**University of
Zurich^{UZH}**

**Zurich Open Repository and
Archive**

University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2020

BIIT: Standardization of Blockchain-based I2oT Systems in the I4 Era

Niya, Sina Rafati ; Schiller, Eryk ; Cepilov, Ile ; Stiller, Burkhard

Abstract: In Industry 4.0 (I4), the Industrial Internet of Things (I2oT) data streams are prone to significant data manipulation risks. The integration of Blockchains (BC) with I2oT may become a solution preventing from this problem. This paper provides a blockchain-agnostic Blockchain I2oT (BI2oT) architecture called BIIT that allows developing a broad range of BC applications fully integrating Internet of Things (IoT). The mechanisms introduced in BIIT aim at solutions that provide data reliability, limit the computational overhead, and enhance energy efficiency. BIIT is evaluated through real-world experimentation.

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-187940>

Conference or Workshop Item

Accepted Version

Originally published at:

Niya, Sina Rafati; Schiller, Eryk; Cepilov, Ile; Stiller, Burkhard (2020). BIIT: Standardization of Blockchain-based I2oT Systems in the I4 Era. In: IEEE/IFIP Network Operations and Management Symposium (NOMS), Budapest, 20 April 2020 - 24 April 2020. IEEE, 1-9.

BIIT: Standardization of Blockchain-based I²oT Systems in the I4 Era

Sina Rafati Niya, Eryk Schiller, Ile Cepilov, Burkhard Stiller
Communication Systems Group CSG@IfI, University of Zürich
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
Emails: [rafati|schiller|stiller]@ifi.uzh.ch, ile.cepilov@uzh.ch

Abstract—In Industry 4.0 (I4), the Industrial Internet of Things (I²oT) data streams are prone to significant data manipulation risks. The integration of Blockchains (BC) with I²oT may become a solution preventing from this problem. This paper provides a blockchain-agnostic Blockchain I²oT (BI²oT) architecture called BIIT that allows developing a broad range of BC applications fully integrating Internet of Things (IoT). The mechanisms introduced in BIIT aim at solutions that provide data reliability, limit the computational overhead, and enhance energy efficiency. BIIT is evaluated through real-world experimentation.

I. INTRODUCTION

The Internet of Things (IoT) provides important characteristics including certain level of intelligence allowing for automation of data collection and actuation that does not require continuous human intervention. Thus, collected data volumes, measurement resolution, and coverage may be enhanced in comparison to current manufacturing processes. In Industrial Internet of Things (I²oT), end devices continuously record data about the environment and provide it for further processing. The knowledge derived from collected information allows for optimizations of industrial processes, thus, proving a great value for the industry.

The variation of IoT technologies created a broad heterogeneous ecosystem composed of distinct hardware and software components. The advent of different network techniques for the IoT promises an appropriate Quality of Service (QoS) in different industrial domains, especially when the desired Service Level Agreement (SLA) may be contracted with the network operator, *e.g.*, a Mobile Network Operator (MNO) in the case of cellular technologies. However, the management of industrial systems can be seriously impacted by the integration of IoT. Typically, resource constrained, easily accessible, and heterogeneous IoT devices could be deployed in large numbers over an industrial environment raising questions about performance (*i.e.*, QoS), authenticity, security, privacy, and trust in the data collection process. These problems can emerge at different levels of the Industry 4.0 (I4) integrated system. To this end, recent research has shown interest in integrating Blockchain (BC) to bring trust and transparency in applications composed of untrustworthy actors. The integration of BCs with IoT abbreviated as BIoT may bring authenticity, security, and trust into the I²oT ecosystem.

BCs are public ledgers of the Peer-to-Peer (P2P) architecture that contain data records provided by users or machines,

e.g., Machine-to-Machine (M2M) applications. The content of the ledger is organized as a trusted ordered list of blocks, *i.e.*, *blockchain*, queryable by third parties. The BC emerges through a *mining* process performed among participating peers. Beyond financial markets, BCs are currently deployed in various industrial applications such as transportation, energy, or healthcare [33].

IoT-based systems are often restricted in terms of processing power, storage, networking, and energy, thus, typical resource-hungry BC cannot be directly ported towards IoT devices. Therefore, the management of enterprise I4 infrastructures based on I²oT for the automation of data acquisition needs appropriate methodologies to tackle the integration challenges. There is an increasing attention focused on the BIoT-integrated systems [27] supporting various technologies such as supply chain monitoring [32], [35], smart industries [45], and industrial pollution monitoring [36]. The primary interest of BIoT lies in data authenticity and immutability used in policy based control loops involving measurements, data collection, and control. Moreover, the BIoT system provides data transparency through the introduction of security and consensus mechanisms on top of properly organized IoT perception and network layers [31], which guarantee an appropriate level of data authenticity and integrity, before the data is actually inserted into the BC. The lessons learnt in BIoT can be applicable towards I4 providing a fully operational BC I²oT (BI²oT) combining trustworthiness of the data collection coupled to the specific QoS requirements of a given industry branch.

IoT devices are often of heterogeneous hardware and software architectures having different communicating devices equipped with various communication stacks. Therefore, the *BIoT standardization* is required to properly run necessary BC functions on heterogeneous devices as well as to communicate over different networks. To provide a fully functional and trustworthy BI²oT architecture, there is a need for the strict separation of functions (*e.g.*, originating transactions (TXs), validating TXs) among IoT devices and other computing infrastructures (*e.g.*, edge-nodes, cloud infrastructures). When the function placement is specified, the auxiliary components, *i.e.*, Application Programming Interface (API), have to be standardized to allow for the management and development of a broad range of use-cases based on different underlying BC solutions, *e.g.*, Bitcoin (BTC) [5], Ethereum (ETH) [7]. It is worth noting that there exist more than 2700 BCs on the

market, therefore, a *universal API* able to integrate a large spectrum of solutions is required.

This paper proposes a comprehensive, fully developed, and BC-oriented architecture for BI²oT systems abbreviated as BIIT. This novel design allows for automated heterogeneous data collection performed by the IoT infrastructure. The mechanisms included in the design provide data reliability, overcome limits of the computing overhead, and enhance the energy efficiency of BI²oT integrated networks.

This paper is organized as follows. Sect. II discusses the related work. Sect. III provides a functional architecture design obtained by dividing responsibilities among I²oT systems and other computing infrastructures. The implementation of an BI²oT integrated system with different IoT networking stacks is described in Sect. IV and it is then followed by the experimental validation of the developed solutions in Sect. V. Finally, Sect. VI concludes this paper.

II. RELATED WORK

I4 [47] is a revolution in smart manufacturing, which enables the fully automated production life-cycle management. I4 should employ Internet of Things (IoT), middlewares for data acquisition and actuation, and other components such as Enterprise Resource Planning (ERP) systems introducing secure data operations (*i.e.*, management and processing). In I4, all processes and operations starting with the production of raw materials and followed by logistics, production, inventory, and recycling, should be covered by IoT, in which the complete information about the performance or failures of industrial processes can be collected using sensors and derived upon information processing. Current ERP systems [4] already possess certain capacity of secure data management and processing of often disconnected industrial systems. In the future, the system will benefit from the full connectivity using high capacity data streams coming from IoT devices. Therefore, autonomous industrial management of fully connected systems will emerge using closed control loops provided through policy management. Moreover, the information on industrial processes could be exchanged with external partners allowing for fully self-managing and optimal supply chains. However, to fully accomplish this goal, several problems of IoT data streams such as performance, authenticity, security, privacy, and trust have to be simultaneously tackled. This paper aims to solve these issues through the integration of BIOT systems with I4. In the following subsections, we survey current problems in the IoT, BC, and BIOT domain.

A. Internet of Things

In IoT systems, there are several architectures with varying levels of centralization. In the most popular fully centralized approaches, the application platforms located in the Internet (*e.g.*, in the cloud) retrieve raw data from acquisition networks, process the data, and present the processed information towards third party services [25]. Therefore, a central cloud provider is often used to store significant amounts of data and make it publicly or privately available. Currently, there are

many platforms following this paradigm such as Amazon Web Services (AWS) IoT [1], Azure IoT solution accelerators [3], or The Things Network (TTN) [12].

There are many threats related to IoT architectures, which cause the disruption of the IoT integrated systems and raise questions about the system and data reliability [42]. For instance, (1) Denial of Service (DoS) attacks can exhaust the system by overloading various resources (*e.g.*, computing or physical spectrum). As an example, jamming of the communication channel can disrupt communication in a given area. (2) Destruction of the physical infrastructure is feasible, while the IoT devices are often insecure against physical access. Another threat relates to (3) Eavesdropping, which relates to extracting secret information through the overhearing of communication channels. However, when the infrastructure is compromised through either node capturing (*e.g.*, tempering the node for key extraction) and (4) Node controlling, the attacker could benefit from the information circulating in the entire network (*e.g.*, if a single master key were used to protect the communication).

To allow for information confidentiality, authenticity, and integrity, different mechanisms are specified through cryptographic mechanisms at various levels of the system including IoT perception, network, and middle-ware layers [31] as well as the application platform typically residing in the Internet (*e.g.*, AWS IoT [1], Azure IoT [3], or TTN [12]).

Furthermore, a fully trusted system [42] should be fault tolerant, *i.e.*, composed of robust implementations and usable systems integrating intrusion detection, prevention techniques, and recovery services allowing for high system reliability. Finally, the system should bring open interfaces (*i.e.*, an API to easily integrate with other business applications) offering significant market penetration, long time return on investment for involved companies (*e.g.*, a company lock-in), scalability to grow with the number of connected devices, and interoperability among different heterogeneous devices.

In terms of wireless connectivity, IoT comes with different communication standards. As an example, Bluetooth Low Energy (BLE) [29] or IEEE 802.15.4 [16] can be classified as Wireless Personal Area Networks (WPAN). Low Power Wide Area Networks (LP-WANs) include the Long Range Wide Area Network (LoRaWAN) or cellular standards developed by the 3rd Generation Partnership Project (3GPP) solutions such as Extended Coverage Global System for Mobile Communications Internet of Things (EC-GSM-IoT), Narrow Band-IoT (NB-IoT), and LTE Machine Type Communication (MTC) [48]. It is worth noting that IoT networking uses both Internet Protocol (IP)-full stacks (*e.g.*, Thread LR-WPAN [13] or cellular LP WANs) or IP-free protocol stacks (*e.g.*, ZigBee [26], [28] LR-WPAN or LoRaWAN [15] LP-WAN). Considering LP-WPANs, there was a large standardization effort performed in 1990s and 2000s leading to the advent of the IEEE 802.15.4 standard and the ZigBee specification. The ZigBee protocol stack was meant to provide cross-vendor interoperability through the specification of the management and data planes. ZigBee allows for the autonomous provi-

sioning of multi-hop networking, configurable security, and application management. In particular, the *Security Service Provider* module was developed to standardize security functions required in the ZigBee network, *i.e.*, *Advanced Encryption Standard* (AES), and *ZigBee Device Object* (ZDO) was provided as an auxiliary component for network management. We believe that similar standardization activities have to be performed in BIoT/BI²oT systems to standardize appropriate system components (*e.g.*, API) allowing for the seamless integration of different BC technologies with heterogeneous heavily resource-constrained IoT platforms.

B. Blockchains

Blockchains (BCs) or *Distributed Ledgers* [19] provide new means for trustworthy services instantiated among untrustworthy peers. Potentially, BCs can revolutionize various areas of economy [36], [37], [39] storing data in a secured, temper-proof, and usable distributed system that can be accessed and publicly queried by third parties. Typically, BCs are divided into several classes such as *private* and *public* as well as *permissioned* and *permission-less*. Moreover, BCs maintain decentralized and trusted *consensus mechanisms*. The consensus mechanism is a *principle-set* that provides a global agreement in the system, which does not require a central authority. There are several consensus strategies defined in the literature such as Proof-of-Work (PoW) [34], Proof-of-Stake (PoS) [40], Byzantine Fault Tolerance (BFT) [46] implementing the Practical BFT (PBFT) [21] algorithm, or Proof-of-Authority (PoA) [14], which are adopted by the most popular BC systems. Typically, the mining operation in PoW BC (*e.g.*, BTC, ETH) requires high processing power. Other BC functions such as full nodes acting as wallets/validators require significant amounts of networking to propagate blocks and TXs and high capacity storage to keep the content of the BC. However, the BC length in PoW does not grow too fast due to the relatively low system capacity (*e.g.*, 3.3–7 TXs Per Second (TPS) in BTC [24]).

C. IoT Performing as Blockchain Nodes

IoT systems integrating BCs can provide solutions in the area of open communication offering data integrity, reliability, and authenticity [38]. However, BCs raise questions about performance (*i.e.*, QoS), as popular PoW consensus strategies (*e.g.*, BTC) [34], [49] have known scalability issues in terms of processed TPS. Moreover BC raise interoperability problems among heterogeneous resource constrained IoT devices. Typically, BC participants (*i.e.*, wallets, full nodes, miners) use cryptographic functions applied to TXs and blocks. The performance of a few selected cryptographic functions such as SHA-256 for sealing TXs in a block upon BTC mining or the light-weight and IoT promising Elliptic Curve Digital Signature Algorithm (ECDSA) Ed25519 with SHA-512 [17] for signing TXs is studied among different computing architectures and summarized in Table I.

The performance of cryptographic algorithms are evaluated in the time domain among the TelosB (MSP430) [41] device

with an 8 MHz CPU, ATmega 2560 (Arduino Mega) [8] node with a 16 MHz CPU, Raspberry Pi 3 (RPI) [11] system with 1.2 GHz, and a regular PC with an Intel i7 CPU of 2.4 GHz. The SHA-256 implementation [23] and the TweetNaCl SHA-512 and Ed25519 implementations [18], [30] are used in this benchmark and evaluated in terms of computed Hashes Per Second (HPS) for SHA-256 and SHA-512 or Signatures Per Second for Ed25519. Those implementations are written in pure C and are portable among a broad spectrum of computing devices starting with the MSP430 and finishing with the Intel processor families. Please notice that there are better implementations available. For example, the Arduino Cryptographic Library (ACL) [2] for the AVR CPU family of the ATmega 2560 board was evaluated in our Ed25519 benchmarks 9.23 times faster than TweetNaCl, however, it is at the price of code portability. Moreover, when micro-controllers start natively support cryptographic functions in hardware, *e.g.*, Ed25519, the performance of cryptographic primitives may grow again by several magnitudes providing a couple of signatures per second.

It is, therefore, concluded that the IoT mining of BTC PoW (*e.g.*, SHA-256) on heavily constrained devices will never materialize as for example one Application-Specific Integrated Circuit (ASIC) device of 20 Tera TPS (TTPS) in BTC [10] might be as efficient as billions of TelosB or Arduino resource-constrained devices. Moreover, even signing a TX with Ed25519 seems heavy for constrained devices as well.

TABLE I
PERFORMANCE OF SELECTED CRYPTOGRAPHIC FUNCTIONS ON
DIFFERENT HARDWARE PLATFORMS.

	TelosB [△]	ATmega 2560	RPI*	Intel-i7*
SHA-256 [HPS]	34.22	79.8	31989	182216
SHA-512 [HPS]	4.79 [†]	10.46	12194	97655
Ed25519 [signatures/s]	0.0036 [†]	0.0179	30.1	84.2

[△] the code is run using the Contiki 2.7 OS.

[†] compiled with the unpatched MSP430 GCC 4.6.3 having the compiler code optimization switched off (-O0); for all other studied compiler optimizations techniques (*i.e.*, -Os, -O1...3 compiler options), the TweetNaCl did not return appropriate results (*i.e.*, hashes or signatures),

* a single threaded code executed on one core using the Linux OS.

Table II summarizes different functions in popular existing BCs according to the use of CPU, RAM, storage, and networking. Different functions are classified according to the *low*, *moderate*, *high*, and *impossible* scale assuming an arbitrary BC of high popularity, when a given function runs on the IoT device, *i.e.*, without offloading, *e.g.*, through Mobile Edge Computing (MEC), etc. We assumed IoT constrained peers having CPU, RAM, network, and storage equal to 10 MHz, 10 kB, 100 bps – 100 kbps, and 100 kB – 1 MB respectively, which overlaps with the IoT device of Class 0 [20], represented by the TelosB or AVR device families. We are not going to provide the complete analysis, however, we illustrate our

reasoning using the PoW Mining scenario, in which BTC is used for evaluations. Having an average block-time of 10 mins, block-size of 1–4 MB, the performance of an ASIC miner expressed in tens TTPS, we evaluated that “PoW Mining” is not possible due to CPU (*cf.* Table I) and memory constraints and requires very high network and storage utilization.

TABLE II
REQUIRED USE OF RESOURCES FOR DIFFERENT SOFTWARE FUNCTIONS
IN BCs FROM THE IoT RESOURCE-CONSTRAINED DEVICES’
PERSPECTIVE.

BC Function	CPU	Memory	Storage	Network
BFT Ordering	high	impossible	high	impossible
PoA Ordering	high	impossible	high	impossible
PoW Mining	impossible	impossible	high	high
PoS Mining	high	impossible	impossible	high
Full Node [‡]	impossible	impossible	impossible	impossible
Light Node [◊]	high	high	impossible	high
Submit TX	high	low	low	low
Verify TX	impossible	high	impossible	impossible

‡ A full node providing TX and block verification, which stores the complete chain.

◊ The light node possessing portion of the BC; submits and verifies TXs.

D. Requirements of I4

As discussed, IoT technologies in their current state suffer from deficits in terms of data authenticity. Other problems related to security and privacy are often tackled by IoT networking architectures, *e.g.*, CoAP [44], ZigBee [26], [28], which do not, however, guarantee data manipulations at the later information processing stages. BC offers data immutability, however, trust, transparency, and reliability of the data stored in BIoT systems are only valid if (1) Data Collection, (2) Data Transmission, and (3) Data Storage is appropriately handled to overcome manipulations before the data is transmitted into the BC. Other challenges in industrial use-cases are the radio noise and obstacles in radio propagation. Thus, the signal quality has to be monitored to provide an appropriate level of QoS.

The evaluation of BIoT technologies lead to the immediate question whether the IoT can be appropriately integrated with BC-integrated use cases. Based on the analysis on IoT devices and BCs primitives (*e.g.*, the duration of hashing and signing in TX preparation), we considered that TX submission can be originated directly from IoT devices, while more resource and computing intensive operations have to be delegated to high capacity infrastructures such as clouds.

The following question that arises is whether the BIoT integrated systems adhere to the requirements of I4. In order to facilitate the development of BI²oT systems, the standardization effort is required for the BIoT integration. Specifically, a set of concrete IoT management approaches have to be developed in I4 that will facilitate the development of use-cases using the underlying standard APIs that guarantee an appropriate level of QoS required among different industry branches.

III. BIIT DESIGN AND ARCHITECTURE PRINCIPLES

To address the BI²oT integration challenges in I4, this paper proposes a set of management approaches called BIIT

with the goal of managing the underlying heavily constrained IoT infrastructures in the BIoT-enabled I4 setups. The core objectives considered in BIIT are explained in the following subsection.

A. BIIT Objectives

1) BIIT needs to provide a configurable BIoT transmission scheme. The flexible APIs shall enable an IoT logic compatible with the targeted BC implementation. The BIIT architecture and approaches shall integrate various IoT networking technologies, *e.g.*, Thread, LoRaWAN, and cellular networks. Thus, management components defined in BIIT need to optimize the transmission on-the-fly (*i.e.*, provide appropriate fragmentation and take into account the underlying network characteristics) to provide an optimized solution that guarantees an appropriate level of TX throughput, packet loss, and energy efficiency [43].

2) IoT shall be compliant to the target BC by using similar network primitives (*e.g.*, TX data format). A TX signed by an IoT device assures the authenticity of the data submitted to the BC. However, BIIT shall preserve the BC-specific TX fields providing an appropriate level of BC security (*e.g.*, ETH nonce TX fields protecting against double spending).

3) BIIT needs to provide an appropriate level of security upon data collection as well as configurable APIs to cover authentication, encryption, and data integrity (*e.g.*, Message Digest, ECDSA) upon TX submission. The management level decisions to be taken can be based on parameters such as (a) security level to be maintained, (b) computational complexity, (c) storage requirement, and (d) power efficiency. Thus, BIIT shall enable and consider the security APIs for hashing, signing, and encrypting messages in a similar way as *Security Service Provider* module was integrated in ZigBee to provide the required level of privacy.

B. BIIT Architecture

Adhering to the required dimensions, this paper proposes the architecture of BIIT (*cf.* Fig. 1). BIIT explicitly defines the following components on IoT devices, and uses them to offer the BI²oT integrated management.

IoT Devices: are end devices used for data collection that communicate with other systems components through the IoT network infrastructure.

BC Wallet: is a software component placed on an IoT device that contains device specific credentials such as BC address, private key, destination addresses, balance, TX counter, etc. This information is essential to provide data authenticity, integrity, and trust. Using the Configuration engine, the BC wallet configures the lower layer components, *i.e.*, Network Adaptation Layer (NAL), to produce appropriate TXs optimized for the underlying network technology. Moreover, the BC wallet issues data packets that shall be converted through the NAL into fully fledged TXs on the southbound interfaces. The data packets may contain information on the the industrial process monitored.

Security Functions: analogically to the Security Service

Fig. 1. BIIT: Blockchain-IoT Compliance Architecture for Industry 4.0.

Provider defined in ZigBee [26], [28], a generic BIIoT architecture has to consider a security engine (*i.e.*, API) producing a large number of hash functions (*e.g.*, SHA-256) and ECDSA signature types (*e.g.*, Ed25519), which guarantee the compatibility with a large spectrum of BC protocols.

Configuration: this engine manages the configuration of the wallet and NAL. The Wallet is configured by the Enterprise End User (EEU) using the management plane. In turn, the configuration of the NAL is requested by the wallet, which demands the computation of appropriate hash functions (*e.g.*, SHA-256) or ECDSA signatures (*e.g.*, Ed25519). Furthermore, the Configuration engine provides the necessary keys (*e.g.*, Ed25519 private key) to the NAL and negotiates the data-plane message format used by the NAL on the south-bound network interfaces.

Software Based Network Adaptation Layer: The configuration of NAL is initiated by the Configuration engine. NAL is a supporting component of the wallet; it receives data packets from the wallet on the data-plane, computes the required hash functions (*e.g.*, SHA-256) or ECDSA signatures (*e.g.*, Ed25519), assembles TXs, and sends them to the underlying network protocol stack. NAL optimizes the packet transmission towards the underlying network layer (*i.e.*, message fragmentation, Automatic Repeat reQuest (ARQ), TX aggregation [43]) and provides TX compatibility with the targeted BC (*e.g.*, BTC, ETH). The adaptation is required as in many situations, the TXs cannot be sent directly by the network layer, *e.g.*, the regular TX size may already exceed the Maximum Transfer Unit (MTU) of the IoT network.

IoT Network Protocol Stack: the physical, data link, and networking layers enabling the communication between IoT nodes and IoT Gateways (GWs), *e.g.*, LoRa GW, and LTE MTC evolved Node B (eNB)).

IoT Network Gateways: this component relies data packets

between the IoT network and other computing infrastructures, *e.g.*, Edge-Nodes and cloud infrastructures.

Network Core: provides the data collection point, such as the TTN servers [12], in which the IoT data is stored temporarily and can be accessed by other system components (*e.g.*, BC Client)

IoT Edge Node: Since BIIT tackles heavily constrained devices, IoT nodes are not envisioned to play the role of full BC clients. The edge node functionality can be provided on less resource constrained devices such as a RPI helping the BC Wallets on IoT devices. In some cases, *e.g.*, in LoRa, the Edge Nodes derive the TXs from TX chunks sent by the IoT device (*cf.* Sect. IV) as for example in LoRa, the regular TX size may exceed the LoRa MTU. It is worth noting that some Edge Nodes can also act as a gateway relying packets from the IoT network towards other computing infrastructures.

BC Client: Typical full BC clients require significant amounts of resources (*e.g.*, storage at the order of GB in the case of BTC or ETH) and are not implementable on IoT devices. The BC Client residing on the Edge Node is an auxiliary element helping out IoT wallets to submit/verify TXs using the Edge Node resources.

IoT Network Plugin: receives the IoT data stored in the network core, *e.g.*, uses Message Queuing Telemetry Transport (MQTT) to access data, and provides the collected packets towards the BC Client using an API.

TCP/IP: The BC client communicates using the TCP/IP stack with other computing infrastructures (*e.g.*, BC miners).

Blockchain: any type of BC (*e.g.*, BTC, ETH) can be attached to the BIIT architecture working on physical or cloud infrastructures.

IV. BIIT IMPLEMENTATION

BIIT is a BI²oT architecture aligned with the 3 major objectives introduced in the Sect. III. In order to have a clear vision on the BIIT management platform, the implementation details for different IoT networks are provided in this section while the implementations is available at [22].

Fig. 2. BIIT Implementation with LoRa – Components' Engagement View.

A. BIIT: Management of the TTN Networks

LoRa networks endure air time and packet size limitations [15]. Thus, aligned with these specific characteristics, the LoRa-to-BC integration management based on BIIT is implemented as follows. Fig. 2 represents the module implementation of the LoRa network including IoT devices and edge nodes.

IoT device: IoT sensors collecting data from industrial installations are connected directly to Arduino Mega (AT2560) boards [8]. The connection is wired-based and enabled through the General Purpose Input/Output (GPIO) pins. Each Arduino device is equipped with a LoRa shield [9] and the whole setup is referred to as the LoRa node. Each LoRa node is able to use the LoRaWAN protocol. The TXs originated by the Arduino wallet have to be signed with the Ed25519 ECDSA using a private key and other parameters stored on the IoT device (e.g., ETH TX nonce). Moreover, in the TTN protocol, the data exchanged over the network is encrypted, therefore the solution meets the privacy requirements of the private BC as well.

IoT Network Gateways: The nearest LoRa GW receives the data and forwards it to the TTN *Network Collector*, which is responsible for decrypting the data and presenting it to third applications registered with the TTN back-end (e.g., through the MQTT protocol).

IoT Edge node: As Edge node, an RPI device is deployed to constantly listen to the TTN sever and to receive messages originated by the LoRa nodes. The RPI is responsible for collecting the relying the TX between the TTN network and the BC miners.

BC TTN Plugin: The role of the BC TTN Plugin is to connect to the back-end of the TTN network through the MQTT protocol, retrieve and acknowledge data chunks originated by the LoRa nodes, and present the reconstructed TX originated by the LoRa nods at the BC Client (cf. Fig. 3).

BC Client: RPI also runs a BC client. Its purpose is the verification of TXs received from the *BC TTN Plugin* and forwarding them to the BC, i.e., Bazo [40]. In our small

Fig. 3. Data Flow and Instantiating of a Blockchain Wallet on IoT Devices.

demonstration setup, the BC miner also resides on the RPI device, therefore, the BC Client can directly submit a TX to the BC. Moreover, both the BC TTN Plugin, and BC Client are involved in the LoRa node recognition procedure (cf. Fig. 3): when a new LoRa node appears in the network, it presents the public key, which is in turn registered with the Bazo permissioned BC. The initial funds are then stored on the LoRa node account to allow for the submission of upcoming TXs (i.e., the LoRa node needs to directly pay the TX fees). This greatly improves the management of new IoT devices joining the network. In the future, we also plan to specify the TX verification procedure. As indicated in Table II, the TX verification is a costly operation for a restricted IoT device. The verification procedure will be, therefore, offloaded from the IoT wallet to the more powerful Edge Node. The wallet will trust the Edge Node BC Client and use it to confirm whether the previously issued TXs were successfully included in the BC. Such a mechanism could drastically increase the fault tolerance in the BI²oT system.

The main issue of BI²oT systems using LoRa is the transmission performance, i.e., maximizing TX throughput, while simultaneously providing the data integrity using cryptographic signatures. In [43], we discovered that the combination of Listen Before Talk (LBT) Medium Access Control (MAC), the ARQ mechanism to handle retransmissions, TX aggregation (i.e., aggregating several data packets into a single BC TX) as well as TX fragmentation into small data chunks (when the TX payload is larger than the interface MTU) can provide a good trade-off between the TX network capacity, packet loss, and energy efficiency. Moreover, the Ed25519 ECDSA is very costly on the IoT device and should be rarely used to limit both the payload (i.e., sending 64 B signature)

Fig. 4. LoRa to Bazo Blockchain Transmission Protocol.

and computational overheads (*cf.* Table I). BIIT manages this condition through a LoRa specific transmission scheme between the LoRa node and the TTN-plugin (*cf.* Fig. 4).

The data collected by IoT devices has to be divided into smaller chunks (*i.e.*, 51 B) to fit the LoRa MTU with Spreading Factor 12. The employed TXs shall not create high overhead, therefore signing is employed only upon N th data packet, which reduces the computational overhead caused by the Ed25519 ECDSA algorithm (*i.e.*, residing in the Security engine) and payload overhead caused by the Ed25519 ECDSA signature. By signing the data at the IoT device, authenticity and integrity of the data can be verified. IoT edge nodes collect the TX payloads from several chunks of data exchanged over the TTN, strip the received messages from the LoRa transmission headers, attach required BC headers, and send the complete TX to the BC Client. The protocol uses Segment Numbers (Seg. Nb.), which are used for tracking and re-sending lost chunks of data. To inform the IoT device about the data reception at the GW side, ACKs are employed. Finally, our mechanism assures that BC TXs derived at the BC Client comply to standard of the targetted BC protocol (*e.g.*, ETH, ETC, Bazo) and may be successfully submitted for mining.

B. BIIT: Management of the Cellular Networks

The management of BIoT cellular networks is much simpler in comparison to LoRa. First of all, the cellular stack is IP-based. Second, IoT technologies provide large MTUs of 1600 B [43], [48], therefore, they can send several data chunks in one shot. The Configuration and Security engine play exactly the same role as in LoRa, however, the NAL is configured differently. NAL accumulates several chunks of data (*e.g.*, measurements) in cache, and then sends one large TX containing multiple data chunks signed with a single Ed25519 signature. The resulting TX is enclosed in a User Datagram Protocol (UDP) data packet, which is confirmed with a single ACK message from the IoT Edge Node. Finally, BIIT uses the UDP Packet Listener on the IoT Edge Node side (*cf.* Fig. 5) as the IoT Network Plugin.

Fig. 5. Cellular TX Design.

C. Configuration and Security Engines

The *Configuration* engine (*cf.* Fig. 1) is responsible for the configuration of NAL, which requires setting up an appropriate TX scheme (*cf.* Subsects. IV-A, IV-B) and providing the secret keys so that the NAL can compute an appropriate ECDSA. NAL uses the API provided by the Security engine, which implements the required cryptographic functions (*e.g.*, Ed25519).

V. EXPERIMENTAL PERFORMANCE EVALUATION

BIIT needs to justify its viability by enhancing the system automation and performance with respect to the considered criteria (*cf.* Sect. III).

A. BIIT Performance in LoRa

In this work, BIIT is evaluated using Arduino Mega Devices (AT Mega 2560) [8] equipped with the Dragino LoRa communication shield [9]. The Edge Node is provided on an RPI 3 hosting the BC TTN Plugin, Bazo BC Client, and Bazo BC miner [22]. RPI also provides a local TTN GW. The Arduino device and TTN-plugin receive End-to-End (E2E) connectivity through the TTN network. The Arduino device sends traffic using LoRa with Bandwidth (BW) of 125 kHz, SF 12, and transmission power equal to 27 dBm. Every segment is retransmitted up to a configurable retry number, when the corresponding ACK does not arrive.

The evaluation of the employed TX scheme for one device is shown in Fig. 6 and Fig. 7. In Fig. 6 (left y axis) presents the amount of total traffic volume exchanged in the network

for a TX consisting of several packets signed (x axis) of size equal to 40 B. The traffic types can be divided into three groups (*cf.* Fig. 4) for remote and local TTN GWs: (a) Data (*i.e.*, payload), which increases upon sending several data chunks of 40 B per TX, (b) Signatures fragments (32 B each), and (c) ACKs (2 B). The Packet and Traffic overhead, which is measured in relative values displayed on the right y axis, consists of signatures and ACKs. Please notice that there is a higher number of ACKs than Data packets, while the Signatures are also acknowledged. The relative overhead decreases in aggregated TXs due to an increased number of data chunks (40 B) signed with one heavy Signature packet (64 B). Fig. 6 and Fig. 7 also compare the transmission performance between a local indoor GW vs. a remote GW, *i.e.*, less retransmissions to a local gateway observed due to better radio conditions. Thus, it is highly recommended to install a local GW especially in suburban environments having weak TTN coverage.

Fig. 6. The Packet Volume Exchanged in The Network Upon One TX Consisting of Several Data Packets.

Fig. 7. The Total Traffic Volume Exchanged in the Network Upon One TX Consisting of Several Data Packets.

B. BIIT Performance in Cellular Networks

In this experiment, a similar setup as in Subsect. V-A is used, however, Arduino Mega Devices (AT Mega 2560) [8] are equipped with the Sixfab Cellular IoT Application Shield [6]. The access to LTE Cat. M networks is provided by Swisscom AG in the Zurich, Switzerland area. TXs are sent in UDP messages. The RPI again plays the role of the IoT Edge Node, however, it receives data from the Swisscom network through

the Internet using the UDP Packet Listener (*cf.* Fig. 5). In our measurements, the node wakes-up, re-attaches to the cellular network, and sends a data packet towards the IoT Edge Node, if there are enough collected data chunks in cache to send. The sending procedure requires the computing of BC headers (also including signatures). The measurements collected on a single devices are presented in Fig. 8. Sending an aggregated TXs (composed on multiple 40 B data chunks, *e.g.*, measurements) has a positive impact on both, the relative BC data overhead (*i.e.*, data transmitted vs. actual data size) and energy efficiency, because the resulting Bazo header is heavy (141 B) and the Ed25519 ECDSA is computationally expensive. Furthermore, the most dominant energy consumption is the result of the cellular connection re-initialization after a deep sleep and the computation of an Ed25519 signature, which can take up to several seconds. We were unable to perform NB-IoT or EC-GSM-IoT experiments; Swisscom does not provide EC-GSM-IoT in the area, and we were unable to connect to their NB-IoT infrastructure. Therefore, only LTE Cat. M is considered.

Fig. 8. BIIT Performance Using The Cellular Connectivity.

VI. SUMMARY AND OUTLOOK

This work provides the specification of a management architecture, which integrates BioT systems with the landscape of I4. The TXs originate directly at the IoT submitting devices and are included in the BC operated on higher performance computing infrastructures. The developed architecture allows for easy management of IoT devices and data streams in the industrial setup, and provides high performance protocols for different IoT network access technologies. The future perspective of this work includes the large scale experiments with BIIT to confirm the system scalability in LoRa and cellular networks.

VII. ACKNOWLEDGEMENTS

This paper was partially supported by (a) the University of Zürich (UZH), Switzerland and (b) the European Union Horizon 2020 Research and Innovation Program under grant agreement No. 830927, namely the Concordia project.

REFERENCES

- [1] "Amazon Web Services - Internet of Things," <https://aws.amazon.com/iot>.
- [2] "Arduino Cryptography Library," <https://github.com/rweather/arduinoilibs>.
- [3] "Azure IoT Solution Accelerators," <https://azure.microsoft.com/en-us/features/iot-accelerators/>, last visit: May 17, 2019.
- [4] "BBV Software Services," <http://www.bbv.ch/images/bbv/pdf/Publikationen/BBV-Poster-Uebersicht-Industrie-4.0.pdf>.
- [5] "Bitcoin is An Innovative Payment Network and a New Kind of Money," <http://bitcoin.org>, last visit: October 1, 2019.
- [6] "Cellular IoT Application Shield for Arduino," <https://sixfab.com/category/arduino-cellular-iot-application-shield>.
- [7] "Ethereum is a Global, Open-Source Platform for Decentralized Applications," <https://www.ethereum.org>, last visit: October 1, 2019.
- [8] "Getting Started with Arduino and Genuino MEGA2560," <https://www.arduino.cc/en/Guide/ArduinoMega2560>.
- [9] "LoRa Shield for Arduino," <http://www.dragino.com/products/loraimem/102-lora-shield.html>, last visit: August 30, 2019.
- [10] "Mining Hardware Comparison," https://en.bitcoin.it/wiki/Mining_hardware_comparison, last visit: October 1, 2019.
- [11] "Raspberry Pi 3 Model B," <https://www.raspberrypi.org/products/raspberry-pi-3-model-b>, last visit: October 2, 2019.
- [12] "The Things Network (TTN)," <https://www.thethingsnetwork.org/>.
- [13] "Thread Group," <https://www.threadgroup.org/>, last visit: May 17, 2019.
- [14] S. D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain," in *Italian Conference on Cyber Security (ITA-SEC 2018)*, January 2018.
- [15] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things," *Sensors*, Vol. 16, No. 9, 2016.
- [16] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless Sensor Networks: A Survey on the State of The Art and the 802.15.4 and ZigBee Standards," *Computer Communications*, Vol. 30, No. 7, pp. 1655 – 1695, 2007, wired/Wireless Internet Communications.
- [17] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, "High-speed High-security Signatures," in *Proceedings of the 13th International Conference on Cryptographic Hardware and Embedded Systems*, ser. CHES'11. Berlin, Heidelberg: Springer-Verlag, 2011.
- [18] D. J. Bernstein, B. van Gastel, W. Janssen, T. Lange, P. Schwabe, and S. Smetsers, "TweetNaCl: A Crypto Library in 100 Tweets," in *Progress in Cryptology - LATINCRYPT 2014*, D. F. Aranha and A. Menezes, Eds. Cham: Springer International Publishing, 2015, pp. 64–83.
- [19] T. Bocek and B. Stiller, *Smart Contracts - Blockchains in the Wings*. Tiergartenstr. 17, 69121 Heidelberg, Germany: Springer, jan 2017, pp. 169–184.
- [20] C. Bormann, M. Ersue, and A. Keranen, "Terminology for Constrained-Node Networks," in *Internet Research Task Force, RFC*, Vol. 7228, May 2014.
- [21] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," *ACM Trans. Comput. Syst.*, Vol. 20, No. 4, pp. 398–461, Nov. 2002.
- [22] I. Cepilov, E. Schiller, and S. Rafati Niya, "BIIT TTN Plugin, Bazoclient and miner Implementations," <https://gitlab.ifi.uzh.ch/csg/biot/>.
- [23] B. Conte, <https://github.com/B-Con/cryptographicalgorithms/blob/master/sha256.c>, last visit: August 29, 2019.
- [24] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer, "On Scaling Decentralized Blockchains," in *Financial Cryptography and Data Security*, J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 106–125.
- [25] A. Dohr, R. Modre-Oprian, M. Drobics, D. Hayn, and G. Schreier, "The Internet of Things for Ambient Assisted Living," in *2010 Seventh International Conference on Information Technology: New Generations*, April 2010, pp. 804–809.
- [26] S. Farahani, *ZigBee Wireless Networks and Transceivers*. Newnes, 2011.
- [27] T. M. Fernández-Caramés and P. Fraga-Lamas, "A Review on the Use of Blockchain for the Internet of Things," *IEEE Access*, Vol. 6, pp. 32 979–33 001, 2018.
- [28] C. Gomez, J. Oller, and J. Paradells, "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology," *Sensors*, Vol. 12, No. 9, pp. 11 734–11 753, 2012.
- [29] D. Gislason, *Zigbee Wireless Networking*. Newnes, 2008.
- [30] S. Josefsson and I. Liusvaara, "Edwards-curve digital signature algorithm (eddsa)," in *Internet Research Task Force, Crypto Forum Research Group, RFC*, Vol. 8032, 2017.
- [31] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges," in *2012 10th International Conference on Frontiers of Information Technology*, December 2012, pp. 257–260.
- [32] N. Kshetri, "Can Blockchain Strengthen the Internet of Things?" *IT Professional*, Vol. 19, No. 4, pp. 68–72, 2017.
- [33] M. Mettler, "Blockchain Technology in Healthcare: The revolution Starts Here," in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, September 2016.
- [34] S. Nakamoto, "Bitcoin: A Peer-to-peer Electronic Cash System," <http://bitcoin.org/bitcoin.pdf>, 2008, last visit: May 17, 2019.
- [35] S. R. Niya, D. Dordevic, A. G. Nabi, T. Mann, and B. Stiller, "A platform-independent, generic-purpose, and blockchain-based supply chain tracking," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, May 2019, pp. 11–12.
- [36] S. R. Niya, S. S. Jha, T. Bocek, and B. Stiller, "Design and Implementation of an Automated and Decentralized Pollution Monitoring System with Blockchains, Smart Contracts, and LoRaWAN," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, April 2018, pp. 1–4.
- [37] S. R. Niya, L. Pelloni, S. Wullschlegler, A. Schaufelbühl, T. Bocek, L. Rajendran, and B. Stiller, "A Blockchain-based Scientific Publishing Platform," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, May 2019, pp. 329–336.
- [38] S. R. Niya, E. Schiller, I. Cepilov, F. Maddaloni, K. Aydinli, T. Surbeck, T. Bocek, and B. Stiller, "Adaptation of Proof-of-Stake-based Blockchains for IoT Data Streams," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, May 2019.
- [39] S. R. Niya, F. Schüpfer, T. Bocek, and B. Stiller, "A Peer-to-peer Purchase and Rental Smart Contract-based Application (PuRSCA)," *Information Technology*, Vol. 60, No. 5, pp. 307–320, oct 2018. [Online]. Available: <https://www.degruyter.com/view/j/itit. ahead-of-print/itit-2017-0036/itit-2017-0036.xml>
- [40] S. R. Niya and B. Stiller, "BAZO: A Proof-of-Stake (PoS) based Blockchain," IFI-TecReport No. 2019.03, Zürich, Switzerland, Tech. Rep., May 2019.
- [41] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-low Power Wireless Research," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, ser. IPSN'05. Piscataway, NJ, USA: IEEE Press, 2005.
- [42] R. Roman, J. Zhou, and J. Lopez, "On the Features and Challenges of Security and Privacy in Distributed Internet of Things," *Computer Networks*, Vol. 57, No. 10, pp. 2266–2279, 2013.
- [43] E. Schiller, S. Rafati Niya, T. Surbeck, and B. Stiller, "Scalable Transport Mechanisms for Blockchain IoT Applications," in *2019 IEEE 44th Conference on Local Computer Networks (LCN) (LCN 2019)*, Osnabrück, Germany, October 2019.
- [44] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," in *Internet Research Task Force, RFC*, Vol. 7252, 2014.
- [45] J. J. Sikorski, J. Haughton, and M. Kraft, "Blockchain Technology in the Chemical Industry: Machine-to-Machine Electricity Market," *Applied Energy*, Vol. 195, pp. 234 – 246, 2017.
- [46] J. Sousa, A. Bessani, and M. Vukolic, "A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2018, pp. 51–58.
- [47] B. Tjahjono, C. Esplugues, E. Ares, and G. Pelaez, "What Does Industry 4.0 Mean to Supply Chain?" *Procedia Manufacturing*, Vol. 13, pp. 1175–1182, 2017.
- [48] A. D. Zayas and P. Merino, "The 3GPP NB-IoT system Architecture for The Internet of Things," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2017, pp. 277–282.
- [49] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, June 2017, pp. 557–564.