Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 6): pp. 6476-6488
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.06.632

Comparative Study Of Opc Ua, Mqtt, And Onem2m For Interoperable Industrial Iot Communication

Mrs. M. M. Ghotkar VMV Commerce, JMT Arts and JJP Science College, Wardhaman Nagar, Nagpur,-08

Abstract
This research paper presents a comparative analysis of OPC UA, MQTT, and OneM2M, three prominent communication protocols in the context of Industrial IoT (IIoT). The study evaluates these protocols based on predefined criteria to assess their performance, scalability, security, and adaptability. The findings offer valuable insights into their suitability for achieving interoperable communication in IIoT environments, contributing to the ongoing discourse on IIoT protocol selection and interoperability.

Keywords: IioT, Protocol, OPC UA, MQTTm OneM2M, Communication Protocol, scalability.

1. Introduction

The Industrial Internet of Things (IIoT) has emerged as a transformative force in industrial processes, offering unprecedented opportunities for real-time data exchange, automation, and enhanced operational efficiency. As industries embrace IIoT technologies, a critical challenge arises: how to ensure seamless communication and interoperability among the myriad of devices and systems operating in heterogeneous environments. This challenge underscores the importance of selecting and implementing communication protocols that can facilitate effective data exchange and collaboration in IIoT ecosystems. [1]

At the heart of this challenge lies the choice of communication protocols, each designed with unique features and capabilities. In this context, this research paper embarks on a comprehensive comparative study of three key communication protocols that have gained prominence in the realm of IIoT: OPC UA (Open Platform Communications Unified Architecture), MQTT (Message Queuing Telemetry Transport), and OneM2M (oneM2M Partnership Project). [2]

The selection of these protocols is a reflection of their widespread adoption and their potential to address the intricate communication requirements of IIoT. OPC UA, known for its robustness and data modeling capabilities, has found favor in industries where data integrity and security are paramount. MQTT, with its lightweight and publish-subscribe architecture, excels in resource-constrained environments, while OneM2M offers a standardized approach to IoT communication, promising interoperability on a global scale.

The fundamental question that drives this research is: which of these protocols is best suited for achieving interoperable IIoT communication? To answer this question, we employ a systematic methodology that involves an in-depth evaluation of OPC UA, MQTT, and OneM2M against predefined criteria and metrics. These criteria encompass performance, scalability, security, and adaptability, all critical aspects in the context of IIoT applications.
By delving into the comparative analysis of these protocols, we aim to provide valuable insights for decision-makers, engineers, and researchers navigating the complex landscape of IIoT protocol selection. The outcomes of this study not only shed light on the strengths and weaknesses of each protocol but also contribute to the ongoing discourse on achieving seamless interoperability in the Industrial Internet of Things.
In the subsequent sections of this paper, we delve into a detailed examination of OPC UA, MQTT, and OneM2M, understanding of protocols, comparative analysis and conclude with implications and avenues for further research. As IIoT continues to reshape industries, the choice of communication protocols takes on ever-increasing significance, making this study a timely and pertinent contribution to the field.

2. Understanding Protocols

2.1 OPC UA: OPC UA is an open, platform-independent, and secure communication protocol and framework designed for industrial applications. It enables seamless data exchange and interoperability between various devices, systems, and software applications in industrial automation, manufacturing, and other industrial domains. [3]

2.1.1 Key Characteristics and Features of OPC UA:

OPC UA is designed to work on different operating systems, hardware platforms, and communication technologies, allowing it to operate in diverse industrial environments. [4] OPC UA follows a service-oriented architecture (SOA), where interactions are achieved through a set of well-defined services. These services include data access, method calls, event notifications, and more. [3] OPC UA provides a robust information modeling framework. It allows for the representation of complex data structures, hierarchies, and relationships, making it suitable for modeling diverse industrial data.[5] Security is a fundamental aspect of OPC UA. It offers features like encryption, user authentication, access control, and data integrity mechanisms to ensure the confidentiality and integrity of data exchanged in industrial systems. [6] OPC UA is scalable to accommodate various deployment sizes, from small-scale setups to large industrial networks with thousands of devices. It supports a wide range of data types used in industrial automation, including numeric, string, arrays, structures, and more, making it suitable for handling diverse data.[7] OPC UA provides mechanisms for discovering and identifying devices, endpoints, and available services on the network. This aids in dynamic configuration and system setup.[8] OPC UA includes features for accessing historical data, allowing users to retrieve and analyze past data points, which is essential for industrial monitoring and analysis.[9]

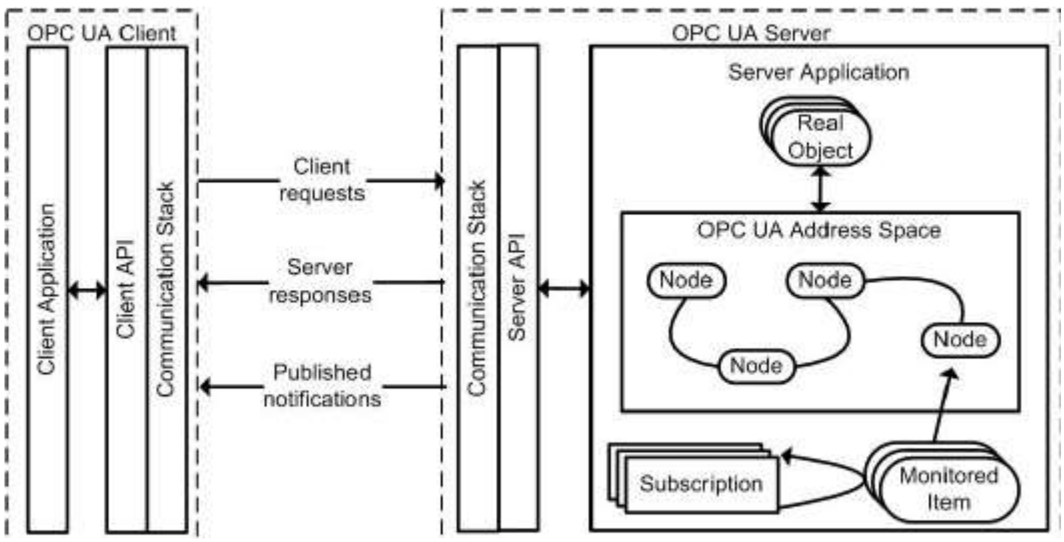
2.1.2 Components of OPC UA:

Server: The OPC UA server is responsible for exposing data and services to clients. It represents a device, system, or application in the OPC UA architecture.
Client: The client initiates communication with the server to request data, invoke methods, and receive notifications. It can be a human-machine interface (HMI), software application, or another industrial system. [3]

Information Model: OPC UA defines an information model that represents the structure and semantics of data and services. This model allows for the standardized description of industrial assets, such as sensors, actuators, and processes.

Endpoints: OPC UA communication occurs between endpoints, which are network-accessible addresses where servers provide services. These endpoints can use different communication protocols, including HTTPS, MQTT, and more.

Security Policies: OPC UA supports various security policies, allowing organizations to choose the level of security that matches their requirements. These policies include Basic256, Basic256Sha256, and others.



OPC UA Client -Server Architecture [10]

2.1.3 Use Cases for OPC UA:

Industrial Automation: OPC UA is widely used in industrial automation and control systems to enable secure and standardized data exchange between programmable logic controllers (PLCs), human-machine interfaces (HMIs), and supervisory control and data acquisition (SCADA) systems.

Manufacturing: OPC UA plays a significant role in optimizing manufacturing processes by providing real-time data visibility and control over production equipment and systems.

Energy Management: It is employed in energy management systems to monitor and control power distribution, grid management, and renewable energy sources.[11]

Smart Buildings: OPC UA is used in smart building systems to integrate various building automation components such as HVAC, lighting, and security systems.[12]

Healthcare: In healthcare, OPC UA ensures interoperability between medical devices and information systems, enabling efficient data exchange and monitoring.

2.2 MQTT

MQTT is a lightweight, publish-subscribe messaging protocol designed for efficient communication between devices and applications in environments with constrained bandwidth, high latency, or unreliable network connections. It follows a client-server model [13] and is particularly well-suited for the Internet of Things (IoT) and Machine-to-Machine (M2M) communication. [14]

2.2.1 Key Characteristics and Features of MQTT:

MQTT operates on a publish-subscribe model, where devices (publishers) send messages to specific topics, and other devices (subscribers) receive messages from those topics. This decouples senders and receivers, allowing for flexible communication.[15] MQTT supports three QoS levels (0, 1, and 2) that define the message delivery guarantees. These levels range from no guaranteed delivery (QoS 0) to guaranteed delivery and acknowledgment (QoS 2), allowing users to choose the level of reliability needed for their application. MQTT is designed to be minimal and efficient, making it suitable for resource-constrained devices with limited processing power and memory. The protocol has a small message header, reducing the overhead associated with data transmission. MQTT uses a binary format that minimizes bandwidth consumption, making it ideal for applications with limited network bandwidth.

MQTT allows clients to specify a "last will" message that is published to a specified topic in case the client unexpectedly disconnects. This feature is useful for monitoring device health and status. MQTT supports retained messages, which means that the last message published on a topic is saved and delivered to new subscribers. This allows clients to retrieve the most recent information when they connect. Clients can establish persistent sessions with MQTT brokers, ensuring that messages are queued for delivery even when clients are offline or disconnected temporarily.While MQTT itself does not define security features, it can be used in conjunction with other security protocols such as TLS/SSL for encryption and authentication, ensuring secure communication.[16]. MQTT brokers can handle multiple clients simultaneously, making it suitable for scenarios with a large number of devices.

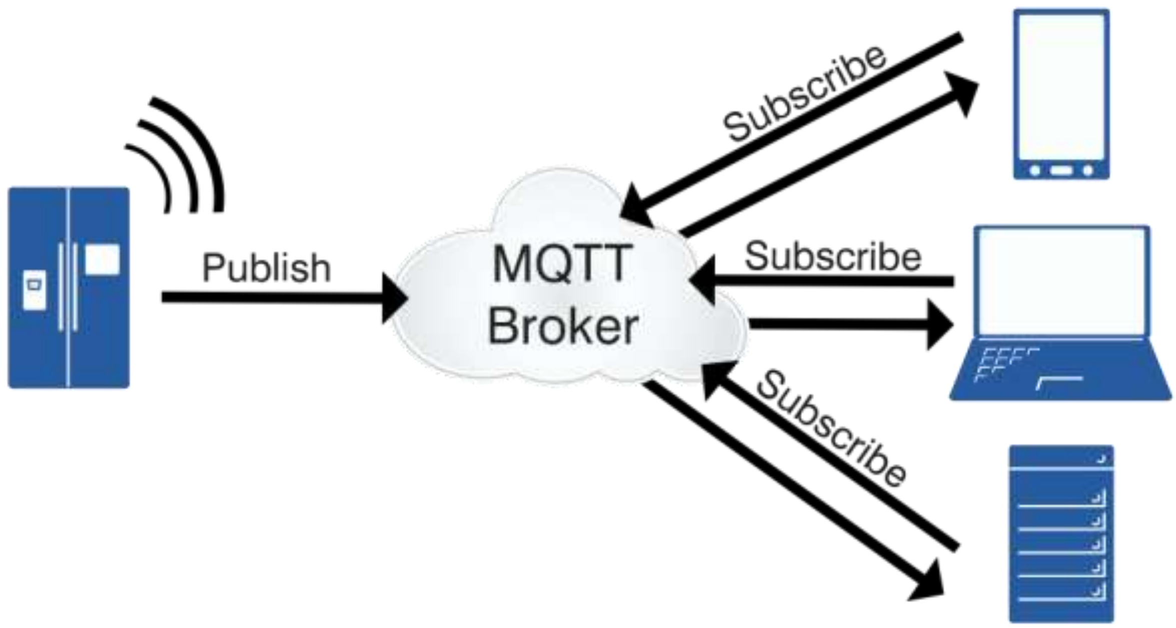
2.2.2 Components of MQTT:

Broker: The MQTT broker acts as an intermediary between publishers and subscribers. It receives published messages and forwards them to relevant subscribers based on topic subscriptions.

Publisher: A publisher is an MQTT client that sends messages to the broker, specifying a topic under which the message is published.

Subscriber: A subscriber is an MQTT client that subscribes to one or more topics to receive messages published under those topics.

Topic: Topics are used to categorize messages. Publishers send messages to specific topics, and subscribers express their interest in particular topics by subscribing to them.



MQTT Architecture (https://www.segger.com/products/security-iot/emmqtt/#mqtt-background)

2.2.3 Use Cases for MQTT:

IoT and M2M Communication: MQTT is widely used in IoT and M2M applications to enable devices and sensors to communicate with centralized systems or with each other. It is ideal for scenarios where low bandwidth, reliability, and scalability are critical. [15]
Remote Monitoring and Control: MQTT is employed in applications like remote monitoring of industrial equipment, agricultural sensors, and home automation systems. It allows for real-time data exchange and control commands.

Telemetry and Data Logging: MQTT is used for collecting telemetry data from remote devices, such as weather stations or vehicle tracking systems. Data can be logged and analyzed for various purposes.

Smart Grids and Energy Management: In the energy sector, MQTT is used to monitor and control smart grid devices, allowing utilities to optimize power distribution and respond to demand.

Asset Tracking: MQTT is employed in asset tracking solutions, enabling real-time location data to be communicated and analyzed.

2.3 OneM2M

OneM2M, short for "one Machine to Machine," is a global standards initiative that provides a common framework and specifications for Machine-to-Machine (M2M) and Internet of Things (IoT) communication. It aims to ensure interoperability and seamless communication between various IoT devices, platforms, and applications, regardless of their manufacturers or technologies. [17]

Key Characteristics and Features of OneM2M:

OneM2M is an international standards organization that defines a comprehensive set of specifications for IoT and M2M communication. These standards help ensure global interoperability and compatibility. OneM2M employs a hierarchical architecture with different functional layers. This architecture allows for modular and scalable deployments, making it suitable for various IoT scenarios. OneM2M defines a common service layer that includes essential functions such as data management, security, access control, and device management. This common layer ensures uniformity and consistency across different implementations.[18] OneM2M's primary focus is on interoperability. It enables different devices and applications to communicate seamlessly by defining standardized interfaces, protocols, and data models. Security is a fundamental aspect of OneM2M. It includes features like authentication, encryption, and access control to protect data and ensure the privacy and integrity of information exchanged in IoT systems. OneM2M defines a standardized approach to data modeling, allowing for consistent representation and interpretation of data across different devices and applications. This is achieved using resources and resource types.

OneM2M's architecture supports scalability to accommodate a wide range of IoT deployments, from small-scale installations to large-scale, city-wide implementations. OneM2M provides application enablement through standardized APIs and interfaces, making it easier for developers to create IoT applications that can access and control devices and data.

2.3.2 Components of OneM2M:

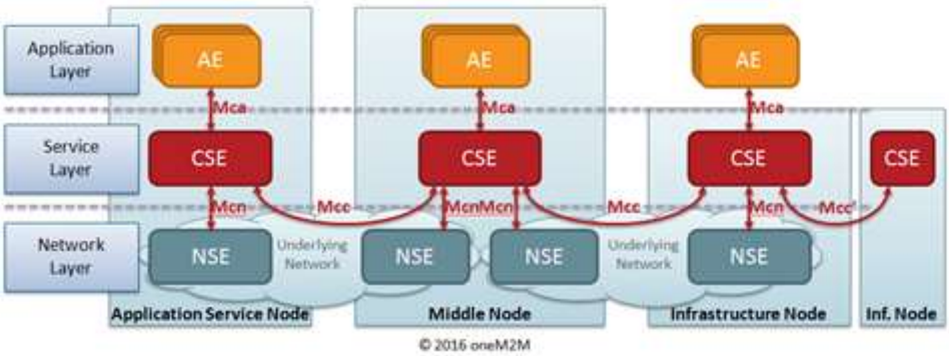
Common Services Entity (CSE): The CSE is a core component in the OneM2M architecture. It provides essential services such as data storage, security, access control, and device management. Multiple CSEs can exist in a hierarchical structure.

Application Entity (AE): An AE represents an application or service that interacts with the CSE. It can retrieve, update, and manipulate data stored within the CSE. AEs can be deployed in various IoT devices and systems.

Resource: Resources are fundamental data elements in OneM2M. They represent entities such as sensors, actuators, devices, or pieces of information. Resources are organized in a hierarchical structure and can be accessed and manipulated by AEs.

Container: Containers are collections of resources that provide logical grouping and organization of data. They simplify resource management and enable efficient data access.

Subscription: Subscriptions allow AEs to receive notifications when specific events or changes occur in the IoT system. Subscribers can be notified about updates to resources or specific conditions being met.



One M2M Functional Architecture [18]

2.3.3 Use Cases for OneM2M:

Smart Cities: OneM2M is used in smart city projects to integrate and manage various IoT devices and services, including smart traffic management, waste management, and environmental monitoring.

Industrial IoT: It is employed in industrial IoT applications to enable interoperability and data exchange between machines, sensors, and control systems, leading to improved automation and efficiency.

Agriculture: OneM2M is used in precision agriculture for remote monitoring of crops, soil conditions, and equipment. It helps farmers make data-driven decisions for resource optimization.

Healthcare: In healthcare, OneM2M facilitates the integration of medical devices, patient monitoring systems, and healthcare data platforms, improving patient care and safety.

Energy Management: OneM2M is applied in energy management and smart grid systems to monitor and control power distribution, optimize energy consumption, and integrate renewable energy sources.

OneM2M's standardized approach to IoT and M2M communication has contributed to its adoption in various industries and use cases. It addresses the challenges of interoperability, scalability, and security, making it a valuable framework for building robust and interconnected IoT ecosystems.

3. Comparison of protocol on the basis of Performance, Scalability, Security, Adaptability

Criterion / Protocol	**OPC UA**	**MQTT**	**OneM2M**
Performance	Low latency	Low to moderate latency	Low latency
	High throughput	High throughput	High throughput
	Low message overhead	Low message overhead	Low message overhead
Scalability	Scalable, suitable for large-scale industrial systems	Scalable, suitable for many devices	Scalable, designed for large-scale IoT Ecosystem
		Supports moderate device counts	
		May require efficient broker setup	
Security	Strong authentication and encryption	Basic authentication and encryption	Strong authentication and encryption
	Comprehensive data integrity mechanism	Limited data integrity features	Robust data integrity mechanisms
		Vulnerable to security breaches	
Adaptability	Supports complex data modeling and diverse data types	Flexible for handling various data formats	Versatile, accommodating diverse data types and formats

	Provides rich information	Limited data modeling capabilities	Offers hierarchical architecture

4. Role of Interoperability in IIoT

Achieving interoperability in Industrial Internet of Things (IIoT) is of paramount importance [19] due to several compelling reasons like

1.Integration of Heterogeneous Devices: In industrial settings, a wide range of devices, sensors, machines, and systems from different manufacturers and generations coexist. Interoperability ensures that these heterogeneous devices can communicate and collaborate seamlessly. Without interoperability, data silos and communication barriers can hinder productivity and innovation. [20]

2.Optimized Operations: Interoperable IIoT systems enable efficient and automated processes. Devices and systems can share data in real time, allowing for better decision-making, resource allocation, and optimization of operations. This leads to increased efficiency, reduced downtime, and cost savings.

3. Scalability: As industrial environments expand, the need for scalability becomes crucial. Interoperable systems can scale easily by adding new devices or components without extensive reconfiguration. This flexibility is essential for accommodating growth and evolving business needs.

4. Data-Driven Insights: IIoT generates vast amounts of data. Interoperability allows data from diverse sources to be aggregated and analyzed cohesively. This enables organizations to extract valuable insights, predict maintenance needs, and make data-driven decisions to improve processes and product quality.

5. Cross-Domain Collaboration: In many industries, collaboration between different domains or stakeholders is essential. Interoperability enables data sharing between organizations, supply chain partners, or regulatory bodies, fostering collaboration and transparency.

6. Future-Proofing: Interoperable systems are more adaptable to future technologies and standards. They can accommodate new devices or communication protocols without requiring a complete overhaul of existing infrastructure, ensuring longevity and competitiveness.

7.Global Connectivity: As industries expand globally, ensuring that IIoT devices and systems can communicate across geographical boundaries becomes crucial. Interoperability standards enable seamless global connectivity, facilitating international trade and operations.

5. Comparison of OPC UA, MQTT, and OneM2M with reference to the interoperability

OPC UA (Open Platform Communications Unified Architecture):

Interoperability: OPC UA is known for its strong emphasis on interoperability. It provides a standardized, platform-independent communication framework that allows different devices and systems, regardless of the manufacturer, to exchange data seamlessly [21][22][23][24][25]. OPC UA achieves interoperability by defining a common information model and communication protocols.

Information Model: OPC UA's Information Model allows for the standardized representation of data, making it easier for devices to understand and communicate with each other. [26]

Interoperability Profiles: OPC UA defines profiles that specify how devices should interact, ensuring consistency and interoperability across different implementations.
Adoption: OPC UA has been widely adopted in industrial automation and manufacturing, where interoperability is critical for integrating devices from various vendors.

MQTT (Message Queuing Telemetry Transport):

Interoperability: MQTT is designed to be lightweight and efficient, which can make it easier to implement and use across different devices and platforms. However, its interoperability depends largely on how it is implemented and configured.[27]

Publish-Subscribe Model: MQTT's publish-subscribe model allows multiple clients to subscribe to and receive messages from topics, making it suitable for scenarios where interoperability between devices is needed.

Standardization: MQTT is a well-known protocol with many open-source implementations, making it accessible for developers and device manufacturers.[16][28]

Use Cases: MQTT is often used in IoT applications where interoperability is essential, such as smart homes and industrial IoT.

OneM2M (OneM2M Partnership Project):

Interoperability: OneM2M is specifically designed to address interoperability challenges in IoT ecosystems. It provides a standardized approach to IoT communication, facilitating interoperability between diverse devices, applications, and platforms.

Hierarchical Architecture: OneM2M's hierarchical architecture separates common services, applications, and connectivity layers, allowing for interoperability at multiple levels.
Standardized Interfaces: OneM2M defines standardized interfaces and protocols, enabling different devices and systems to communicate seamlessly.

Global Standard: OneM2M is an international standard with the goal of ensuring global interoperability in IoT deployments, making it well-suited for applications that require cross-border compatibility.

Conclusion

While all three protocols offer some level of interoperability, OPC UA is known for its robust and standardized approach to achieving interoperability in industrial environments. MQTT provides interoperability benefits due to its lightweight nature and widespread adoption, particularly in IoT applications. OneM2M is designed from the ground up to address interoperability challenges in IoT ecosystems and is focused on providing a global standard for cross-platform compatibility. The choice of protocol should align with specific interoperability requirements and the nature of Industrial IoT application.

References

1. S. Vitturi, C. Zunino, and T. Sauter, "Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G," Proc. IEEE, vol. 107, no. 6, pp. 944–961, Jun. 2019, doi: 10.1109/JPROC.2019.2913443.
2. B. H. Corak, F. Y. Okay, M. Guzel, S. Murt, and S. Ozdemir, "Comparative Analysis of IoT Communication Protocols," in 2018 International Symposium on Networks, Computers and Communications (ISNCC), Rome: IEEE, Jun. 2018, pp. 1–6. doi: 10.1109/ISNCC.2018.8530963.
3. S. Vitturi, C. Zunino, and T. Sauter, "Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G," Proc. IEEE, vol. 107, no. 6, pp. 944–961, Jun. 2019, doi: 10.1109/JPROC.2019.2913443.
4. B. H. Corak, F. Y. Okay, M. Guzel, S. Murt, and S. Ozdemir, "Comparative Analysis of IoT Communication Protocols," in 2018 International Symposium on Networks, Computers and Communications (ISNCC), Rome: IEEE, Jun. 2018, pp. 1–6. doi: 10.1109/ISNCC.2018.8530963.
5. OPC Foundation, "OPC UA Part 1 - Overview and Concepts Release 1.04 Specification." 2017. [Online]. Available: http://www.opcfoundation.org/UA/Part1/
6. "OPC communications," p. 19.
7. OPC Foundation, "OPC UA Part 5 - Information Model Release 1.04 Specification." [Online]. Available: http://www.opcfoundation.org/UA/Part5/

8. “OPC UA Part 2 - Security Model Release 1.04 Specification.pdf.”
9. OPC Foundation, “OPC UA Part 3 - Address Space Model 1.05.02 Specification.” [Online]. Available: http://www.opcfoundation.org/UA/Part3/
10. “OPC UA Part 12 - Discovery and Global Services Release 1.04 Specification.pdf.”
11. “OPC UA Part 11 - Historical Access Release 1.04 Specification.pdf.”
12. J. Imtiaz and J. Jasperneite, “Common Automation Protocol Architecture and Real-time Interface (CAPRI),” in Kommunikation unter Echtzeitbedingungen, W. A. Halang, Ed., in Informatik aktuell. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 79–88. doi: 10.1007/978-3-642-33707-9_9.
13. Y. He, P. Wu, Y. Wang, F. Tao, and B. K. K. Hon, “An OPC UA based framework for predicting energy consumption of machine tools,” Procedia CIRP, vol. 90, pp. 568–572, 2020, doi: 10.1016/j.procir.2020.02.133.
14. V. Marinakis, “Big Data for Energy Management and Energy-Efficient Buildings,” Energies, vol. 13, no. 7, p. 1555, Mar. 2020, doi: 10.3390/en13071555.
15. R. A. Light, “Mosquitto: server and client implementation of the MQTT protocol,” J. Open Source Softw., vol. 2, no. 13, p. 265, May 2017, doi: 10.21105/joss.00265.
16. “MQTT Service Broker for Enabling the Interoperability of Smart City Systems.”
17. M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, “Internet of Things: Survey and open issues of MQTT protocol,” in 2017 International Conference on Engineering & MIS (ICEMIS), Monastir: IEEE, May 2017, pp. 1–6. doi: 10.1109/ICEMIS.2017.8273112.
18. D. Dinculeană and X. Cheng, “Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices,” Appl. Sci., vol. 9, no. 5, p. 848, Feb. 2019, doi: 10.3390/app9050848.
19. S. K. Datta and C. Bonnet, “A Lightweight Frameworl< for Efficient M2M Device Management in oneM2M Architecture”.
20. “Using oneM2M.pdf.” [Online]. Available: https://onem2m.org/using-onem2m/developers/basics
21. F. Lelli, “Interoperability of the Time of Industry 4.0 and the Internet of Things,” Future Internet, vol. 11, no. 2, p. 36, Feb. 2019, doi: 10.3390/fi11020036.
22. M. Felser, M. Rentschler, and O. Kleineberg, “Coexistence Standardization of Operation Technology and Information Technology,” Proc. IEEE, vol. 107, no. 6, pp. 962–976, Jun. 2019, doi: 10.1109/JPROC.2019.2901314.
23. M. Hoffmann, C. Büscher, T. Meisen, and S. Jeschke, “Continuous Integration of Field Level Production Data into Top-level Information Systems Using the OPC Interface Standard,” Procedia CIRP, vol. 41, pp. 496–501, 2016, doi: 10.1016/j.procir.2015.12.059.
24. S. Cavalieri and S. Mulè, “Interoperability between OPC UA and oneM2M,” J. Internet Serv. Appl., vol. 12, no. 1, p. 13, Dec. 2021, doi: 10.1186/s13174-021-00144-9.
25. S. Cavalieri, M. G. Salafia, and M. S. Scroppo, “Towards interoperability between OPC UA and OCF,” J. Ind. Inf. Integr., vol. 15, pp. 122–137, Sep. 2019, doi: 10.1016/j.jii.2019.01.002.

26. H. Derhamy, J. Ronnholm, J. Delsing, J. Eliasson, and J. van Deventer, “Protocol interoperability of OPC UA in service oriented architectures,” in 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), Emden: IEEE, Jul. 2017, pp. 44–50. doi: 10.1109/INDIN.2017.8104744.
27. S. Kannoth, F. Schnicke, and P. O. Antonino, “Enabling Industry 4.0 Communication Protocol Interoperability: An OPC UA Case Study,” in 7th Conference on the Engineering of Computer Based Systems, Novi Sad Serbia: ACM, May 2021, pp. 1–9. doi: 10.1145/3459960.3459977.
28. F. Pauker, T. Frühwirth, B. Kittl, and W. Kastner, “A Systematic Approach to OPC UA Information Model Design,” Procedia CIRP, vol. 57, pp. 321–326, 2016, doi: 10.1016/j.procir.2016.11.056.
29. R. A. Abdelouahid, M. Oqaidi, and A. Marzak, “Towards to a new IoT Interoperability Architecture,” in 2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD), Marrakech, Morocco: IEEE, Nov. 2018, pp. 148–154. doi: 10.1109/ITMC.2018.8691216.
30. M. Dave, J. Doshi, and H. Arolkar, “MQTT- CoAP Interconnector: IoT Interoperability Solution for Application Layer Protocols,” in 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India: IEEE, Oct. 2020, pp. 122–127. doi: 10.1109/I-SMAC49090.2020.9243377.