Industry 4.0: Smart Contract-based Industrial Internet of Things Process Management

Charles Tim Batista Garrocho[*†], Célio Márcio Soares Ferreira[†], Ailton Sávio Sacramento Júnior[†],
Carlos Frederico Marcelo da Cunha Cavalcanti[†], and Ricardo Augusto Rabelo Oliveira[†]
[*]*Minas Gerais Federal Institute of Education, Science and Technology*
Ouro Branco, Minas Gerais, Brazil
charles.garrocho@ifmg.edu.br
[†]*Computer Science Department, Federal University of Ouro Preto*
Ouro Preto, Minas Gerais, Brazil
celio@linuxplace.com.br, ailton.junior@aluno.ufop.edu.br, cfmcc@iceb.ufop.br, rrabelo@gmail.com

***Abstract*—Through the Internet of Industrial Things, significant investments in the industry are expected. In this new environment, machine-to-machine communication showed immediate potential. However, most communication implementations require a trusted intermediary. The introduction of smart contracts can enable communication without the need for a trusted intermediary. To provide security and decentralization in industrial communication processes, smart contract-based middleware is proposed. This proposal is evaluated for impacts against the stringent communication requirements required by industrial applications. Experimental results show that while this approach offers greater security and decentralization than traditional proposals, blockchain-based smart contracts cannot yet be applied to industrial systems due to blocking time.**

***Index Terms*—IIoT, M2M, Blockchain, Smart Contracts**

I. Introduction

Industry 4.0 refers to the fourth industrial revolution that transforms industrial manufacturing systems into cyber production systems, introducing emerging paradigms of information and communication, such as the Internet of Things (IoT) [1]. An Accenture [2] study points out that by 2020, IoT investments in the world should reach US$ 500 billion. With the introduction of Industrial IoT (IIoT) in the factory, a 30% increase in productivity is expected, generating a very optimistic investment forecast of US$ 13 trillion by 2030.

IIoT devices have low processing and storage, low bandwidth for data transmission and collection, and limited autonomy [3]. With the popularization of these devices and in the face of such constraints, it was necessary to develop new Machine-to-Machine (M2M) communication protocols to address these limitations [4]. Currently, IIoT devices can communicate directly or through protocols that work with the Publish-Subscribe paradigm, which allows data to be available to multiple consumers.

Today, with M2M communication applied in industry 4.0, there is a quest for complete process automation as well as the removal of repetitive, often dangerous and business-critical tasks. Industry 4.0 encompasses disruptive technologies and standards that lead to complete decentralization of the supply chain, logistics, and process control [5]. The basis of this industry is to connect machines, systems, and assets to create intelligent networks that assist in productive control.

M2M communication has immediate potential in industrial applications. However, most M2M communication implementations use a communication model that requires a trusted central node [6]. Using blockchain-based smart contracts in M2M communication would allow the use of a decentralized peer-to-peer (P2P) network without the need for an intermediary. M2M applications that act independently or without user interference need to be transparent, secure, and traceable.

Several works related to M2M communication security are introducing concepts of this area into practice in IIoT applications [7]. However, classic countermeasures against threats adopted by general-purpose networks (e.g., firewalls and intrusion detection systems) are based on centralized infrastructure that requires significant investments and makes the network fail-safe because network nodes need a central node to exchange messages with each other.

To make communication secure and decentralized, this article introduces smart contract-based middleware for M2M communications at IIoT. This proposal allows a decentralized P2P network to be used for M2M communication between IoT devices without the need for a trusted intermediary. Also, this proposal allows IIoT applications to operate transparently (without user interference), safe and traceable, ensuring that operators can verify the actions of IIoT applications.

Despite the many benefits that smart contracting can offer, this proposal presents time challenges that influence the real-time communication of IIoT devices. Industrial applications impose stringent latency requirements on communication between network nodes to maintain stability and control performance [8]. Thus, the impact of the application of this middleware on the industrial environment is evaluated and the challenges and problems are presented and discussed.

The remainder of this paper is organized as follows: Section II presents the background. Section III presents the architecture and operation of the middleware. Section IV presents the proof of concept, the scenario, and the evaluation metrics. Section V analyzes and discusses the results, and presents challenges and opportunities. Finally, Section VI presents the conclusions.

II. Background

The role of industrial networks is becoming increasingly crucial as they are expected to meet the demanding new industry 4.0 requirements [9]. Industrial networks are used, among others, to monitor conditions, manufacturing processes, and predictive maintenance. These networks have typical configurations, traffic, and performance requirements that make them different from traditional communication systems. Thus, industrial networks are designed to meet the requirements derived from their various fields of application. The most critical requirements are time, reliability and flexibility [10].

A. Industrial Process Automation Systems (IPAS)

IPAS are based on a five-level hierarchy [11]. The sensor and actuator level consist of field devices that communicate over a wireless network with a Gateway that bridge to the control level. The control level consists of devices such as Programmable Logic Controller (PLC) and Distributed Control System (DCS) that control devices in the field, and an interface for Internet Protocol (IP) based network at supervisory level. At the supervisory level, processes are monitored and executed by workers. Finally, in the last tier, there is enterprise and factory management and process-related data sent to the cloud.

IPAS comprise many nodes, logically positioned at various hierarchical levels and distributed over large geographical areas [12]. Many servers and Human-Machine Interface (HMI) computers are used for interaction to the control level. In this context, blockchain can decentralize or support decision making on both internal processes in a factory and external processes in a supply chain. Such an approach can make industrial automation systems fully decentralized and automated.

B. Blockchain-based Smart Contracts

As a decentralized P2P network, blockchain has no single point of failure, presents excellent fault tolerance and implements an unchanging ledger in which each transaction, after completed and included in the blockchain, cannot be erased ou changed [13]. As a P2P network, blockchain has an intrinsic characteristic of being highly scalable. Because all transactions are encrypted, blockchain ensuring security in all transactions, and as a public ledger, auditable and transparency.

Bitcoin is the most successful digital currency in the world. He uses a distributed public book that was the genesis of the term blockchain [14]. Already Ethereum has become a popular platform for blockchain applications, providing new features like smart contracts that significantly contribute to the generation of new application possibilities [15]. Ethereum is a global, open-source platform for decentralized applications.

On Ethereum, can write code that controls digital value, runs exactly as programmed, and is accessible anywhere in the world. Ethereum's smart contracts are computationally "turing-complete" programs, can be written in languages like Solidity, and when compiled, generate bytecodes that run on a machine. Ethereum Virtual Machine (EVM), allowing to create an arbitrary rule machine with transaction formats and state transition functions.

C. Related Work

In this section, we present work related to blockchain-based smart contracts applied in IIoT communication and industrial processes. Following the Kitchenham protocol [16], we performed searches between May and July 2019 on the computer databases: ACM Digital Library, Google Scholar, IEEE Xplore Digital Library, and ScienceDirect (Elsevier). We use the "AND" operator to cross over the following keywords: IIoT, blockchain, smart contracts, and communication. Finally, we selected only articles that provided a full text and published in less than three years were selected.

Blockchain has been applied in some areas of application to solve specific problems. For example, the work [17] uses 5G (*Fifth Generation Cellular Network Technology*) network slice broker on a blockchain to reduce service creation time and enable manufacturing equipment to autonomously and dynamically acquire the share needed for more efficient operations. The work [18] features a blockchain-compatible data sharing and collection scheme and deep reinforcement learning to create a reliable and secure communication environment. However, this paper does not consider the stringent requirements such as latency and reliability that M2M communication requires.

The work [19] presents a blockchain for power trading automation through decentralized M2M communication, getting rid of a trusted intermediary. A credit-based payment blockchain scheme is introduced to support fast and frequent energy trading, and reduce delays in transaction confirmations. Already the work [20] presents a consensus mechanism that reduces the computational cost and accelerates block generation in blockchain; however, the proposed mechanism reduces the level of security in the blockchain network.

Finally, the work [21] presents a blockchain-based industrial network architecture for some industrial case studies. This paper presents the objective of generalizing blockchain application in M2M communication by proposing blockchain as an intrinsic component of M2M communication. This proposal is aimed at the core of the network and aims to make the communication between devices and processes that make up industrial automation systems safer. However, such a proposal is not appropriate as IIoT field devices are not designed to perform operations different from what they were designed to do. Also, the field device network is not IP based, which does not allow communication with the blockchain.

After this research, it is clear that few works of blockchain-based smart contracts were developed for the industrial field. The presented and analyzed works have the objective of automating specific processes horizontally in the communication between factories. However, the analysis and implementation of smart contracts in IPAS vertical communication were not performed by the studies found. Thus, in this paper, we propose and evaluate middleware that inserts smart contracts into the factory to automate IPAS vertical communication. Such an approach integrates with field devices as an intrinsic component in the IPAS hierarchy.

III. Smart Contract Middleware Design for Industrial Process Automation Systems

To intermediate the process control level with the other upper layers of the IPAS hierarchy, the middleware *Industrial Smart Contract Monitor (ISCOM)* is presented. In a large industrial plant, there may be several sectors. In each sector, there is an ISCOM that controls and executes contracts by ordering tasks on field devices. Thus, as illustrated in Figure 1, ISCOM architecture is divided into four modules:

- *Cliente-Contract Interface*: this module provides interaction with supervisory-level HMI device systems as well as enterprise and cloud device systems;
- *Monitoring Manager*: this module performs contract monitoring on internal and external blockchain networks. Interacts with other modules according to state changes in monitored contracts;
- *State Change Manager*: this module monitors field device states and makes state changes to contracts related to these devices in the blockchain network;
- *Execution Manager*: this module receives information from the Contract Monitoring Manager module and requests actions to be performed on actuator devices by communicating with the process control level;

During the operation of ISCOM modules, all actions and processes are recorded in the blockchain through smart contracts. However, many smart contracts and field devices are controlled by ISCOM, making an internal database necessary for the secure and organized storage of this information. Thus, in addition to the modules presented, ISCOM architecture has two databases:

- *Action Data*: this database stores information defined by the supervisory or corporate level for ISCOM decision making in industrial processes;
- *Contracts*: this database stores the identifications of factory contracts as well as external contracts authenticated by the decentralized oracle network (DON).

ISCOM is strategically positioned at the supervision level as it is the last level with IP-based communication where it is required for interaction with the blockchain network. Also, the supervision level enables direct communication with devices in the field through devices such as PLC and DCS. In this way, ISCOM may request a field device to perform service under a smart contract. Besides, ISCOM integrates with HMI, enabling monitoring of smart contract process actions with shop floor workers.

A. Operation

Interaction between ISCOM middleware can be performed by both shop floor workers through HMI and at the corporate level through workstations. The *Client-Contract Interface* module enables this interaction through a permission-level user interface. The corporate level has full permissions, which allows you to deploy, execute, write, and read smart contracts on the blockchain network. The supervision level has partial permissions, which allows writing and reading in smart contracts on the blockchain network.

As illustrated in Figure 2, ISCOM middleware initializes its services through the *Monitoring Manager* module that mediates communication to the blockchain network. Smart contracts are deployed blockchain network. Execution of contracts on the blockchain network depends on actions requested by ISCOM through changes in sensor states that are monitored by the *State Change Manager* module. Any change to the smart contract is recorded in the blockchain and the *Monitoring Manager* module identifies and prompts the *Execution Manager* module to perform an action on an actuator.

Besides, each factory has its private blockchain network where field devices enforce its contracts at a large industrial plant. In a supply chain, one factory can influence the processes of another factory. Thus, smart contracts can define process actions according to external demands. To connect and intermediate smart contracts between private blockchain networks, a DON network is used.

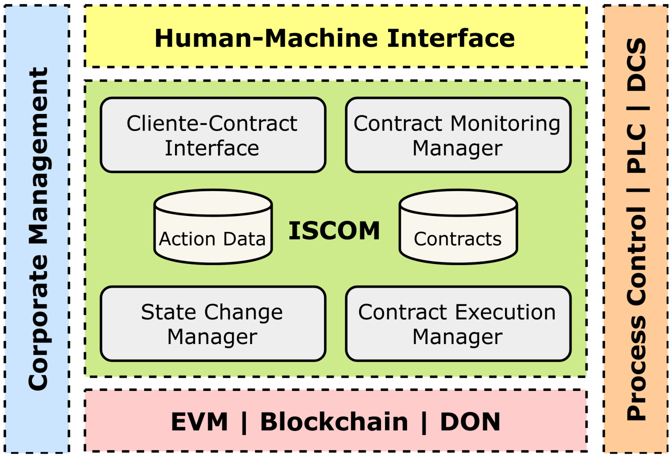

Fig. 1. ISCOM architecture modules.

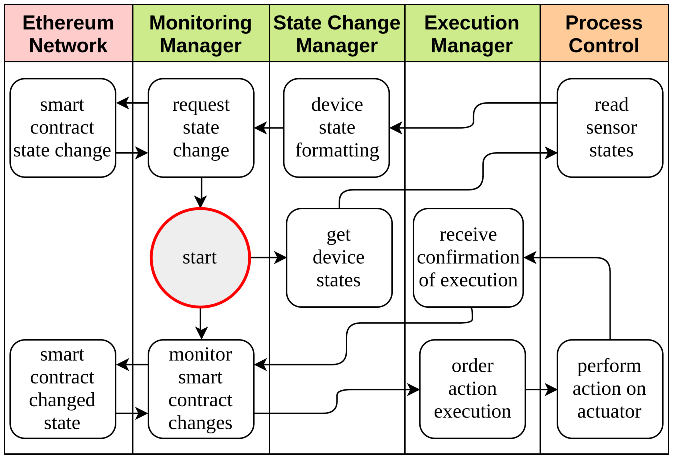

Fig. 2. Sequence diagram of ISCOM operation and states.

IV. Proof of Concept and Evaluation

To evaluate ISCOM, a system was developed to simulate smart contracts in industry 4.0, as illustrated in Figure 3. For this, we use two Raspberry Pi 3 boards representing field devices. Representing the process control layer, we use a Notebook with Mosquitto Broker for the Message Queuing Telemetry Transport (MQTT) server, where sensors publish state changes and actuators subscribe to services published by ISCOM. For ISCOM, Go Ethereum (Geth) protocol was used with a Flask server, and all its management modules were implemented. For the blockchain network and smart contract implementation, the Ropsten Ethereum test network was used.

The proof of concept works in such a way that after the sensing device changes state, it publishes this change in MQTT. ISCOM subscribes to sensor publications and reviews related to smart contracts. ISCOM also carries out execution orders for operations by publishing to MQTT. Actuator devices subscribe to services published by ISCOM and perform operations. A state contract has been developed, which is presented in Listing 1 in the Solidity language. The contract consists of: a variable STATE referring to the sensor state; a SET_STATE function to change the state value; and a GET_STATE function to get the state value.

Although the smart contracts proposed it is a promising way of M2M communication in industry 4.0 applications, some blockchain gaps need to be addressed to adopt the solution massively. The blockchain has a lock time, which takes a node to confirm the valid block as inserted into the list, ensuring consistency of the entire list so that all nodes have the same valid list. Therefore, it is necessary to evaluate this blocking time in the industrial environment.

Most industrial systems perform M2M communication in which applies real-time requirements. For example, latency requirements may range from 10–100ms. Given this, it is necessary to evaluate the impact of latency and blocking time on real-time communication of industrial processes. Therefore, in this evaluation, two experiments were performed in which they measured: *Blocking Time*: response time of sensor device requests to change state in the smart contract; *Latency Time*: delay time leading to communication between field devices and ISCOM server. The two experiments were repeated 100 times. The results data were obtained through calculations in the system itself, with a confidence interval of 95%.

```
pragma solidity ^0.5.1;

contract ConceptProof {
    uint8 private state;

    constructor() public {
        state = 0;
    }

    function set_state(uint8 _state) public {
        require(_state >= 0 && _state <= 2);
        state = _state;
    }

    function get_state() public view returns(uint8)
        {
        return state;
    }
}
```

Listing 1. Proof of concept smart contract.

In these two experiments, we evaluated how the blockchain network blocking time delays and the field device communication latency times behave according to the number of client devices that will request in parallel to change the smart contract to the ISCOM server. This experiment was performed with one, five, ten, and twenty clients simultaneously. Currently, there is an effort to apply wireless networks in the industry, so wireless networks have been used and evaluated for communication to field devices.

Table I shows the specifications of the devices used in these experiments. All devices had Wi-Fi communication from a router in the lab. We use the IEEE 802.11g standard, Request to Send (RTS) / Clear to Send (CTS), and the device's standard transmit (Tx) power of 20 dBm. In theory, the IEEE 802.11g standard is 54 Mbps bandwidth, but in practice, through the iperf tool, we identify 20 to 30 Mbps bandwidth on the LAN.

TABLE I
Specification of the devices, network, and environment used in the experiments.

Device	Processor	RAM	Distance	LAN Bandwidth
Notebook	Intel Core i5-4200 2.60 GHz	8 GB	10 meters	20–30 Mbps
Raspberry Pi 3	4X ARM Cortex-A53 1.2GHz	1 GB	50 meters	20–30 Mbps

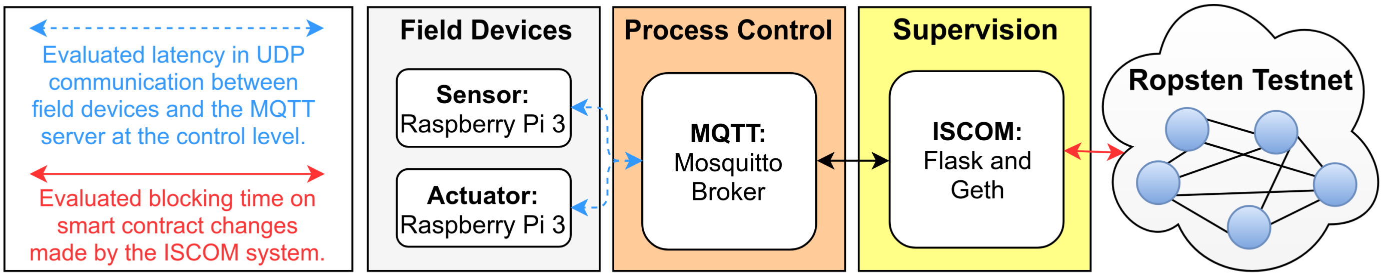
Evaluated latency in UDP communication between field devices and the MQTT server at the control level.
Evaluated blocking time on smart contract changes made by the ISCOM system.
Field Devices
Sensor: Raspberry Pi 3
Actuator: Raspberry Pi 3
Process Control
MQTT: Mosquitto Broker
Supervision
ISCOM: Flask and Geth
Ropsten Testnet

Fig. 3. Illustration of a proof of concept for the ISCOM evaluation environment.

V. Analysis and Discussion of Results

For a better understanding of the experimental results, the boxplot graph was used for the illustration in Figures 4 and 5. The boxplot identifies where 50% of the most likely values (in the box), the median (orange line in the box) and the extreme values (vertical line) are located. The limits (extreme values) in Figure 4 and Figure 5 respectively represent communication latency timeouts and write lock time on a smart contract.

As described in Figure 4, the results of the second experiment point to a high blocking time for changing a smart contract, with time-averaged 25 seconds with a standard deviation of 16 seconds. In some tests, the change of contracts even had a blocking time of more than 1 minute. An analysis of the Ropsten test network showed that this variation is related to the mining time and work proof algorithm. Although changing the work proof algorithm is an alternative to reduce the delay, this can be considered a significant challenge, since the shorter the work proof lock time, the less secure the block is to be inserted into the blockchain.

Therefore, the results of the second experiment present problems with high delay time and high variation of this delay. In industrial systems, this time is not suitable for processes where it can slow down decision making and compromise system time constraints. Compared to real-time M2M communication, the impact of blocking time is the greatest, as industrial real-time systems apply time requirements ranging from 10–100 ms. Thus, the results of both experiments showed that it is not possible to guarantee maximum times.

From the tests performed in this first experiment, it was identified that it is only possible to change the smart contract with only one customer at a time, and not be able to make simultaneous changes. Thus, the results of the latency time experiment show comparisons according to customer numbers in parallel. Already in the results of the blocking time experiment, only the time for one request at a time and not in parallel is presented.

Therefore, the tests conducted show that blockchain-based smart contracts operations always performed without real competition. The explanation for this limitation is that all smart contracts are executed in series by the miners before being attached to the blockchain. These smart contracts are subsequently executed in series by validators to verify that the miners performed them correctly. This serial running operation limits system throughput and does not exploit current concurrent multicore and cluster architectures.

Figure 5 illustrates the results of the second experiment. It can be observed that the latency time increases as the number of field devices increases. This time increment is because the ISCOM server has more work with larger numbers of field devices, so the communication channel gets busier with more packets at a time to be transmitted and processed. Also, another aspect influenced the increase in latency time. As the number of devices increased, the number of occurrences and interferences also increased. So because of these interferences, packets had a longer time to reach their destinations.

Therefore, the results of the second experiment present problems regarding the latency time vary according to the number of field devices. In a factory, multiple devices will be connected to gateways, and these devices require low latency and varying communication. Thus, the integration of IP based wireless networks with the MQTT protocol applied in the process control layer of automation systems has not been suitable for industrial environments.

Based on the results of the experiments and although the latency time in the second experiment was below 100 ms, we conclude that there was a considerable variation of latency and it can affect the communication time between machines seriously. Also, the blocking time shown in smart contract changes is not feasible for general real-time applications, aggravated by non-parallel execution on smart contract calls. Therefore, changes to the Ethereum implementation are required to enable a smart contract change for real-time systems.

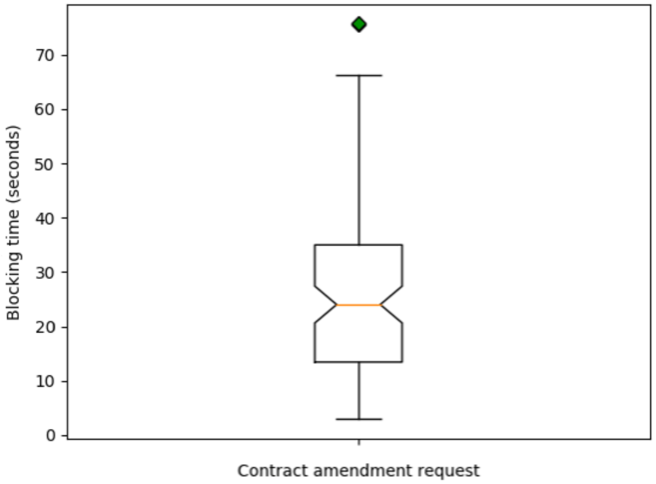

Fig. 4. Results of the blocking time experiment.

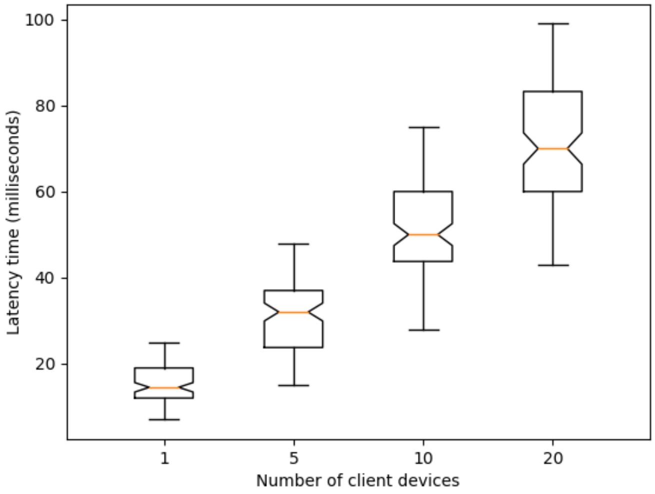

Fig. 5. Results of the latency time experiment.

A. Challenges and Opportunities

Industry 4.0 technologies can benefit from the use of smart contracts, but their application also presents challenges in many ways. Implementing a blockchain today can help cloud-based solutions provide redundancy for storage needs, while at the same time this local blockchain implementation is currently challenging to replicate on IIoT nodes due to its memory constraints and computational.

In this context, it is critical to quantify the amount of computational energy required for a field device to interact directly with the blockchain network through smart contracts rather than using ISCOM middleware. Moreover, the inclusion of ISCOM at the highest levels of industrial control is not sufficient, as communication between field devices and the process control level is extremely insecure and vulnerable to malicious attacks.

Another essential aspect that should be evaluated, is the mobility of field devices. With IIoT devices in constant motion, the communication network will face a high dynamism and consequently large amounts of connectivity failures [22]. This scenario will partition the communication network, reducing the range of communication between the devices and hence reducing communication opportunities with other nearby devices as well as with gateway devices.

Finally, new generations of IIoT devices are expected to be equipped with better hardware specifications, which will allow direct communication with smart contracts, reducing response times [23]. In this context, the smart contract is expected to operate within an environment in which it must adapt its capabilities to the context of the IIoT devices. This scenario will contribute to the viability of the Pervasive Computing [24] paradigm, where devices run applications and integrate seamlessly with field devices.

VI. Final Considerations

The introduction of ISCOM middleware at the IPAS hierarchy supervision level has resulted in total process decentralization and automated communication across the supply chain. Also, through ISCOM it was possible to explore the impacts of blockchain-based smart contracts on a plant's vertical communication structure.

Besides, tests have shown in real experiments that the high and variable blocking time for smart contract changes is sometimes unsuitable for M2M real-time communications, requiring alternatives to meet real-time system requirements.

For future work, simulation scenarios will be extended to assess the ISCOM's behavior in environments with large amounts of IIoT devices. Also, mobility scenarios will be applied in these future studies to evaluate the ISCOM. This evaluation environment will allow an understanding of various aspects of the ISCOM operation for the industrial environment.

Acknowledgment

We acknowledge support from the IFMG and UFOP, and the Brazilian research agencies CNPq, Capes, FAPEMIG.

References

[1] D. Serpanos and M. Wolf, "Industrial internet of things," in *Internet-of-Things (IoT) Systems*, pp. 37–54, Springer, 2018.

[2] Accenture, "Winning with the industrial internet of things." Available online at: https://www.accenture.com/t00010101T000000Z__w__/it-it/_acnmedia/PDF-5/Accenture-Industrial-Internet-of-Things-Positioning-Paper-Report-2015.pdf. Accessed on 04/07/2019, 2015.

[3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[4] N. Kshetri, "Can blockchain strengthen the internet of things?," *IT professional*, vol. 19, no. 4, pp. 68–72, 2017.

[5] C. J. Bartodziej, "The concept industry 4.0," in *The Concept Industry 4.0*, pp. 27–50, Springer, 2017.

[6] T. Borangiu, D. Trentesaux, A. Thomas, P. Leitão, and J. Barata, "Digital transformation of manufacturing through cloud services and resource virtualization," *Computers in Industry*, vol. 108, pp. 150–162, 2019.

[7] S. Zhong, H. Zhong, X. Huang, P. Yang, J. Shi, L. Xie, and K. Wang, *Security and Privacy for Next-Generation Wireless Networks*. Springer, 2019.

[8] R. Candell, M. Kashef, Y. Liu, K. B. Lee, and S. Foufou, "Industrial wireless systems guidelines: Practical considerations and deployment life cycle," *IEEE Industrial Electronics Magazine*, vol. 12, no. 4, pp. 6–17, 2018.

[9] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE industrial electronics magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[10] M. Felser, "Real-time ethernet–industry prospective," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1118–1129, 2005.

[11] S. Vitturi, C. Zunino, and T. Sauter, "Industrial communication systems and their future challenges: Next-generation ethernet, iiot, and 5g," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 944–961, 2019.

[12] K. Sharma, *Overview of industrial process automation*. Elsevier, 2016.

[13] M. Banerjee, J. Lee, and K.-K. R. Choo, "A blockchain future for internet of things security: A position paper," *Digital Communications and Networks*, vol. 4, no. 3, pp. 149–160, 2018.

[14] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[15] A. Miller, Z. Cai, and S. Jha, "Smart contracts and opportunities for formal methods," in *International Symposium on Leveraging Applications of Formal Methods*, pp. 280–299, Springer, 2018.

[16] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.

[17] J. Backman, S. Yrjölä, K. Valtanen, and O. Mämmelä, "Blockchain network slice broker in 5g: Slice leasing in factory of the future use case," in *2018 Internet of Things Business Models, Users, and Networks*, pp. 1–8, IEEE, 2018.

[18] C. H. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for industrial iot with deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, 2018.

[19] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE transactions on industrial informatics*, vol. 14, no. 8, pp. 3690–3700, 2018.

[20] Y. Liu, K. Wang, Y. Lin, and W. Xu, "Lightchain: A lightweight blockchain system for industrial internet of things," *IEEE Transactions on Industrial Informatics*, 2019.

[21] M. Y. Afanasev, Y. V. Fedosov, A. A. Krylova, and S. A. Shorokhov, "An application of blockchain and smart contracts for machine-to-machine communications in cyber-physical production systems," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pp. 13–19, IEEE, 2018.

[22] M. Lucas-Estañ, M. Sepulcre, T. Raptis, A. Passarella, and M. Conti, "Emerging trends in hybrid wireless communication and data management for the industry 4.0," *Electronics*, vol. 7, no. 12, p. 400, 2018.

[23] F. Tramarin, A. K. Mok, and S. Han, "Real-time and reliable industrial control over wireless lans: Algorithms, protocols, and future directions," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1027–1052, 2019.

[24] M. Weiser, "The computer for the 21 st century," *Scientific american*, vol. 265, no. 3, pp. 94–105, 1991.