MoTI: An MQTT-Oriented Architecture Enabling Interoperability of IoT Devices in Industry 4.0

Camila C. de Barros, Arlindo F. da Conceição, and Vladimir E. M. Rocha

***Abstract*—The increasing adoption of Industrial Internet of Things (IIoT) systems intensifies the need for secure and interoperable communication across heterogeneous industrial environments. This paper proposes MoTI (MQTT-oriented Things Interoperability), a modular architecture that integrates MQTT-based communication, mutual TLS authentication, structured metadata logging, and decentralized storage through IPFS. The solution was implemented and experimentally validated using 1500 messages distributed across three Quality of Service (QoS) levels under controlled load conditions. Considering the complete end-to-end processing pipeline, including MQTT transmission, metadata registration, IPFS persistence, and acknowledgment, statistical analysis revealed significant variability differences across QoS levels. QoS levels 1 and 2 exhibited stable and predictable RTT behavior, whereas QoS level 0 showed substantially higher dispersion and extreme outliers. The results demonstrate that stronger delivery guarantees can be adopted without significant impact on average response time, reinforcing the suitability of QoS 2 for delay-sensitive industrial applications. MoTI provides a practical, experimentally validated approach for secure, traceable, and scalable cross-organizational data exchange in distributed industrial environments.**

Link to graphical and video abstracts, and to code:
https://latamt.ieeer9.org/index.php/transactions/article/view/10400

***Index Terms*—Interoperability, Internet of Things, Industry 4.0, Decentralization, IPFS, MQTT.**

I. Introduction

In recent decades, technological advancements have driven significant transformations in the industrial sector. Industry 4.0 marks a new era in automation and production control, characterized by the integration of technologies such as the Internet of Things (IoT), cloud computing, artificial intelligence, and blockchain. In this context, IoT devices play a central role by enabling real-time data collection, transmission, and analysis to optimize operations, reduce costs, and increase productivity.

The United Nations Conference on Trade and Development (UNCTAD) developed the Technology Readiness Index to assess the adoption of emerging technologies across countries, considering technological infrastructure, industrial R&D capacity, skilled labor, and access to funding. According to this index, Brazil has gradually improved its position, ranking 41st in 2021, 40th in 2023, and 38th in 2025, leading among developing countries with similar per capita income. Nevertheless, the ranking highlights the underexplored potential of Brazil's industrial sector in adopting emerging technologies [1].

Although Brazil leads its income group in technological readiness, significant challenges remain in implementing Industry 4.0, particularly due to the limited adoption of enabling technologies by domestic industries. This scenario presents an opportunity for research and innovation. As Moura states, "Technological dominance is a decisive factor for the competitive position of nations and companies" [2].

The integration of these technologies into industrial information systems introduces new research demands, particularly regarding the quality, integrity, and accessibility of data generated by sensors and connected devices. The lack of interoperability between IoT devices from different manufacturers and organizations hampers efficient data integration, creating both technical and organizational barriers. Furthermore, issues related to security, reliability, and scalability in information sharing become increasingly critical. Recent studies reinforce the importance of solutions that ensure secure and interoperable communication, especially in environments composed of devices from diverse manufacturers, protocols, and architectures [3].

This paper extends previous research entitled "Integration of MQTT Services Using Blockchain" [4], which introduced an initial and exploratory version of the proposal, then referred to as MobI. The present work represents a substantial evolution in scope and technical depth, incorporating a mature architecture, enhanced security mechanisms, decentralized storage, and experimental performance evaluation not included in the earlier study.

Based on these premises, this paper presents MoTI (MQTT-oriented Things Interoperability), an interoperable communication architecture designed for Industry 4.0 environments. MoTI addresses integration barriers among heterogeneous industrial IoT devices and promotes decentralized data storage through the InterPlanetary File System (IPFS). The architecture adopts the Message Queuing Telemetry Transport (MQTT) protocol, widely used in IoT applications due to its lightweight design and efficiency in resource-constrained networks. Communication is managed through differentiated Quality of Service (QoS) levels, with level 2 applied to critical messages to ensure reliable delivery. This strategy enables message deduplication, delivery confirmation, and resilience in environments subject to network instability.

The MoTI solution seeks to ensure reliability, accessibility,

The associate editor coordinating the review of this manuscript and approving it for publication was Mónica K. Huerta (*Corresponding author: Camila Cabral de Barros*).

C. C. Barros, and A. F. Conceição are with the Department of Science and Technology, Federal University of São Paulo, São José dos Campos, Brazil (e-mails: barros.camila@unifesp.br, and arlindo.conceicao@unifesp.br).

V. E. M. Rocha is with the Center for Mathematics, Computation and Cognition, Federal University of ABC, Santo André, Brazil (e-mail: vladimir.rocha@ufabc.edu.br).

and traceability of the information exchanged between different industrial systems, while preserving scalability and security in operations. It represents a practical and viable initiative in response to national and global challenges concerning the adoption of enabling Industry 4.0 technologies, particularly in terms of interoperability and decentralized data flows.

The remainder of this paper is organized as follows. Section II outlines the challenges of achieving secure and interoperable communication among heterogeneous IoT devices in industrial environments. Section III reviews recent research on interoperability, MQTT communication, and decentralized architectures, identifying existing limitations and motivating the proposed solution. Section IV presents the MoTI architecture, detailing its components, implementation, and validation. Section V provides an empirical evaluation, including end-to-end response time analysis, QoS comparisons, variability assessment, and stress testing under increased message volume. Section VI describes practical application scenarios emphasizing interoperability and data privacy. Finally, Section VII presents concluding remarks and future research directions.

II. Problem Statement

The architecture developed in this project is intended for application in industrial environments, with a particular focus on production lines where sensors and IoT devices are integrated with machines and control systems. However, its scope is not limited to the shop floor (i.e., the factory production area). It also extends to administrative areas, such as management and executive departments. Additionally, manufacturers, suppliers, and service providers can access the solution's interface and the data it generates, in accordance with predefined authorization levels.

In a modern supply chain, it is essential to establish robust interoperability capable of integrating data and processes across different organizations and industrial ecosystems. This requires efficient interconnection among supply chain actors and the integration of production environments based on smart systems that frequently employ devices from various manufacturers using different languages and standards.

This scenario underscores the need for a secure, interoperable, and scalable communication architecture capable of operating across heterogeneous systems. Today, many IoT devices used in industry follow different standards and lack effective interoperability, making data sharing between companies, suppliers, and service providers difficult. Therefore, a solution is needed that extends beyond basic device connectivity. It must support access control, message standardization, and data prioritization to ensure reliable operation in complex, multi-stakeholder environments.

III. Literature Review

Several studies related to the integration of IoT device communication using the MQTT protocol have been proposed by researchers. The reviewed literature analyzes interoperability across multiple technological domains, including the Internet of Things, cloud computing, artificial intelligence, data storage, communication networks, and smart manufacturing. These works highlight the challenges involved in integrating different platforms, devices, and systems, as well as solutions to ensure efficient and secure information exchange. The key focus areas include:

- **IoT and Communication Networks:** Methods for connecting heterogeneous devices while ensuring compatibility and efficiency.
- **Security:** Strategies for integrating different networks and improving the reliability of data exchange.
- **Cloud Computing and AI:** Standards to facilitate interoperability between service providers and machine learning models.

Jacob et al. [5] analyze interoperability in distributed artificial intelligence systems, proposing methods to ensure efficient communication among intelligent agents. Sivakumar et al. [6] focus on interoperability in machine learning systems, emphasizing the standardization of data formats and models to support algorithm reuse and integration. Falahi et al. [7] investigate interoperability in augmented reality (AR) systems and suggest frameworks for integrating various AR platforms and devices.

Roy et al. [8] propose a comprehensive solution to IoT-related attacks using blockchain combined with a QoS monitoring system for congestion control. A mobile application was developed to present users with different QoS indicators to improve understanding. The MQTT Mosquitto broker is used to mediate data transmission between consumers and cloud providers. The authors introduce a consensus mechanism called Proof of Interoperability (PoI), intended to address some limitations of the traditional Proof of Work (PoW) model. This mechanism acts as a controller of the computational effort needed to perform essential tasks, such as verifying whether communications meet structured and semantic constraints.

Focusing on interoperability in smart manufacturing systems, Liu et al. [9] propose an approach to integrate machines and control systems from different vendors. Salzano et al. [10] address interoperability in big data analytics systems, proposing methods to integrate and analyze data from diverse sources.

Chen et al. [11] focus on effective management of data generated by IoT devices and enhanced interoperability among trusted IoT groups. They propose a decentralized IoT data management platform based on Hyperledger Fabric blockchain and smart contracts. Data from IoT devices are sent via the Kafka MQTT broker to a node called the Management Hub, which interfaces with the blockchain network by translating incoming CoAP messages into JSON-RPC. The system also provides compliance-based authorization for privacy protection, allowing data owners to define consent policies and record all data operations using encryption and smart contracts on the blockchain.

In a different approach, Taherkordi et al. [12] explore interoperability among IoT smart contracts through microservices developed in different programming languages. The proposed solution supports interoperability by interpreting transactions produced by each smart contract into a general, blockchain-traceable format.

Although the reviewed studies emphasize the importance of interoperability, security, and blockchain-based mechanisms, important gaps remain in the literature. First, few works provide a complete end-to-end implementation combining MQTT communication with decentralized storage mechanisms such as IPFS in real industrial contexts. Second, experimental validation is often limited to conceptual evaluations or performance metrics restricted to blockchain layers, without a detailed analysis of round-trip time (RTT) across different QoS levels in heterogeneous environments. Third, cross-organizational interoperability is frequently discussed at a conceptual level, but rarely demonstrated through practical architectures capable of integrating devices, users, and institutions with differentiated access control and verifiable data trails.

In particular, the combined use of MQTT communication, structured metadata logging, and distributed content-addressed storage remains underexplored in Industry 4.0 applications, especially when supported by quantitative performance validation under realistic load conditions.

TABLE I
COMPARATIVE ANALYSIS OF RELATED ARCHITECTURES

Criteria	Roy	Chen	Liu	Taherkordi	MoTI
MQTT communication	Yes	Yes	No	No	**Yes**
Cross-organizational interoperability	Partial	Partial	Partial	Partial	**Yes**
TLS/mTLS authentication	Not specified	Partial	Not specified	Not specified	**Yes**
Decentralized storage	Blockchain	Blockchain	Blockchain	Blockchain	**IPFS**
IPFS integration	No	No	No	No	**Yes**
Structured metadata logging	Limited	Partial	Limited	Limited	**Yes**
RTT experimental evaluation	No	No	No	No	**Yes**
QoS-level analysis	Partial	No	No	No	**Yes**
Stress testing under load	No	No	No	No	**Yes**
Practical end-to-end implementation	Limited	Conceptual	Framework	Conceptual	**Yes**

Table I highlights that existing approaches focus on isolated aspects such as blockchain-based traceability, MQTT communication, or conceptual interoperability frameworks. In contrast, MoTI integrates secure MQTT communication, IPFS-based decentralized storage, structured metadata logging, differentiated QoS analysis, and statistical RTT validation within a unified and operational architecture.

To address these limitations, the next section presents the MoTI solution, designed to integrate secure MQTT communication, QoS-based prioritization, mutual TLS authentication, decentralized IPFS storage, and experimental RTT evaluation within a single interoperable framework.

IV. MOTI SOLUTION

This work presents a solution focused on achieving communication interoperability among industrial IoT devices from different organizations and systems, using the MQTT protocol. Named MoTI (MQTT-oriented Things Interoperability), the architecture enables secure, reliable, and scalable integration across heterogeneous devices, with support for message differentiation through QoS levels.

The MoTI solution enables multiple stakeholders, including companies, institutions, and service providers, to securely access data generated by industrial IoT devices. The architecture is compatible with legacy environments and integrates decentralized storage through IPFS to ensure distributed data persistence and controlled access beyond the local network boundary.

Critical messages, transmitted with QoS level 2, are prioritized and authenticated by a dedicated component that applies SHA-256 hashing to ensure data integrity.

Fig. 1 presents the proposed architecture, whose main components are detailed below.

- **Admin:** Represents the administrative client responsible for initial network setup, device registration, permission configuration, and overall application monitoring. The Admin has full access to all architecture features.
- **User:** Represents end users who can access published data in an authenticated manner. User access is restricted to authorized devices and datasets.
- **IoT Devices:** These elements are responsible for collecting and sending industrial data (such as sensor and actuator outputs) to the MQTT broker. Devices may belong to different companies, institutions, or providers, reinforcing the system's interoperability.
- **MQTT Broker:** The central component that manages asynchronous communication between devices and clients. It handles topic subscriptions, routes messages according to defined QoS levels, and ensures message delivery as specified by the service quality agreement.
- **Interceptor:** Acts as an intermediary between the broker and the remaining modules. It inspects, classifies, and redirects incoming messages based on their criticality level. Messages with QoS level 2 are forwarded to the Authenticator, while those with QoS 0 or 1 are routed directly to the storage module.
- **Authenticator:** Adds an extra layer of security for critical messages (QoS 2), applying SHA-256 hashing to ensure data integrity. The generated hash can later be validated by authorized recipients.
- **IPFS:** Responsible for the decentralized storage of data generated by IoT devices. By adopting IPFS, the architecture enables data persistence beyond the local network, supporting scalability, availability, and fault tolerance.

A. Structure

Fig. 2 presents the sequence diagram of the MoTI operation, illustrating the process from participant registration to data publication, verification, and retrieval. The flow varies depending on the applied QoS level.

Each step is briefly described below to clarify its role in the overall process:

- **DeviceRegistration:** The administrator registers each IoT device with the MQTT broker (Mosquitto), defining its publishing topics, desired QoS level, and the digital certificates required for authentication.
- **UserRegistration:** The administrator also registers authorized users, issuing their digital certificates based on a locally managed Certificate Authority (CA).

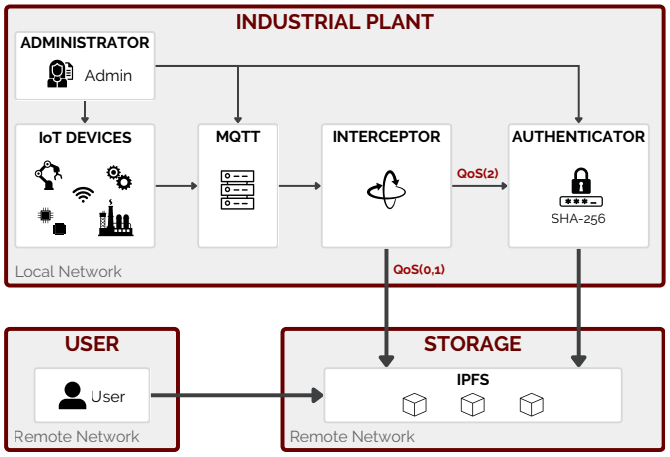

Fig. 1. MoTI architecture showing components, QoS-based routing, and IPFS storage

- **AccessRegistration:** Based on the previous registrations, access rights are defined. Mutual TLS authentication ensures that only authorized agents are allowed to publish or consume data.
- **SendData – IoT Device to MQTT:** The IoT device publishes messages to the Mosquitto broker via MQTT, using mutual TLS to ensure encryption and authentication. The QoS level may vary between 0, 1, or 2, depending on the criticality of the data.
- **SendData – MQTT to IPFS (QoS 0 and 1):** Messages published with QoS 0 or 1 are intercepted by a subscriber (interceptor), which validates the broker's certificate and forwards the data to IPFS over HTTPS with mTLS. In this case, data integrity is implicitly trusted due to the secure connection.
- **RequestData – IPFS (QoS 0 and 1):** When a user requests non-critical data (QoS 0 or 1), the system verifies authenticity via TLS and retrieves the content from IPFS using the previously stored hash (CID).
- **SendData – MQTT to IPFS (QoS 2):** For critical data, the subscriber forwards messages to IPFS with an additional step: a SHA-256 hash is calculated to ensure data integrity between the source and destination. This hash is included in the JSON metadata associated with the CID.
- **RequestData – IPFS (QoS 2):** When retrieving data stored with QoS 2, the system authenticates the user and verifies the SHA-256 hash to ensure that the data received from IPFS exactly matches the data originally sent, mitigating any risk of tampering.
- **CID and Metadata Logging:** After sending data to IPFS, the generated CID is registered remotely along with associated metadata (topic, payload, QoS, timestamp, and SHA-256 hash) in a data.jsonl file stored in the EC2 backend. This step ensures traceability and enables future queries.

B. Authentication and Security

Protecting communication and data in industrial IoT systems is essential to ensure information integrity, confidentiality, and authenticity. In the MoTI solution, security is embedded from the architectural design phase through practical implementation, encompassing multiple layers and agents, each with specific authentication and access control mechanisms. The adopted strategies for each critical component are described below.

- **IoT Devices:** Devices acting as publishers are responsible for generating and sending data to the MQTT broker. Each device uses valid digital certificates, previously issued and configured, to establish mutual TLS authentication with the broker. This approach prevents unauthorized devices from publishing data and ensures that the communication channel is encrypted, protecting data in transit from interception or tampering.
- **MQTT Broker:** The broker used (Mosquitto) is configured to require valid client certificates (`require_certificate true`), reinforcing the security model based on mutual TLS (mTLS) authentication. This ensures that both publishers and any subscribers wishing to consume data must undergo identity verification, validated by a locally managed Certificate Authority (CA) within the MoTI infrastructure.
- **IPFS (QoS 0 and 1):** For data transmitted with QoS levels 0 or 1, which do not guarantee exactly-once delivery (at most once and at least once), the IPFS layer is secured using HTTPS and client certificates. Access to the IPFS API endpoint is routed through a reverse proxy (Nginx) configured with mTLS authentication, requiring that only clients with trusted certificates may add files to the distributed network. This prevents malicious insertion attacks and ensures strict access control to the storage layer.
- **IPFS (QoS 2):** When data is sent with QoS level 2 (exactly once), additional verification steps are applied beyond the previous protections. A SHA-256 hash is computed for each message to ensure its integrity prior to persistence. This guarantees that the stored data exactly matches the original content, preventing tampering or duplication.
- **User:** Access to data persisted in IPFS is managed through a remote log file maintained on the server, which stores metadata such as timestamp, topic, payload, and CID. This file can be accessed via SSH using an authorized private key, ensuring that only pre-registered users with proper credentials can consult the full records. Additionally, CIDs allow public access to information via IPFS gateways, maintaining data traceability without compromising write control.

The combination of these mechanisms results in a robust architecture with end-to-end security, tailored to the requirements of sensitive and interoperable industrial applications.

C. Execution

The MoTI solution repository, available at https://github.com/Camila-Barros/IEEE-MoTI, provides the technical resources required for system deployment and

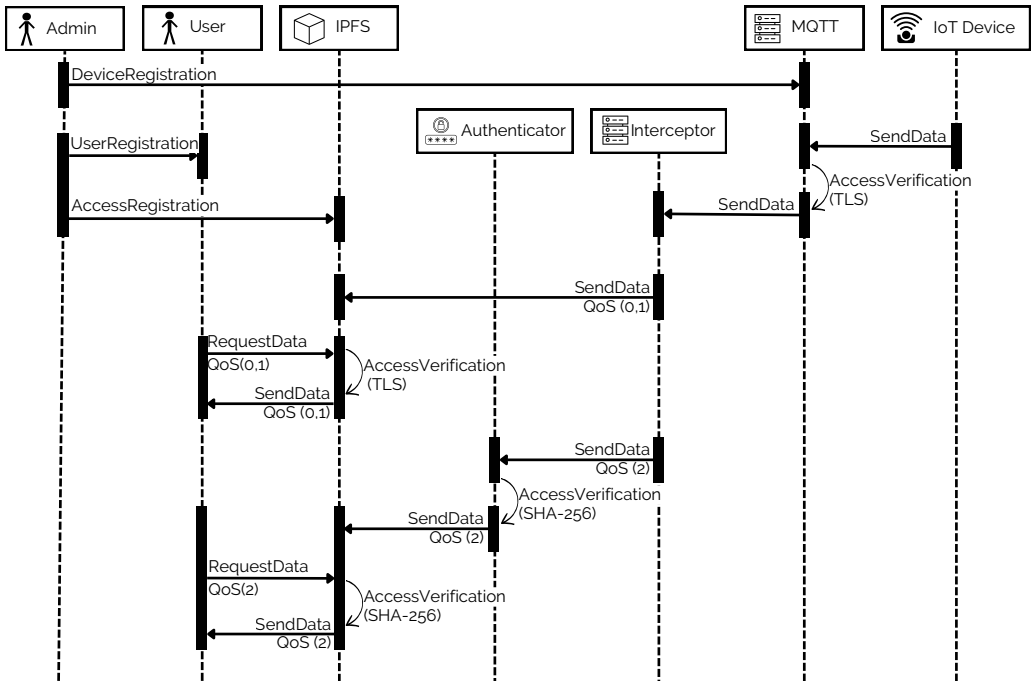

Fig. 2. MoTI sequence diagram including registration, communication, and IPFS storage

operation, including source code, configuration files, and usage documentation.

Below are excerpts from MQTT publishing and subscribing service logs that illustrate the behavior of QoS level 2 during message exchange. In this flow, each message is validated, enriched with metadata, forwarded to IPFS, and confirmed via an acknowledgment containing the same identifier, enabling accurate calculation of round-trip time (RTT). These logs make it possible to observe the MQTT control sequence in practice, offering clearer insight into how message confirmation is handled internally.

The publisher sends a JSON payload containing four variables (temperature, pressure, torque, and humidity), simulating data from an IoT device installed in "machine 1" of a "Beta factory", as shown below:

```
nov 21 23:05:17 Started moti-publisher.service -
MoTI MQTT Publisher (BetaFactory/machine1).

nov 21 23:05:20 Sending PUBLISH (d0, q2, r0, m2),
'b'BetaFactory/machine1/temperature'', (107 bytes)
nov 21 23:05:20 Publishing: temperature=31.75
(QoS=2, id=20a8040e-4fb9-4379-bcf4-15bd6a775dd2) in
BetaFactory/machine1/temperature
nov 21 23:05:23 RTT temperature=31.75
(id=20a8040e-4fb9-4379-bcf4-15bd6a775dd2)=>2525.5ms

nov 21 23:05:25 Sending PUBLISH (d0, q2, r0, m3),
'b'BetaFactory/machine1/pressure'', (102 bytes)
nov 21 23:05:25 Publishing: pressure=3.08
(QoS=2, id=f2ba54ab-9861-4588-a492-758e096a6199) in
BetaFactory/machine1/pressure
nov 21 23:05:28 RTT pressure=3.08
(id=f2ba54ab-9861-4588-a492-758e096a6199)=>2264.0ms

nov 21 23:05:30 Sending PUBLISH (d0, q2, r0, m4),
'b'BetaFactory/machine1/torque'', (101 bytes)
nov 21 23:05:30 Publishing: torque=88.5
(QoS=2, id=e02e3ede-6a2b-4e2a-8a9d-5144de2e7005) in
BetaFactory/machine1/torque
nov 21 23:05:33 RTT torque=88.5
(id=e02e3ede-6a2b-4e2a-8a9d-5144de2e7005)=>2222.5ms

nov 21 23:05:36 Sending PUBLISH (d0, q2, r0, m5),
'b'BetaFactory/machine1/humidity'', (102 bytes)
nov 21 23:05:36 Publishing: humidity=35.0
(QoS=2, id=b7c53e0b-dce0-464b-b600-348d84b4d4ea) in
BetaFactory/machine1/humidity
nov 21 23:05:38 RTT humidity=35.0
(id=b7c53e0b-dce0-464b-b600-348d84b4d4ea)=>2202.5ms
```

On the subscriber side, the payload is received, its structure validated, metadata recorded, the content forwarded to IPFS, and an acknowledgment is returned with the corresponding identifier, ensuring end-to-end traceability.

```
nov 21 23:05:17 Started moti-subscriber.service -
MoTI MQTT→IPFS Subscriber Service.

nov 21 23:05:18 CONNECTED and registered on
BetaFactory/machine1/temperature
nov 21 23:05:18 CONNECTED and registered on
BetaFactory/machine1/pressure
nov 21 23:05:18 CONNECTED and registered on
BetaFactory/machine1/torque
nov 21 23:05:18 CONNECTED and registered on
BetaFactory/machine1/humidity

nov 21 23:05:20 Received temperatura=31.75 from
BetaFactory/machine1/temperature
(QoS=2, id=20a8040e-4fb9-4379-bcf4-15bd6a775dd2)
nov 21 23:05:21 Sent to IPFS:
CID=QmSy8FFF4LuN2L2XDP3kcKKoBYYNcnGmqM4a7wtXERuDZP
nov 21 23:05:21 Notified moti/ipfs/notify with
CID=QmSy8FFF4LuN2L2XDP3kcKKoBYYNcnGmqM4a7wtXERuDZP
nov 21 23:05:23 Remote append OK
nov 21 23:05:23 https://ipfs.io/ipfs/QmSy8FFF4LuN2L2
XDP3kcKKoBYYNcnGmqM4a7wtXERuDZP
nov 21 23:05:23 ACK sent (temperature) to
id=20a8040e-4fb9-4379-bcf4-15bd6a775dd2
```

As shown in the log, the acknowledgment is issued only

after IPFS persistence and the remote metadata append operation are successfully completed. Consequently, the measured RTT captures the full end-to-end processing pipeline rather than only the MQTT transmission time.

After being persisted in IPFS, the processed data is automatically appended to a CSV log file, which can be exported as an Excel spreadsheet by authenticated users via mutual TLS. This log maintains structured metadata associated with each transaction, including timestamp, topic, payload value, QoS level, and the corresponding CID, enabling traceability and auditability of stored information. Table II presents an example of the records retrieved by an authorized user.

TABLE II
EXAMPLE OF DATA EXTRACTED FROM IPFS

m	Time	Topic	Value	QoS	CID
1	23:05:23	temperature	31.75	2	QmSy8FFF4LuN2L2XDP3kcKKoBYYNcnGmqM4a7wtXERuDZP
2	23:05:28	pressure	3.08	2	QmUB7yHEtEMXBVXdjn5AnS2mefXUVDF4wBjksVxcjHsMqo
3	23:05:33	torque	88.50	2	QmSsvqAbzH5FibMippTrNtSGDwA8easRvCfZKWFrt8TMky
4	23:05:38	humidity	35.00	2	QmPxoNEh9qoCUeB1Fczp1QhGQnp4fZnGvsmnG52prwzVdA
5	23:05:53	temperature	21.46	2	QmY4kGfW5JcKmxHiDC2ZqHsZiek6zYZP47EiRJ8qLdYcXn
6	23:05:58	pressure	3.56	2	QmR9jMDXjYKMst2VEqjyKma8B6Urhu7Hpdz7MFL9bMgmX8
7	23:06:03	torque	136.80	2	QmVaGKAJ6EwQ52zewUrLgvvU11JRGdLjiMFjE1qZhupE3T
8	23:06:08	humidity	46.10	2	QmQ2sLzUqam2uisuosDp9px27QuZSnhxxCwrJtNxEcfKww

V. RESULTS

In this section, we present the results of the round trip time (RTT) measurements for messages published by the simulator device until their storage in IPFS. Each transmitted message includes a unique identifier (ID) and a transmission timestamp. At the end of the process, the application returns an acknowledgment with a reception timestamp, which makes it possible to compute the RTT.

To provide a formal representation of the performance metric, the end-to-end message flow in the MoTI architecture can be modeled as follows. For QoS 0 and QoS 1, the flow is:

Publisher → Broker → Interceptor → IPFS → Metadata logging → Acknowledgment.

For QoS 2, an additional integrity step is included:

Publisher → Broker → Interceptor → Authenticator (SHA-256) → IPFS → Metadata logging → Acknowledgment.

Let $t_{publish}$ denote the timestamp recorded at the moment of message publication and t_{ack} the timestamp recorded upon acknowledgment reception. The round-trip time is defined as shown in (1):

$$RTT = t_{ack} - t_{publish} \tag{1}$$

Considering the complete processing pipeline, the RTT includes MQTT transmission, interceptor processing, optional hashing for QoS 2, IPFS persistence, metadata logging in the backend, and the acknowledgment return. Therefore, the RTT can be decomposed as shown in (2):

$$RTT = T_{MQTT} + T_{processing} + T_{hash} + T_{IPFS} + T_{logging} + T_{ack} \tag{2}$$

where T_{MQTT} represents MQTT transmission time, $T_{processing}$ corresponds to interceptor and metadata handling, T_{hash} applies only to QoS 2 (SHA-256 computation), T_{IPFS} accounts for distributed storage latency, $T_{logging}$ represents the backend metadata append operation, and T_{ack} is the acknowledgment return time.

Table III shows a sample of ten data transmissions for each QoS level (0, 1, and 2). The purpose of this table is to illustrate the structure of the collected data, including the RTT, the simulated variable, and the date and time of each transaction. This sample serves as an example of the format and content of the messages used throughout the experiment.

TABLE III
SAMPLE RTT DATA FOR QOS LEVELS 0, 1, AND 2

ID	Time	m	Topic	Value	RTT
QoS 0					
51dc5aec-222b-436e-8688-aff19a0649f0	19:20:11	1	temperature	27.45	2846.5
68980425-1ee6-4e7f-9211-79dc7c7b4266	19:20:16	2	pressure	2.52	2192.8
5c8be53a-1174-4cc8-a133-368205935962	19:20:21	3	torque	207.10	2125.0
5035f576-0ecb-4f44-ae56-5673e13f6bf4	19:20:26	4	humidity	70.50	2096.7
7a17632b-5da1-42c3-8647-657cdec0ec17	19:20:41	5	temperature	31.93	2152.6
c8c4cf07-15f5-4ec7-8581-3fe83e0e331b	19:20:46	6	pressure	4.56	2115.8
2a1a6bcb-c4b0-48be-9e73-95669bf1a85d	19:20:51	7	torque	54.50	2113.9
4ec57747-13a4-4468-8c67-e9efa2e2d9e6	19:20:56	8	humidity	39.50	2154.2
QoS 1					
639dc21e-b1aa-475a-b5de-a7467098df5d	21:59:48	1	temperature	38.16	2449.1
5c78626d-caa1-4189-8e76-eae24bb7a4fd	21:59:53	2	pressure	1.63	2217.4
029fa87f-75e4-4ccd-8d7b-47f94d7c297c	21:59:58	3	torque	98.40	2258.6
b9ad7122-1f39-4d6d-ba49-8b74cc3baa7f	22:00:03	4	humidity	36.10	2169.4
2a1ca329-08d1-440d-acfb-492a4e35f8ef	22:00:18	5	temperature	31.03	2176.0
34942f99-976c-4d69-b6b4-7012d01cc550	22:00:23	6	pressure	3.40	2218.8
0710bc9f-d04f-4558-acdd-e79cde498abd	22:00:28	7	torque	184.30	2188.3
ef1091b9-3ca3-48a2-97af-8d2ed19f6aba	22:00:33	8	humidity	72.20	2222.3
QoS 2					
20a8040e-4fb9-4379-bcf4-15bd6a775dd2	23:05:23	1	temperature	31.75	2525.5
f2ba54ab-9861-4588-a492-758e096a6199	23:05:28	2	pressure	3.08	2264.0
e02e3ede-6a2b-4e2a-8a9d-5144de2e7005	23:05:33	3	torque	88.50	2222.5
b7c53e0b-dce0-464b-b600-348d84b4d4ea	23:05:38	4	humidity	35.00	2202.5
64bbfbd4-55e6-476c-8b85-8fe58f9ba17a	23:05:53	5	temperature	21.46	2199.7
e03995f5-2322-4be0-9f8c-4bbcbb0d589d	23:05:58	6	pressure	3.56	2172.5
e3fba2fc-544f-4893-b322-435b6b588184	23:06:03	7	torque	136.80	2250.1
36070742-60c7-43f4-8c90-2b9ecd0c7709	23:06:08	8	humidity	46.10	2207.3

During implementation, data were collected under different loads and operational conditions in order to evaluate system performance at various moments. Among these datasets, a sample of 500 messages per QoS level (a total of 1500 records) was selected for more detailed analysis, as it represents a more stable operational scenario with a higher data volume.

Fig. 3 shows a line chart with the first 50 samples of each group. This view supports an initial comparison of the behaviors without compromising readability.

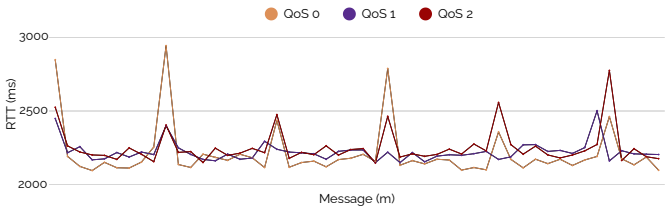

Fig. 3. RTT time series for the first 50 messages per QoS level (0, 1, and 2)

Although the three QoS levels exhibit similar central tendencies in this initial subset, inspection of the complete dataset reveals the presence of extreme outliers, particularly in QoS 0, with several RTT values exceeding 5000 ms. This behavior indicates that, despite its lower protocol overhead, QoS 0 is more susceptible to instability under increased operational load.

Next, individual RTT time series for the 500-message sample of each QoS level are presented. Each figure displays the per-message RTT together with a horizontal reference line indicating the median. The median provides a robust measure of central tendency, minimizing the influence of extreme values in the distribution.

The QoS 0 chart in Fig. 4 shows that this group had the highest variability, with several peaks above 5000 ms and values that exceed 39,000 ms. These outliers strongly influence the mean (2758.5 ms), which moves away from the median (2163.8 ms). This indicates that QoS 0, although theoretically faster because it does not require delivery acknowledgments, is also the most susceptible to delays and instabilities when subjected to higher loads. Such behavior can represent a risk in critical industrial scenarios.

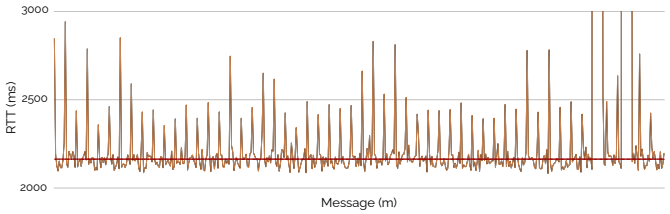

Fig. 4. RTT time series for QoS 0 (500 messages)

The QoS 1 chart in Fig. 5 shows that level 1 presented high stability, with RTTs concentrated in a narrow range and no registered outliers. The mean (2255.9 ms) and the median (2217.6 ms) are very close, and the standard deviation remained low (150.9 ms), resulting in a coefficient of variation (CV) of only 6.69%. This reinforces that the at least once mechanism contributes to more predictable communication, even when the data volume increases.

The QoS 2 chart in Fig. 6 exhibited behavior similar to QoS 1, with stable values and no abrupt variations. The median (2216.0 ms) was practically the same as that of QoS 1, and the mean (2261.3 ms) showed only a slight increase, which is consistent with the additional confirmation steps of the protocol. The standard deviation (192.9 ms) and CV (8.53%) remained low, indicating good predictability without a significant impact on response time.

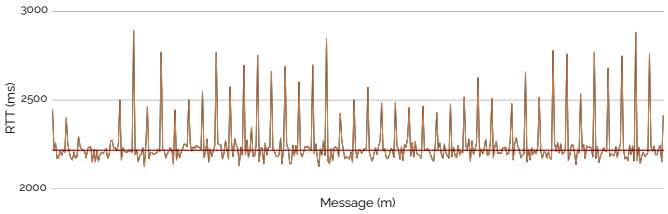

Fig. 5. RTT time series for QoS 1 (500 messages)

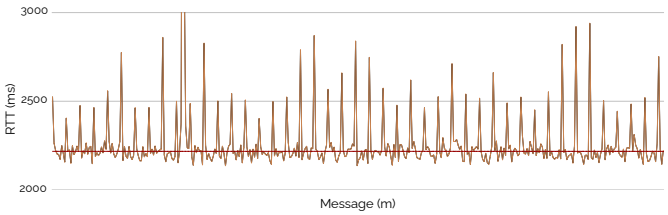

Fig. 6. RTT time series for QoS 2 (500 messages)

Table IV presents a statistical summary of the RTTs obtained for each QoS level, including mean, median, minimum and maximum values, standard deviation, and coefficient of variation (CV). These indicators help to assess the central tendency and stability of the system when operating with higher message complexity. The table also makes it clear that, although the medians remain close across the three groups, the high dispersion of QoS 0 makes it unsuitable for scenarios that are sensitive to delays, while QoS 1 and QoS 2 remain appropriate even under higher load.

TABLE IV
STATISTICAL INDICATORS OF RTT FOR EACH QOS LEVEL

Indicator	QoS 0	QoS 1	QoS 2
Sample size	500	500	500
Mean (ms)	2758	2256	2261
Median (ms)	2164	2218	2216
Minimum (ms)	2082	2125	2137
Maximum (ms)	39764	3344	5110
Standard deviation (ms)	3622	151	193
CV (%)	131.31	6.69	8.53

To provide a more rigorous comparison among QoS levels, statistical hypothesis testing was performed. The results of normality, variance homogeneity, and mean comparison tests are summarized in Table V.

TABLE V
STATISTICAL TEST RESULTS FOR RTT COMPARISON ACROSS QOS LEVELS

Statistical Test	Outcome
Shapiro–Wilk (normality)	$p < 0.05$ (all groups)
Levene (variance equality)	$p < 0.001$
ANOVA	$F = 9.51$, $p < 0.001$
Tukey (QoS0 vs QoS1)	$p < 0.001$
Tukey (QoS0 vs QoS2)	$p < 0.001$
Tukey (QoS1 vs QoS2)	$p = 0.999$

The statistical analysis confirms significant differences across QoS levels. Post-hoc results indicate that QoS 0 differs significantly from QoS 1 and QoS 2, whereas no statistically significant difference is observed between QoS 1 and QoS

2. These findings demonstrate that the primary performance distinction is not associated with average RTT values, which remain relatively close across groups, but rather with the variability of the measurements. In particular, the substantially higher dispersion and extreme outliers observed in QoS 0 contribute to its statistical separation from the other levels. In contrast, QoS 1 and QoS 2 exhibit comparable central tendencies and controlled variability, reinforcing their suitability for industrial scenarios that require predictable communication behavior.

Fig. 7 presents the RTT boxplot for the three QoS levels, with outliers omitted, which makes it easier to visualize the central tendency and the typical spread of values in each group. This chart is useful to evaluate the normal operating behavior of the system, while the full time series charts complement the discussion on stability under load.

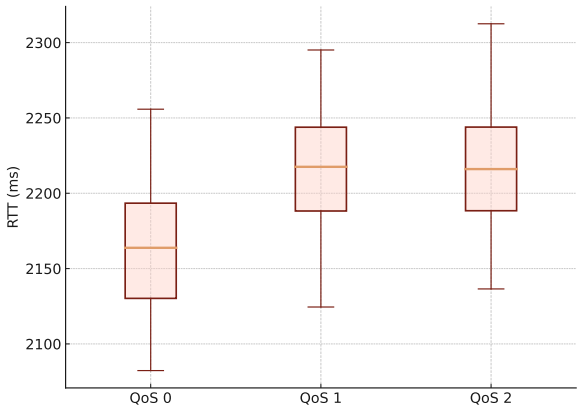

Fig. 7. RTT distribution by QoS level (outliers omitted for visualization clarity)

The experiments showed that the MoTI solution maintains consistent performance under the evaluated conditions. The presence of extreme outliers in QoS 0, which are absent in QoS 1 and QoS 2, reinforces the importance of selecting higher QoS levels in industrial applications that require reliability and operational stability. In this context, QoS 2 emerges as the most suitable option for scenarios in which guaranteed delivery is essential, offering predictable behavior even as the data load increases.

Although the experimental evaluation was conducted under controlled network conditions, it is important to consider that the use of IPFS in geographically distributed or bandwidth-constrained environments may introduce additional latency components. In scenarios involving high propagation delay, intermittent connectivity, or larger payload sizes, content retrieval time may increase due to network traversal and content discovery mechanisms inherent to distributed storage systems.

Further details regarding the experimental setup, extended datasets, and additional analyses are available in the corresponding master's dissertation [14].

Future scalability improvements may include distributed pinning strategies to ensure local availability of frequently accessed content, data compression techniques to reduce payload size, caching mechanisms at edge nodes, and orchestration of multiple IPFS nodes to improve redundancy and access performance. These strategies can mitigate latency fluctuations and enhance robustness in large-scale or resource-constrained industrial deployments.

VI. Scenarios

This section presents illustrative scenarios showing how the MoTI solution can be applied in industrial contexts. The objective is to demonstrate, in a practical manner, its potential to support interoperability, security, and traceability in data exchange among heterogeneous IoT devices and networks.

The scenarios do not represent real deployments, but were designed based on common Industry 4.0 situations involving IoT data collection, secure communication requirements, and distributed storage. In this context, Brazilian initiatives such as the IAsmin Platform foster closer interaction between applied research and industry by addressing topics including interoperability, real-time monitoring, and cybersecurity [13].

A. Scenario A: Manufacturer and End User

This scenario describes the interaction between a manufacturer (Factory Alpha) and an end user (Factory Gamma) during the lifecycle of industrial equipment. Factory Alpha performs Factory Acceptance Tests (FAT) to verify compliance with technical requirements, while Factory Gamma conducts Site Acceptance Tests (SAT) after installation, including inspections, electrical tests, and integration with plant systems such as SCADA or EMS.

Both companies are partners of the IAsmin Platform. Factory Alpha supplies transformers to Factory Gamma, whose plants operate their own substations and present high electrical demand. During FAT and SAT activities, large volumes of reports, logs, and test records are generated, typically stored across disparate systems and shared through email or file repositories, which leads to version ambiguity, missing metadata, and limited traceability. Fig. 8 illustrates the commissioning workflow using the MoTI solution, from FAT execution to data access at Factory Gamma.

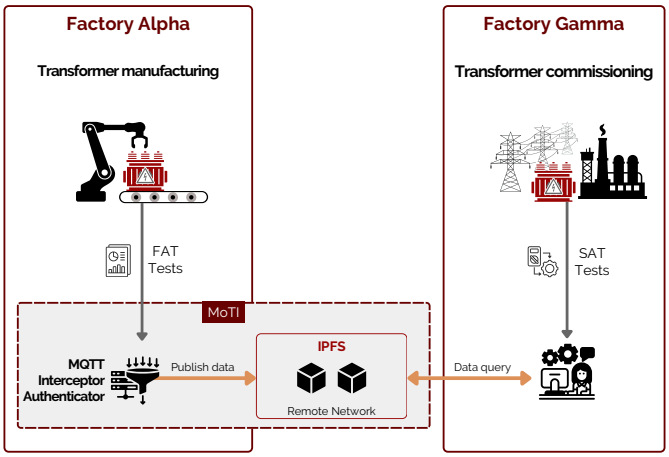

Fig. 8. MoTI application illustrating secure and traceable data exchange in an industrial environment in scenario A

With MoTI, measurements are published as JSON messages containing test data, timestamps, and standardized metadata. The Interceptor applies Quality of Service (QoS) rules, routing critical items with QoS 2 through additional security validation, while non-critical data use lower QoS levels. All content is stored in IPFS, generating a CID that is registered together with its hash and metadata, ensuring integrity and change detection.

At the end of FAT and SAT activities, MoTI automatically generates evidence indexes listing all related CIDs. Factory Gamma can directly retrieve verified documents from IPFS using these identifiers, eliminating manual file consolidation, email exchanges, and version conflicts.

The benefits appear across several aspects. Traceability improves because each report or evidence has a CID that guarantees file integrity, eliminating version confusion and facilitating audits. Organization and speed increase because the FAT and SAT dossiers become living evidence indexes instead of manually assembled collections of documents. Interoperability becomes more secure with certificate-based authenticated publishing and clear permissions for writing and reading, allowing Factory Alpha and Gamma to work from the same information base, each with its defined role. Reliability improves as well, since the use of QoS 2 for critical items reduces the chance of silent data loss. In practice, the MoTI solution makes commissioning clearer, verifiable, and collaborative, and prepares the foundation for smoother day-to-day operations with faster responses and fewer conflicts.

B. Scenario B: Educational and Research Institutions

Educational and research institutions that have agreements with industries, such as those associated with the IAsmin Platform, can use MoTI to access and analyze industrial data in a standardized and traceable manner. In this scenario, industrial partners publish measurements through MoTI, while academic groups retrieve the data via IPFS using CIDs and associated metadata. Fig. 9 illustrates the data exchange workflow for academic use.

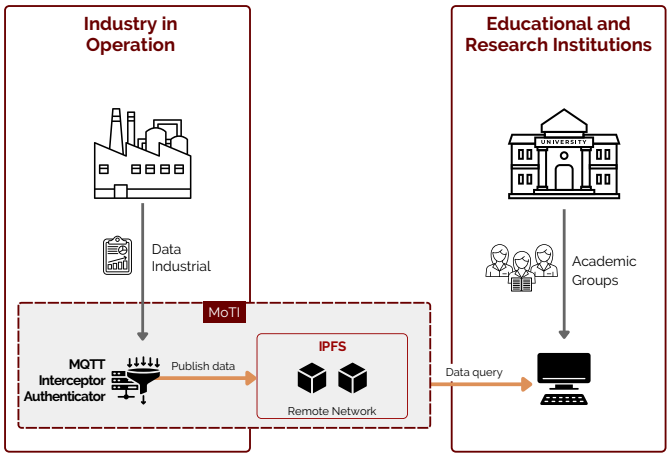

Fig. 9. MoTI application enabling secure and traceable data sharing between industrial and academic environments in scenario B

In practice, the laboratory receives read access to one or more datasets and works with lists of CIDs. Researchers import the list into their analysis environment, download only the required artifacts, and reference the same CIDs in reports or publications. This ensures that any team can reproduce the study by retrieving exactly the same files.

From an interoperability perspective, the benefits are clear. Data are delivered in JSON format with a minimal set of standardized fields, which reduces ambiguity and simplifies ingestion into ETL processes, databases, and analysis notebooks. Content addressing through CIDs keeps references stable over time. Since transport and storage are decoupled, MoTI publishes data with appropriate QoS levels and stores them in IPFS, avoiding reliance on a single server while benefiting from caching and distributed access across teams.

Regarding privacy and confidentiality, MoTI incorporates complementary safeguards. Access is controlled through read profiles that define which CIDs are included in the academic dataset and who is authorized to consult them. Before sharing, the industrial partner may apply data minimization or pseudonymization to sensitive fields, such as replacing serial numbers with neutral identifiers and removing operator names. Each file has its own hash and CID, creating a clear record of when it was generated and under which procedure. This protects the company's compliance requirements while strengthening scientific reproducibility.

The result is an efficient workflow for both sides. Academic institutions receive well-structured, analysis-ready data that are easy to cite. The industrial sector maintains control over what is shared, with verifiable integrity and without the overhead of manually preparing folders and document versions. In practice, MoTI shortens the distance between the factory floor and the research lab and helps transform technical evidence into actionable industrial knowledge.

VII. Conclusions

Throughout this work, the MoTI solution was proposed and implemented with a focus on interoperability, traceability, and distributed access to industrial data. The architecture integrates MQTT publishing with configurable Quality of Service levels, mutual authentication through certificates, subscriber-side processing, and IPFS storage with content-based identification via CIDs. This combination enables the standardization of metadata, the registration of hashes, and the construction of a verifiable end-to-end trail from the publication of telemetry data to the retrieval of the corresponding file. In controlled experiments comprising 1500 total messages across three QoS levels, MoTI demonstrated consistent end-to-end performance under increased message volume. Statistical analysis confirmed significant variability differences across QoS levels, primarily due to the substantially higher dispersion observed in QoS 0. Under the tested conditions, QoS 1 and QoS 2 maintained stable and predictable behavior, with low dispersion and minimal latency overhead. In contrast, QoS 0 exhibited high variability, with substantially higher dispersion and extreme RTT outliers. Although QoS 0 may appear theoretically faster due to the absence of acknowledgment mechanisms, its susceptibility to instability makes it unsuitable for delay-sensitive or safety-critical industrial applications.

Notably, QoS 2 provided enhanced delivery guarantees and integrity verification without introducing a significant increase in average RTT when compared to QoS 1.

In practical terms, the benefits are clear. Authenticated publishing and standardized records reduce ambiguity and manual rework. IPFS storage enables controlled sharing outside the local network while preserving content integrity. The combination of QoS, TLS, and hash verification provides an auditable trail useful for commissioning, operation, and support. The modular design facilitates evolution, allowing new topics, variables, and rules to be added without restructuring the system, and enabling new actors to participate with clearly defined access profiles.

Future developments can advance along three main directions. The first is the selective integration of blockchain to anchor CIDs and hashes of critical events, apply smart contracts for versioning FAT and SAT documents, and issue verifiable conformity attestations, preferably in permissioned networks when consortium governance is required. The second involves conducting geographically distributed tests, with publishers and subscribers located in different regions, to evaluate RTT, delay fluctuations, resilience to propagation latency, and behavior in scenarios involving packet loss and reconnection. The third direction concerns scalability strategies in IPFS, including distributed pinning policies, orchestration of multiple nodes, and evaluation of the impact of compression and variable payload sizes.

In summary, the MoTI solution demonstrated technical feasibility supported by experimental validation and statistical analysis, reinforcing its applicability in distributed industrial environments that require interoperability, traceability, and predictable communication behavior. By integrating MQTT, TLS, IPFS, and structured metadata within a unified and operational architecture, MoTI provides a practical and scalable approach for secure cross-organizational data exchange. Continued development, including selective blockchain anchoring and broader scalability assessments, may further consolidate MoTI as a foundation for auditable, collaborative, and performance-aware industrial IoT applications.

ACKNOWLEDGMENTS

This research was partially funded by FAPESP, grant numbers 2023/00783-7 and 2020/09850-0.

REFERENCES

[1] UNCTAD, "Technology and Innovation Report 2025," *United Nations Conference on Trade and Development*, Geneva, Switzerland, 2025. [Online]. Available: https://unctad.org/publication/technology-and-innovation-report-2025. Accessed: Nov. 2025.

[2] L. R. Moura, "A caminho da Indústria 4.0 – Fundamentos e orientações para a transformação digital na Indústria," São Paulo, Brazil: Brazil Publishing, 2020, doi: 10.31012/978-65-5861-336-7.

[3] S. Li, M. Iqbal, and N. Saxena, "Future Industry Internet of Things with Zero-trust Security," *Information Systems Frontiers*, vol. 26, pp. 1653–1666, 2024, doi: 10.1007/s10796-021-10199-5.

[4] C. C. Barros, A. F. Conceição, and V. Rocha, "Integração de Serviços MQTT Usando Blockchain," in *Proc.* 2023 15th IEEE International Conference on Industry Applications (INDUSCON), pp. 605-606, Nov. 2023, doi: 10.1109/induscon58041.2023.10374720.

[5] P. M. Jacob, P. Mani, S. Simon, R. R. Varghese, K. John, and N. Aniyan, "An Integrated Blockchain Approach to Model Secure Internet of Things based Systems," in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI),* Sakheer, Bahrain, pp. 1-5, Oct. 2020, doi: 10.1109/ICDABI51230.2020.9325617.

[6] E. Sivakumar, G. Ganesan, and Ragavi, "Harnessing I4.0 Technologies for Climate Smart Agriculture and Food Security," in *Proceedings of the 5th International Conference on Future Networks and Distributed Systems (ICFNDS),* pp. 504–510, Dec. 2021, doi: 10.1145/3508072.3508175.

[7] M. Falahi, A. Vasilateanu, N. Goga, G. Suciu, M.A. Sachian, R. Florescu, and S.D. Stanciu, "Improving Security and Scalability in Smart Grids using Blockchain Technologies," in *Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES)*, pp. 1-7, Aug. 2022, doi: 10.1145/3538969.3544441.

[8] D. G. Roy, P. Das, D. De, and R. Buyya, "QoS-aware secure transaction framework for internet of things using blockchain mechanism," *Journal of Network and Computer Applications*, Elsevier BV, vol. 144, pp. 59–78, Oct. 2019, doi: 10.1016/j.jnca.2019.06.014.

[9] X. L. Liu, W. M. Wang, H. Guo, A. V. Barenji, Z. Li, and G. Q. Huang, "Industrial blockchain based framework for product lifecycle management in industry 4.0," *Robotics and Computer-Integrated Manufacturing,* vol. 63, Jun. 2020, doi: 10.1016/j.rcim.2019.101897.

[10] F. Salzano, L. Marchesi, R. Pareschi, and R. Tonelli, "Integrating blockchain technology within an information ecosystem," *Blockchain: Research and Applications,* vol. 5, no. 4, Dec. 2024, doi: 10.1016/j.bcra.2024.100225.

[11] H. Cui, Z. Chen, Y. Xi, H. Chen, and J. Hao, "IoT Data Management and Lineage Traceability: A Blockchain-based Solution," in *Proc.* 2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops), pp. 239–244, Aug. 2019, doi: 10.1109/iccchinaw.2019.8849969.

[12] A. Taherkordi, and P. Herrmann, "Pervasive Smart Contracts for Blockchains in IoT Systems," in *Proceedings of the 2018 International Conference on Blockchain Technology and Application (ICBTA),* pp. 6–11, Dec. 2018, doi: 10.1145/3301403.3301405.

[13] IAsmin Platform, "IAsmin: Artificial Intelligence Platform for Industry 4.0," São Paulo, Brazil. [Online]. Available: https://plataformaiasmin.org.br. Accessed: Nov. 2025.

[14] C. C. de Barros, "*MoTI: Uma Arquitetura MQTT para Interoperabilidade entre Dispositivos IoT na Indústria 4.0*," Master's dissertation, Universidade Federal de São Paulo (UNIFESP), São José dos Campos, Brazil, 2026.

Camila C. de Barros holds a postgraduate degree in Industrial Control and Automation Engineering from Instituto Mauá de Tecnologia (São Paulo) and an MBA in Business Management from Fundação Getúlio Vargas (FGV). She has experience in Industrial Automation Engineering and Computer Systems, with a focus on research and development in the areas of Industry 4.0, Internet of Things (IoT), Blockchain, Artificial Intelligence, and Machine Learning.

Arlindo F. da Conceição holds a B.Sc. in Scientific Computing from the University of Taubaté (UNITAU), an M.Sc. from the University of Campinas (UNICAMP), and a Ph.D. in Computer Science from the University of São Paulo (USP). He completed postdoctoral research at Università degli Studi di Milano - Bicocca (Italy) and the University of Oslo (Norway). He is a faculty member in the Graduate Program in Technological Innovation. His research interests include Distributed Systems and Blockchain.

Vladimir E. M. Rocha holds a Ph.D. in Computer Engineering from the Polytechnic School of the University of São Paulo (USP), with a concentration in Software Engineering and Distributed Systems, and M.Sc. in Computer Science from USP. He is currently an Assistant Professor at the Federal University of ABC (UFABC). He has over ten years of experience as a software architect and developer in distributed systems, including work with companies such as Red Hat and Infraero. His research interests include Software Engineering, Distributed Systems, Cloud Computing, Blockchain, and Peer-to-Peer (P2P) architectures.