

IOTA and Smart Contract Based IoT Oxygen Monitoring System for the Traceability and Audit of Confined Spaces in the Shipbuilding Industry [†]

Ángel Niebla-Montero ^{1,2,*} , Iván Froiz-Míguez ^{1,2} , José Varela-Barbeito ³, Paula Fraga-Lamas ^{1,2,*} 
and Tiago M. Fernández-Caramés ^{1,2} 

¹ Centro Mixto de Investigación UDC-Navantia, Universidade da Coruña, Edificio de Batallones, s/n, 15403 Ferrol, Spain; ivan.froiz@udc.es (I.F.-M.); tiago.fernandez@udc.es (T.M.F.-C.)

² Department of Computer Engineering, Faculty of Computer Science, Universidade da Coruña, 15071 A Coruña, Spain

³ Navantia S. A., Unidad de Producción de Ferrol, Taxonera, 15403 Ferrol, Spain; jvarela@navantia.es

* Correspondence: angel.niebla@udc.es (Á.N.-M.); paula.fraga@udc.es (P.F.-L.); Tel.: +34-981167000 (ext. 6051) (P.F.-L.)

[†] Presented at the 10th International Electronic Conference on Sensors and Applications (ECSA-10), 15–30 November 2023; Available online: <https://ecsa-10.sciforum.net/>.

Abstract: Security presents significant challenges due to the exponential growth in the number of Internet of Things (IoT) devices that generate and collect data over the network. It is crucial to ensure the integrity and security of IoT devices, as well as to address issues such as interoperability and trust in data sources. In the proposed article, we present a novel architecture together with its implementation as a proof-of-concept of a traceability and auditing IoT system based on Distributed Ledger Technology (DLT). To demonstrate the applicability of the proposed solution, a smart contract-based system for occupational risk prevention (ORP) has been developed to monitor oxygen concentration in confined spaces that exist in ships and shipyards. The system has been devised for the operators that weld inside the ships of the Spanish shipbuilding company Navantia, which is one of the largest shipbuilders in the world. Specifically, the IOTA network has been used, which benefits the system through its decentralized, secure, and scalable data structure. In addition, the integration of smart contracts allows for establishing predefined rules and conditions, ensuring the execution of logic in a reliable and automated manner. To demonstrate the viability of the system, it has been tested locally and in the IOTA testing environment. Despite the challenges in deploying smart contracts with IOTA, the developed system is considered useful for the traceability and auditing of the oxygen concentrations without the need for any human intervention. Furthermore, it establishes the groundwork for future advancements in IoT traceability and auditing in industrial ORP scenarios.

Keywords: IoT; IOTA; smart contract; Industry 5.0; occupational risk prevention



Citation: Niebla-Montero, Á.; Froiz-Míguez, I.; Varela-Barbeito, J.; Fraga-Lamas, P.; Fernández-Caramés, T.M. IOTA and Smart Contract Based IoT Oxygen Monitoring System for the Traceability and Audit of Confined Spaces in the Shipbuilding Industry. *Eng. Proc.* **2023**, *58*, 120. <https://doi.org/10.3390/ecsa-10-16226>

Academic Editor: Stefano Mariani

Published: 15 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In a traditional database audit, the auditor must physically visit the servers to thoroughly analyze them. This is because, in a client/server architecture, there is no way to know the internal server operations, requiring a thorough examination of its internal workings. On the contrary, the use of Distributed Ledger Technology (DLT) systems provides a significant advantage in terms of transparency [1]. The ability to execute smart contracts in these systems allows any entity involved in the construction of a ship or in its maintenance processes to be fully aware of the actions carried out during the smart contract execution.

The objective of this paper is to describe an Internet of Things (IoT) system for traceability and occupational risk prevention (ORP) auditing that has been developed for the Spanish shipbuilder Navantia. The proposed system is designed for the monitoring and

control of oxygen concentrations in the confined spaces found on ships and shipyards, where operators perform welding tasks.

In the proposed system, initially, the involved parties sign a smart contract in which the terms of use are accepted and the audited database is created. Then, any participating entity will be able to automatically carry out the audit of the agreed-upon database by downloading it via another smart contract. Specifically, the IOTA [2] network has been used to implement a smart contract that, when deployed, creates an OrbitDB [3] database where the data of the monitored parameters will be stored.

The choice of IOTA over other blockchain-based systems is related to the fact that IOTA nodes only perform basic operations that do not require a lot of computational power (for example, operations like storing transactions on a shared ledger, or validating such transactions). As a result, its application in IoT projects is really promising. At this point, it is important to note that IOTA does not use a blockchain, but a Distributed Ledger Technology (DLT) that makes use of Direct Acyclic Graphs (DAGs).

Specifically, the development described in this paper has the following features:

- Smart contract-based. At the first stage, all involved parties sign a contract that outlines the specific terms and conditions for the creation of the database, which will be audited and shared among them.
- Decentralized storage. The database is distributed across multiple nodes using a decentralized storage approach. Each node keeps an immutable copy of the content.
- Traceability for audits. If one of the involved entities wants to validate or audit the stored data, it can easily access the database by executing a specific smart contract. Such an action allows for downloading the latest stored content.

2. Communications Architecture

The proposed communications architecture is divided into two basic layers, each serving a specific purpose:

- Low-Level Layer: This layer, shown in Figure 1, is responsible for managing communications between nodes, represented as Raspberry Pi (RPI) devices, through Bluetooth Low-Energy (BLE) connections. A client-server connection is established via a BLE interface, for connection to Low Power Node (LPN) nodes, and libp2p, an essential component of Interplanetary File System (IPFS) [4], is used for transport layer to ensure data security between peers. Libp2p ensures that each peer has a unique identifier known as a PeerId, which allows any entity to retrieve the public key of the node with which it communicates. This guarantees peer identity and protects the data from unauthorized access or manipulation by third parties.
- High-Level Layer: The IOTA network is established in this layer via the data network known as the Tangle [5]. The Tangle network is a decentralized structure that records data exchanges in an immutable way, ensuring the reliability and integrity of the stored data. One of the key differences between IOTA and other blockchain-based technologies is that there is no distinction between miners and users. All IOTA network nodes can participate in the consensus process, thus promoting the decentralization and scalability of the system. This layer is illustrated in Figure 2.

The definition of a consensus layer, which determines the validity of transactions through mutual approval, is at the heart of consensus in IOTA. Transactions gain more consensus as more subsequent transactions validate them. The proposed architecture includes IOTA nodes, called “Wasps”, that are in charge of executing smart contracts in the Tangle. A critical node known as “GoShimmer” [6] is also deployed to provide network access to the other Wasp nodes. A committee of randomly selected Wasp nodes runs a chain where smart contracts are deployed. The committee’s primary goal is to ensure a smooth transition from one state to the next one, thus enabling distributed, fault-tolerant, and leaderless operations. The IOTA Smart Contracts Protocol (ISCP) enables the private execution of smart contract chains, which are validated by a committee of nodes. Unlike

traditional systems, IOTA smart contracts do not need the execution of all contracts by all nodes in the network, providing greater flexibility and significantly reducing energy consumption and cost [7].

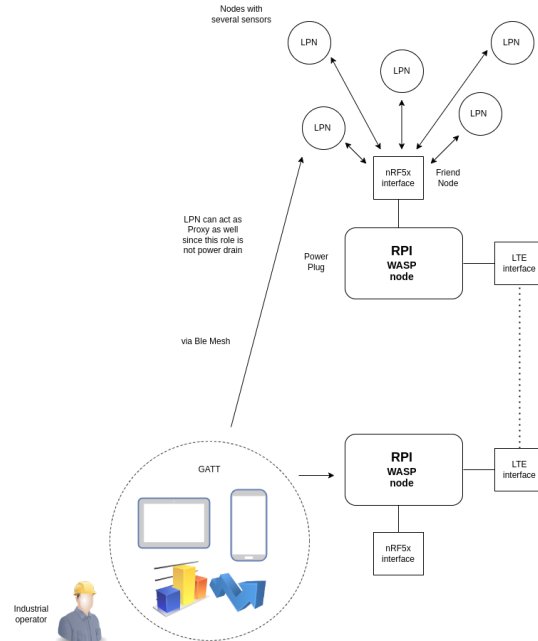


Figure 1. Low-level layer of the communications architecture.

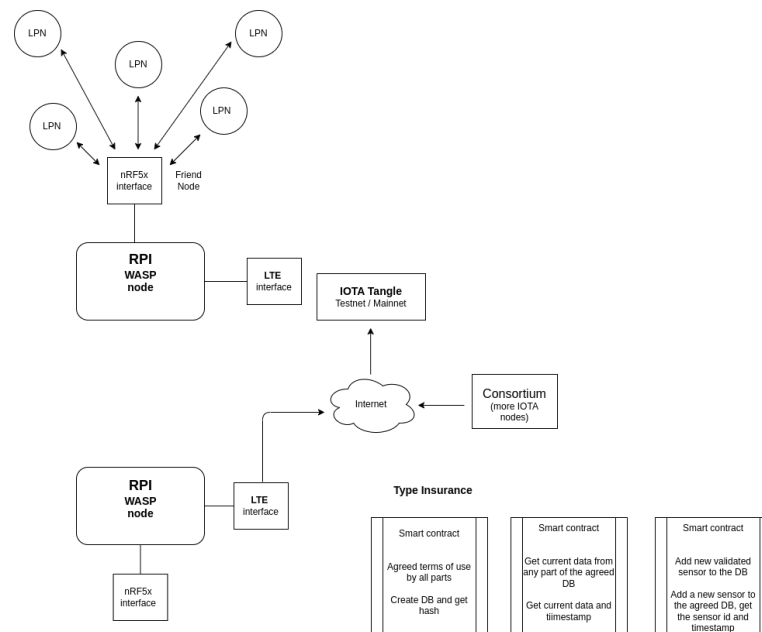


Figure 2. High-level layer of the communications architecture.

3. Implementation

The high-level layer of the architecture is the focus of this paper. The high-level layer includes a locally implementable IOTA private network that can later be integrated into the nodes that constitute the oxygen monitoring system (one such node is shown in Figure 3). The implementation consists of a GoShimmer node for messaging and a Wasp node that runs smart contracts. The proposed architecture is deployed in a series of steps that configure and activate the key elements in an IOTA 2.0 private network.

First, the GoShimmer node (version 0.7.5) is installed, which serves as a central component for network messaging. The client wallet is then downloaded and initialized,

producing a seed that will be used to allocate the required funds. The IOTA network wallet is a must-have tool for token access, transfer and management. It provides a solution for storing secret keys and using them to control tokens in the network. The lack of transaction fees is a distinguishing feature of IOTA [8]. For each transaction, the amount deducted from the sender's wallet equals the amount added to the recipient's wallet.

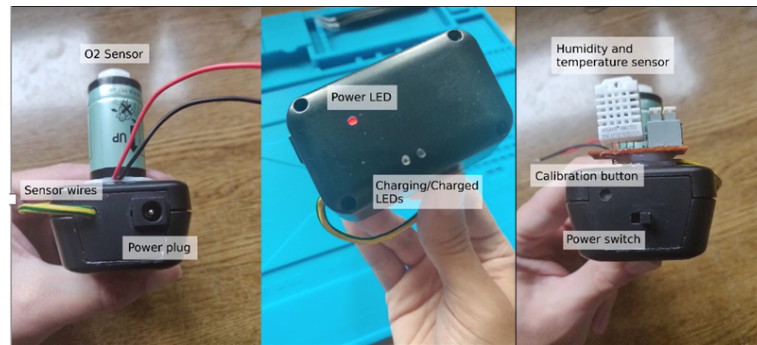


Figure 3. Oxygen level monitoring node.

The next step is to create a custom Genesis snapshot, which includes a transaction that directs funds to the newly initialized wallet. After the GoShimmer node is up and running, the Wasp node is installed, which is critical for executing smart contracts on the Tangle.

The smart contract is created using the Go programming language. A contract is created, and a Processor is established to allow the contract to be initialized with the necessary functions, such as “initialize” (used during contract deployment) and “getOxygen” (providing database values). The “initialize” function receives a value that can be provided as a parameter during contract deployment, and the database is initialized in the system and the received value is added via the “initOrbit” function. The code for the three functions mentioned above can be found in [9].

Following that, the smart contract is imported into the Wasp node and executed when the node boots. After starting the Wasp node, funds are transferred to the Wasp node wallet using the address obtained when starting the node.

Finally, to deploy the smart contract, a chain with a single Wasp node is first built. The funds are then deposited in this chain, allowing the smart contract to be executed later. Finally, the smart contract is deployed in the chain, completing the implementation process of the proposed architecture.

4. Initial Experiments

To verify the proper operation of the proposed system, tests took place in a real-life scenario at Navantia's shipyard in Ferrol. Specifically, one of the previously mentioned nodes (whose main hardware components are displayed in Figure 3) was placed inside a tank of a gas carrier (as shown in Figure 4a,b), since such a scenario is an example where dangerous soldering works can be performed. Data on the percentage of oxygen inside the gas carrier was obtained and stored in the database during these tests, resulting in the creation of a smart contract.

The use of an off-chain environment, such as the one described in Section 3, allows for more agile development when implementing the system. However, once a preliminary version of the system has been developed, it is recommended that it should be tested in a more realistic environment. In this case, IOTA provides a Tangle Devnet environment. To use the IOTA devnet, it must first be deployed the GoShimmer node. Such a node must have a fixed or dynamic public IP address that is not Carrier Grade (this is the primary requirement for synchronization with the Tangle). A GoShimmer node can involve multiple Wasp nodes, implying that multiple committee nodes can use the same GoShimmer node to connect to the Tangle. The Wasp nodes, which can be deployed in any type of network, must be linked to the GoShimmer node for proper operation in the IOTA devnet. To run Smart

Contracts on a GoShimmer node, several plugins must be enabled (remotelog, networkdelay, spammer, prometheus, faucet, and txstream) and TCP port 5000 must be exposed.



Figure 4. Gas carrier where the system was tested (a) and ORP IoT device placed inside a tank of the gas carrier (b).

Then, the GoShimmer node must synchronize its state with the rest of the network when it boots up for the first time. This step is critical because, without it, it will be impossible to request funds, which are required to carry out any operation in the chain. It should be noted that only the most recent version of the GoShimmer node will be synchronized. Once synchronized, internal operations will be performed for load testing and for measuring response times, but they have nothing to do with the use of smart contracts. Figure 5 depicts the synchronized GoShimmer node dashboard [10], showing its performance in operation in terms of messages per second (the amount of messages issued to the network every second) and current tips (transactions that have not yet been approved). The latter is related to the fact that, when a transaction is added to the tangle, it must reference two other transactions (tips) that are already in the Tangle: if the current tips parameter is less than two, the transaction has to wait.

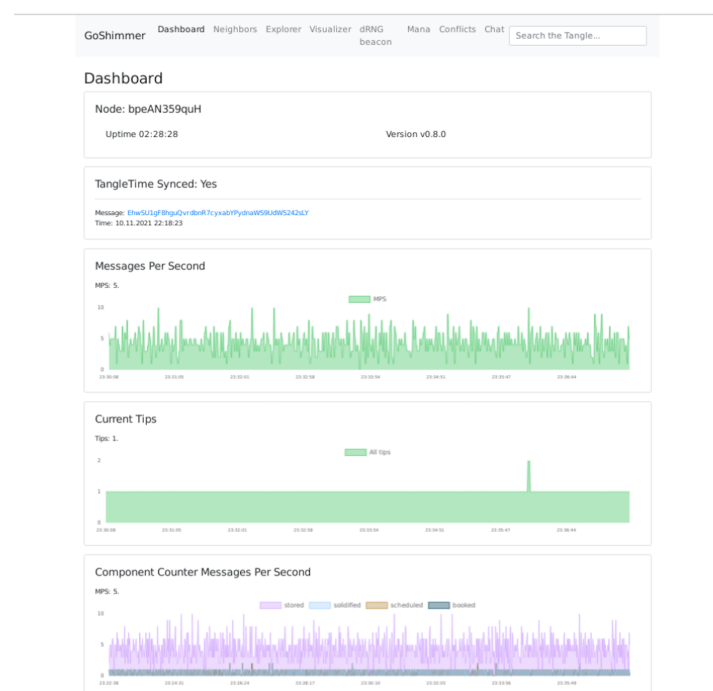


Figure 5. Synchronized GoShimmer node dashboard.

5. Conclusions

This paper presented a novel architecture, together with its implementation, as a proof-of-concept of a traceability and auditing IoT system for monitoring oxygen concentration in confined spaces. Specifically, the paper described a smart contract-based system for occupational risk prevention for monitoring oxygen concentration in confined spaces found in ships and shipyards.

A smart contract was successfully implemented both locally and in the IOTA test environment, and the overall system was tested in a real-world scenario (inside a gas carrier). Given the technology's current state of development, both the environment configuration and smart contract development were performed in IOTA's native language. The initial tests demonstrated the viability of the proposed approach in real shipbuilding scenarios, like inside a tank of a gas carrier.

In conclusion, this project has laid the basis for a more comprehensive and robust solution in the field of IoT traceability and auditing, and it is expected to contribute to future developments and improvements in the IOTA infrastructure, as well as the deployment and operation of smart contracts.

Author Contributions: conceptualization, T.M.F.-C., J.V.-B.; methodology, P.F.-L. and T.M.F.-C.; investigation, A.N.-M., I.F.-M., P.F.-L. and T.M.F.-C.; writing—original draft preparation, A.N.-M., I.F.-M. and T.M.F.-C.; writing—review and editing, A.N.-M., I.F.-M., P.F.-L. and T.M.F.-C.; supervision, T.M.F.-C., J.V.-B.; project administration, T.M.F.-C.; funding acquisition, T.M.F.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by Centro Mixto de Investigación UDC-NAVANTIA (IN853C 2022/01), funded by GAIN (Xunta de Galicia) and ERDF Galicia 2021-2027. In addition, this work has been funded by the Xunta de Galicia (by grant ED431C 2020/15), and by grant PID2020-118857RA-100 (ORBALLO) funded by MCIN/AEI/10.13039/501100011033.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable due to privacy.

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

References

1. Fernández-Caramés, T.M.; Fraga-Lamas, P. Design of a fog computing, blockchain and IoT-based continuous glucose monitoring system for crowdsourcing mHealth. In Proceedings of the 5th International Electronic Conference on Sensors and Applications, Online, 15–30 November 2018.
2. IOTA Wiki. Available online: <https://wiki.iota.org/learn/about-iota/an-introduction-to-iota/> (accessed on 27 September 2023).
3. OrbitDB GitHub Repository. Available online: <https://github.com/orbitdb/orbitdb> (accessed on 27 September 2023).
4. Official IPFS Website. Available online: <https://ipfs.io/> (accessed on 27 September 2023).
5. Popov, S. The tangle. *White Pap.* **2018**, *1*, 30.
6. IOTA Wiki. Setting Up a GoShimmer Node. Available online: <https://wiki.iota.org/goshimmer/tutorials/setup/> (accessed on 27 September 2023).
7. IOTA Wiki. Energy Efficiency. Available online: <https://wiki.iota.org/learn/about-iota/energy-efficiency/> (accessed on 27 September 2023).
8. Popov, S.; Lu, Q. IOTA: Feeless and free. *IEEE Blockchain Tech. Briefs* **2019**. Available online: <https://blockchain.ieee.org/technicalbriefs/january-2019/iota-feeless-and-free/> (accessed on 27 September 2023).
9. Repository with the Code of the Smart Contracts of the Project. Available online: <https://git.armriot.com/umi/autoid/blockchain> (accessed on 27 September 2023).
10. IOTA Wiki. GoShimmer Analysis Dashboard. Available online: https://wiki.iota.org/goshimmer/teamresources/analysis_dashboard (accessed on 27 September 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.