38 (2013) 97–102, http://dx.doi.org/10.1016/j.cose.2013.04.004, cybercrime in the Digital Economy. URL http://www.sciencedirect.com/science/article/pii/S0167404813000801.

[39] H. He, C. Maple, T. Watson, A. Tiwari, J. Mehnen, Y. Jin, B. Gabrys, The security challenges in the iot enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence, in: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 1015–1021.

[40] A.-R. Sadeghi, C. Wachsmann, M. Waidner, Security and privacy challenges in industrial internet of things, in: 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), IEEE, 2015, pp. 1–6.

[41] F.M. Benčić, I.P. Žarko, Distributed ledger technology: blockchain compared to directed acyclic graph, in: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2018, pp. 1569–1570.

[42] A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, Blockchain for iot security and privacy: The case study of a smart home, in: Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on, IEEE, 2017, pp. 618–623.

[43] Aws baas URL https://aws.amazon.com/blockchain/.

[44] Microsoft azure baas. URL https://azure.microsoft.com/en-us/solutions/blockchain/.

[45] Ibm baas. URL https://www.ibm.com/blockchain/.

[46] Oracle baas. URL https://www.oracle.com/cloud/blockchain/index.html.

[47] K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the internet of things, IEEE Access 4 (2016) 2292–2303.

[48] H. Malviya, How blockchain will defend iot, Available at SSRN 2883711.

[49] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, K. Salah, A user authentication scheme of iot devices using blockchain-enabled fog nodes, in: 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), IEEE, 2018, http://dx.doi.org/10.1109/aiccsa.2018.8612856.

[50] H.R. Hasan, K. Salah, Proof of delivery of digital assets using blockchain and smart contracts, IEEE Access 6 (2018) 65439–65448.

[51] J. Benet, Ipfs-content addressed, versioned, p2p file system, arXiv preprint arXiv:1407.3561.

[52] T. Weingärtner, Tokenization of physical assets and the impact of iot and ai, in: eublockchainforum.eu.

[53] G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger, Ethereum Proj. Yellow Pap. 151 (2014) (2014) 1–32.

[54] Ethereum wiki. URL https://github.com/ethereum/wiki/wiki.

[55] L. Xie, A beginner's guide to ethereum tokens, 2018, URL https://blog.coinbase.com/a-beginners-guide-to-ethereum-tokens-fbd5611fe30b.

[56] B. Garner, Physical assets on the blockchain: Why bother?, 2018, URL https://coincentral.com/physical-assets-on-the-blockchain-why-bother/.

[57] M. Westerkamp, F. Victor, A. Küpper, Tracing manufacturing processes using blockchain-based token compositions, Digit. Commun. Netw. (2019) http://dx.doi.org/10.1016/j.dcan.2019.01.007.

[58] A. Koubâa, Robot Operating System (ROS), Springer, 2017.

[59] J. Kerr, K. Nickels, Robot operating systems: Bridging the gap between human and robot, in: Proceedings of the 2012 44th Southeastern Symposium on System Theory (SSST), 2012, pp. 99–104. http://dx.doi.org/10.1109/SSST.2012.6195127.

[60] V. Buterin, et al., Ethereum white paper: a next generation smart contract & decentralized application platform, First version.

[61] D. Mazzei, G. Baldi, G. Montelisciani, G. Fantoni, A full stack for quick prototyping of iot solutions, Ann. Telecommun. 73 (7–8) (2018) 439–449.

[62] D. Mazzei, G. Montelisciani, G. Baldi, A. Baù, M. Cipriani, G. Fantoni, Improving the efficiency of industrial processes with a plug and play iot data acquisition platform, Enterp. Interoper.: Smart Serv. Bus. Impact Enterp. Interoper. (2018) 315–321.

[63] P. Karvelis, T.-A. Michail, D. Mazzei, S. Petsios, A. Bau, G. Montelisciani, C. Stylios, Adopting and embedding machine learning algorithms in microcontroller for weather prediction, in: 2018 International Conference on Intelligent Systems (IS), IEEE, 2018, pp. 474–478.

[64] D. Mazzei, G. Montelisciani, G. Baldi, G. Fantoni, Changing the programming paradigm for the embedded in the iot domain, in: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), IEEE, 2015, pp. 239–244.
[65] D. Hankerson, A. Menezes, Elliptic Curve Cryptography, Springer, 2011.
[66] M. Aarts, Hardware attacks tamper resistance, tamper response and tamper evidence, 2016, Date of retrieval 23.
[67] S.H. Weingart, Physical security devices for computer subsystems: A survey of attacks and defenses, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2000, pp. 302–317.
[68] Mcu tamper protection. URL https://www.digikey.com/en/articles/techzone/2015/aug/tamper-protection-secures-your-valuable-mcu-based-system-ip.
[69] Built-in rtos security for connected embedded devices, 2018, URL https://www.intervalzero.com/embedded/built-in-rtos-security-for-connected-embedded-device/.
[70] S. Karnouskos, Stuxnet worm impact on industrial cyber-physical system security, in: IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2011, pp. 4490–4494.
[71] A. Ukil, J. Sen, S. Koilakonda, Embedded security for internet of things, in: 2011 2nd National Conference on Emerging Trends and Applications in Computer Science, IEEE, 2011, pp. 1–6.
[72] S. Ravi, A. Raghunathan, S. Chakradhar, Tamper resistance mechanisms for secure embedded systems, in: 17th International Conference on VLSI Design. Proceedings, IEEE, 2004, pp. 605–611.
[73] D.S. Alberts, R.E. Hayes, Power to the Edge: Command. Control. in the Information Age, Tech. rep., 2003.
[74] At command set for user equipment, Tech. rep. URL http://www.etsi.org/deliver/etsi_ts/127000_127099/127007/10.03.00_60/ts_127007v100300p.pdf.
[75] R. Alexander, Iota-introduction to the tangle technology: Everything you need to know about the revolutionary blockchain alternative.
[76] R. Ralf, Smart contracts and iota. URL https://medium.com/@ralf/about-smart-contracts-in-iota-626d2bd3619e.
[77] C. Cachin, Architecture of the hyperledger blockchain fabric, in: Workshop on Distributed Cryptocurrencies and Consensus Ledgers, Vol. 310, 2016, p. 4.
[78] N. Atzei, M. Bartoletti, T. Cimoli, A survey of attacks on ethereum smart contracts (sok), in: Principles of Security and Trust, Springer, 2017, pp. 164–186.
[79] K. Delmolino, M. Arnett, A. Kosba, A. Miller, E. Shi, Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab, in: International Conference on Financial Cryptography and Data Security, Springer, 2016, pp. 79–94.
[80] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, A. Hobor, Making smart contracts smarter, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 254–269.
[81] J. Bauer, R.C. Staudemeyer, H.C. Pöhls, A. Fragkiadakis, Ecdsa on things: Iot integrity protection in practise, in: International Conference on Information and Communications Security, Springer, 2016, pp. 3–17.
[82] Q.H. Dang, Sp 800-106. randomized hashing for digital signatures.
[83] Jpmorgan quorum distributed ledger. URL https://github.com/jpmorganchase/quorum.
[84] J.M. Swaminathan, S.F. Smith, N.M. Sadeh, Modeling supply chain dynamics: A multiagent approach*, Decis. Sci. 29 (3) (1998) 607–632, http://dx.doi.org/10.1111/j.1540-5915.1998.tb01356.x, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-5915.1998.tb01356.x.
[85] Digital transformations of industies - logistics industry, Tech. Rep., 2016.
[86] A. Rushton, P. Croucher, P. Baker, The Handbook of Logistics and Distribution Management, fourth ed..
[87] M.M. Queiroz, R. Telles, S.H. Bonilla, Blockchain and supply chain management integration: a systematic review of the literature, Supply Chain Manag.: Int. J. (2019).
[88] Trust in trade - toward stronger supply chains, Tech. Rep., 2016.
[89] What is the bullwhip effect? understanding the concept & definition, 2012, URL http://www.aalhysterforklifts.com.au/index.php/about/blog-post/what_is_the_bullwhip_effect_underst{and}ing_the_concept_definition.

Daniele Mazzei Assistant Professor at the Computer Science Department of The University of Pisa. Daniele graduates in Biomedical Engineering in 2006 and PhD in May 2010 in Automatic Robotic and Bioengineering at the University of Pisa. His research focuses on the Industrial Internet of Things and human-robot/machine interaction. He is co-inventor of 8 patents, co-author of 52 Scopus indexed publications with 747 citations and h-index of 15 and also co-author of 2 book chapters. He is a co-founder of Zerynth

Giacomo Baldi He holds two master degrees, one in computer science and one in medicine. Most of his experiences of work and research stemmed from the ability to mix both fields of expertise in innovative ways. As a computer scientist in 2015 he co-founded and developed Zerynth, a Python virtual machine for embedded devices. In December 2012 with other colleagues, he also co-founded the Pisa FABLAB. He also developed in 2011 an acquisition and analysis software for a wearable electrical impedance tomography device. Giacomo is co-inventor of three patents.

Gualtiero Fantoni Associate Professor at University of Pisa, President of the Pisa Leaning Lab, co-founder of Fablab Pisa and some companies (one university spin-off included). MEng in Mechanical Engineering "summa cum laude" and PhD in Robotics, Automation and Bioengineering at the University of Pisa. Principal investigator of several EU, National and Regional research projects. His research interests are in all kinds of automation: from hardware such as industrial grippers, handling and feeding systems to software such as natural language processing tools for document analysis, patent analysis, chatbot, etc.. His publications include more than 100 peer-reviewed papers, co-inventor of 10 patents.

Gabriele Montelisciani Ph.D in Economics and Management Engineering. Co-founder and CEO at Zerynth. Former Research Associate at the University of Pisa, focusing on methods and tools for early-stage innovation, new business development and value creation from ICT innovation. Project Manager for the project ENDuRE - European Network of Design for Resilient Entrepreneurship (EU Erasmus+ Knowledge Alliance). Research experience in more than 10 EU and National funded projects. Member of the International Review Committee of the International Conference on Engineering, Technology and Innovation (ICE Conference). Author of more than 15 publications and two patents.

Antonio Pitasi Fullstack Software Developer at Zerynth. He graduated in 2019 in Computer Science at the University of Pisa with a thesis on the application of Ethereum blockchain for logistics applications.

Laura Ricci Laura Ricci received the PhD from the University of Pisa, in 1990. She is currently Associate Professor at the textquotedblDepartment of Computer Sciencetextquotedbl, textquotedblUniversity of Pisa, Italy. Her research interests include distributed systems, peer-to-peer networks, cryptocurrencies and blockchains and social network analysis. In these fields, she has co-authored over 150 papers published on international journals and conference/workshop proceedings. She has served as program committee member and chair of several conferences. She has been involved in several research projects and is currently the local coordinator of the H2020 European Project textquotedblHelios: A context-aware Distributed Networking Framework.

Lorenzo Rizzello Embedded Software Engineer at Zerynth. Lorenzo Graduates in Robotics and Automation Engineering in 2018 at the University of Pisa with a thesis investigating a possible role for Blockchain technology in the Robotics field.