Proceedings of the Canadian Society for Mechanical Engineering International Congress
32nd Annual Conference of the Computational Fluid Dynamics Society of Canada
Canadian Society of Rheology Symposium
CSME-CFDSC-CSR 2025
May 25–28, 2025, Montréal, Québec, Canada

A Comparative Analysis of Kafka and MQTT for Real-Time Data Streaming in Cyber-Physical Warehouse System

Vijeet Gahlawat[1*], Ali Aidibe[1*], Nekkunj Pilani[2], Antoine Saucier[1], Soumaya Yacout[1*]
[1]Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montreal, Canada
[2]Department of Computer Science and Operations Research, Université de Montréal, Montreal, Canada
*vijeet.gahlawat@polymtl.ca; ali.aidibe@polymtl.ca; soumaya.yacout@polymtl.ca

Abstract— The increasing need for data streaming solutions in industrial environments, driven by Industry 4.0, necessitates the exploration of advanced technologies capable of moving real-time data generated by connected devices within a Cyber-Physical System (CPS). This paper presents a comparative analysis between Kafka, a distributed data streaming platform, and MQTT, a standards-based messaging protocol, in terms of latency and throughput. The analysis is conducted by experimentation in a CPS testbed within the *Intelligent Cyber-Physical Systems* laboratory at Polytechnique Montreal, Quebec, Canada. The experiments' results demonstrated a higher data throughput and lower latency with Kafka compared to MQTT, and preliminary indications of scalability. This highlight Kafka's potential to facilitate the transition from traditional industrial automation to advanced industrial CPS, thus enabling scalable and efficient data-driven decision-making.

Kafka; MQTT; PLC; real-time data streaming; large language models; industrial automation; cyber-physical system

I. INTRODUCTION

The transition to Industry 4.0 has driven the need for efficient communication systems for real-time monitoring and control of industrial processes. Real-time data acquisition, analysis, and streaming with Programmable Logic Controllers (PLC) are essential for improving operations in smart and connected factories. Real-time data streaming serves as a backbone for enabling real-time analytics and communication within industrial systems. By integrating real-time data with monitoring and analytics, it becomes possible to process information instantaneously, facilitating dynamic decision-making and system interaction [1]. This combination offers multiple benefits, such as running advanced analytics on live data to detect anomalies, predicting equipment failures, or optimizing operations without delay [2]. This integration improves operational safety, system responsiveness, and overall efficiency.

Kafka [3], a distributed data streaming platform, is designed to move, process, and analyze continuous streams of data in real-time across multiple machines or nodes. The authors of [4] integrated Kafka with Ray for parallel processing. Ray is an open-source distributed computing framework that can process and analyze real-time data streams from Kafka. The authors of [5] proposed a Kafka-based Industrial Internet of Things (IIoT) gateway to connect legacy systems with modern Information Technology (IT) infrastructure.

In smart manufacturing, the authors of [6] explored Kafka's role in factory networks for large-scale data collection, while the authors of [7] introduced a multi-layered Kafka-based architecture for predictive maintenance. However, these works focus on Kafka's benefits rather than its comparative performance with other methods or applicability in smart warehousing. Kafka has also been integrated into industrial analytics. In [8], the authors combined Kafka with fuzzy logic to reduce redundant alarms in power plants, and in [9], they explored its role in Digital Twin simulations. The study in [10] conducted a Kafka vs. Message Queuing Telemetry Transport (MQTT) comparison. However, the study lacked the details of the laboratory setup, and the equipment used for conducting the comparison.

To our knowledge, the existing literature does not provide an experimental comparison of Kafka and alternative real-time data streaming and standards-based messaging protocols such as MQTT [11], in the context of a Cyber-Physical System (CPS) testbed.

This paper compares Kafka and MQTT in a CPS testbed of an automated warehouse within the *Intelligent Cyber-Physical Systems* (I-CPS) laboratory at Polytechnique Montreal, Quebec, Canada. The comparison is conducted in terms of latency and throughput. The rest of this paper is organized as follows: Section 2 presents the I-CPS testbed, including the laboratory setup and data flow. Sections 3 and 4 present the experimental setup and the results respectfully. Finally, Section 5 provides the conclusion and outlines future work.

II. I-CPS LABORATORY TESTBED

The I-CPS laboratory (Fig. 1) is a testbed for innovative research in collaboration with industrial partners to develop a smart and connected warehouse. The I-CPS laboratory facilitates the development and validation of solutions related to intelligent decision support via digital twin technology, real-time data-based maintenance for physical asset health management, Human-Machine Interaction (HMI), intelligent robotics, user experience, cybersecurity, and enabling communication network technologies. CPS architecture of the testbed facilitates real-time monitoring, data acquisition, and control of interconnected components. The I-CPS laboratory also features a Decision Theater for real time interactive decision making. It is intended to provide dashboards and real-time monitoring of key performance indicators, system status, and operational data, thereby supporting informed decision-making and efficient warehouse operations.

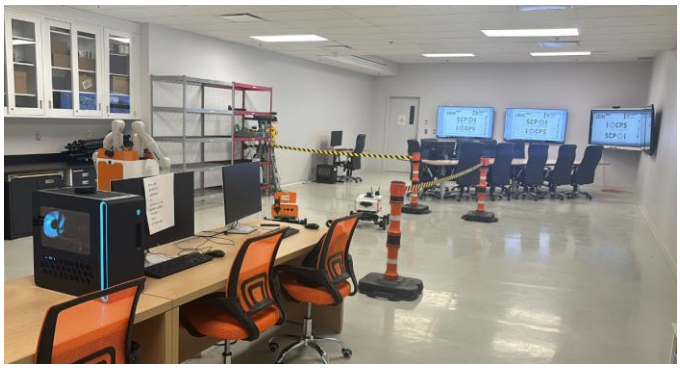
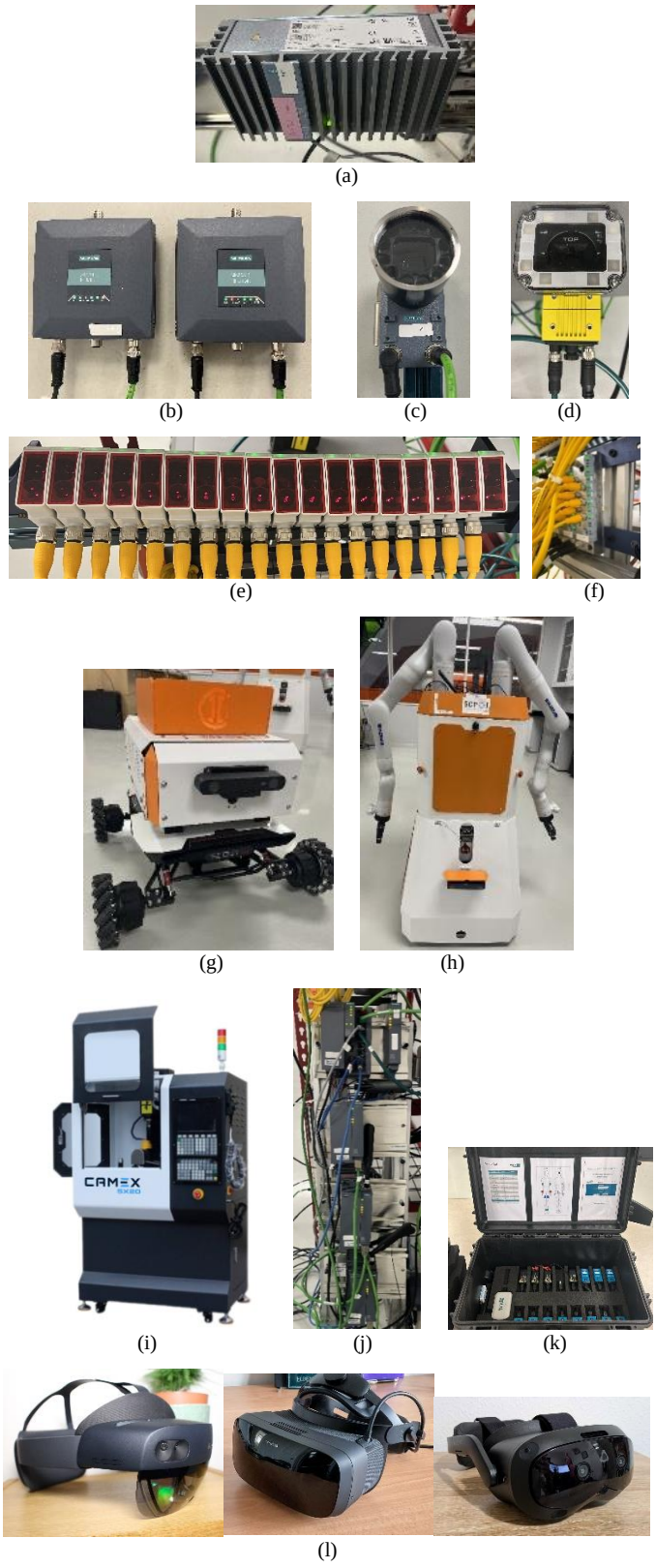


Figure 1. I-CPS Laboratory

At the core of the testbed architecture lies the PLC (Fig. 2), which operates as the central processing unit. The PLC processes receives signals from various sensors and devices, including an Edge PC (Fig. 3a), Radio Frequency Identification (RFID) sensors (Fig. 3b), a Quick-Response (QR) code reader (Fig. 3c), an industrial camera vision system for quality inspection (Fig. 3d), proximity sensors (Fig. 3e), Input/Output (I/O) Link modules (Fig. 3f), three Autonomous Mobile Robots— two transporters AMR (Fig. 3g) and a dual arm manipulator (AMR) (Fig. 3h), a Desktop Computer Numerical Control (CNC) machine (Fig. 3i), and communication devices such as Industrial Ethernet (IE) Switch and Industrial Wireless Local Area Network (IWLAN) Access Points (Fig. 3j), Ergonomic sensors (Fig. 3k), Virtual Reality (VR) and Augmented Reality (AR) devices (Fig. 3l). The PLC generates corresponding output commands for system operations and sends them to Totally Integrated Automation Portal (TIA) software [12].

Figure 2. PLC



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 3. (a) Edge PC; (b) RFID sensors; (c) QR code reader; (d) Camera vision system; (e) Proximity sensors; (f) I/O Link modules (g) Transporter AMR, (h) Dual arm manipulator AMR; (i) Desktop CNC; (j) IE switch and IWLAN Access Points; (k) Ergonomic sensors; (l) VR/AR devices.

Fig. 4 illustrates the I-CPS laboratory setup. Different workstations facilitate: (1) product manufacturing using the Desktop CNC machine; (2) quality inspection via the camera vision system; (3) AMR and product tagging with a Radio-Frequency Identification (RFID) writer; (4) AMR and product tracking with an RFID reader; and (5) product identification via a QR code reader. The testbed also incorporates designated waiting areas, including: (1) unloaded waiting area for transporter AMR; (2) a product pick-up area; (3) receiving storage area for incoming products; (4) shipping storage area for outgoing products; and (5) a quarantine area for non-conforming products.

The warehouse process begins with stocking incoming products manufactured by the Desktop CNC machine, which serves as the supplier. These products are stored in the receiving storage area before progressing through various workstations. When a customer order is received, the transporter and the arm AMRs move the outgoing products to the shipping storage area. During this process, products pass through tracking, inspection and identification workstations to ensure traceability and quality control. Based on inspection results, conforming products proceed to the shipping zone, while non-conforming items are directed to the quarantine zone for further evaluation. They are either reworked or scrapped. All the operations and the decisions are taken by the machines, without any human intervention.

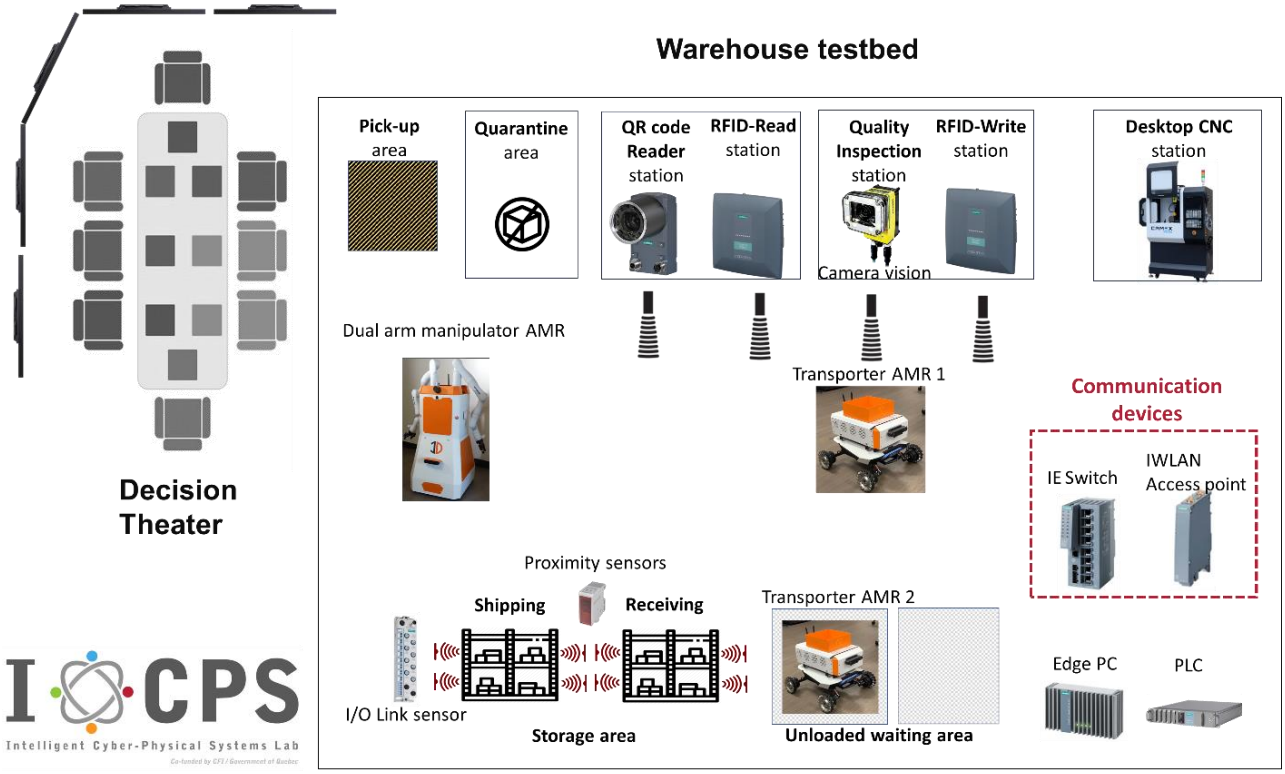


Figure 4. Layout of the I-CPS testbed.

Fig. 5 illustrates the data flow from various sensors and devices to the PLC through the IE Switch, ensuring communication and centralized control within the testbed's CPS. The PLC is an interface for operational control and real-time monitoring via TIA Portal. The IE Switch is an 8-port network switch that directs data traffic. It supports both direct wired connections and, by connecting to Access Points (AP), facilitates wireless communication. This setup enables data exchange between wired and wireless devices on the network. AP1 is directly wired to the IE Switch and acts as a central point, connecting other AP (AP2, AP3, and AP4) to the IE switch and, ultimately, to the PLC for real-time processing. This allows for data exchange between wired and wireless devices on the network.

RFID sensors are connected via a wired data link to AP4. The data from AP4 is transmitted to AP1. Proximity sensors are linked to the I/O Link modules. These modules, connected to AP2, relay the collected data to AP1. The transporters and arm manipulator AMRs communicate indirectly with the PLC. They first transmit data wirelessly to an Edge PC via a Wi-Fi adapter. The Edge PC acts as a communication bridge and sends this data through a wired connection to the IE Switch. The camera vision system is directly wired to the IE Switch. Similarly, the QR code reader is connected via a wired link to AP3 and its corresponding data is then transmitted through AP1 to the IE Switch.

This structured and hierarchical communication setup enables data acquisition, real-time monitoring, and automated decision-making within the testbed. By leveraging wired and wireless connections, the CPS facilitates data transmission across all interconnected components, supporting the testbed's automation requirements. The utilization of TIA Portal software provides an interface for real-time system control, reinforcing the architecture's capability to meet the demands of modern industrial environments.

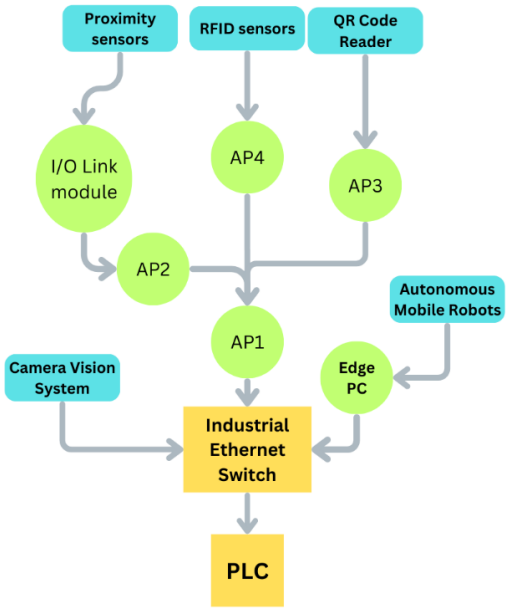


Figure 5. Data flow in the I-CPS testbed.

III. EXPERIMENTAL SETUP

This study evaluates the performance of Kafka and MQTT for industrial real-time data streaming, focusing on latency and throughput, within the testbed of the warehouse CPS presented in section II.

The comparison (Fig. 6) using Kafka and MQTT consisted of the following elements:

- Data Source (PLC): The PLC receive real-time industrial data and machine status. TIA Portal software facilitates data acquisition, and Snap7, a third-party library in Python [13], is used to export this data for external processing. This ensures data transfer for downstream analysis.
- Kafka [3]: It is a distributed platform with brokers, topics, and partitions. Broker is a server or software component that manages the transmission of messages. Data is written on topics and split into partitions for parallel processing. Producers publish messages to topics. Consumers subscribe to topics.
- MQTT [11]: It is a lightweight messaging protocol. It uses a publish/subscribe model with a central client (broker) to handle communication between clients.
- Data Storage and Analytics/Visualization: The streamed data is stored in a database for further analysis or visualization.

The key performance metrics evaluated include:

- Latency: The time interval (in seconds) between the published data from the source (PLC) and its successful delivery to the target system or consumer.
- Throughput: The volume of data successfully processed and transmitted per second, typically measured in frequency of messages sent per second.

We configured Kafka and MQTT to test latency and throughput under the same setup settings. The setup involves deploying Kafka brokers and MQTT brokers on the same network infrastructure, with client applications simulating message production and consumption. The experiments were performed under different workload scenarios, where the workloads represent varying volumes of CPS data that needed to be transmitted through the Kafka and MQTT architectures. Kafka's producers and consumers are configured to measure throughput and latency, while MQTT clients use Quality of Service (QoS) levels 0, 1, and 2 to assess the impact of different delivery guarantees on performance.

In Kafka, producers send messages to a topic, which is partitioned across multiple brokers, and consumers receive the messages, with latency measured from the time the producer sends data to when the consumer receives it. In MQTT, a client publishes messages to the broker, and a subscriber receives them, with latency calculated by comparing the timestamp of message publishing and subscribed. Throughput testing focuses on the volume of data each setup can process within a set timeframe, and it is measured in terms of messages per second. For Kafka, throughput is measured by pushing messages to Kafka topics and determining how many messages the consumer can process per second, while in MQTT, it is evaluated by publishing messages to the broker and observing how much the consumer receives in a given time. We used Prometheus [14] and Grafana [15] for continuous monitoring and visualization of latency and throughput. Prometheus and Grafana are configured with the testbed data streaming architecture requirements to scrape detailed and customized metrics from Kafka and MQTT brokers, as well as from client applications.

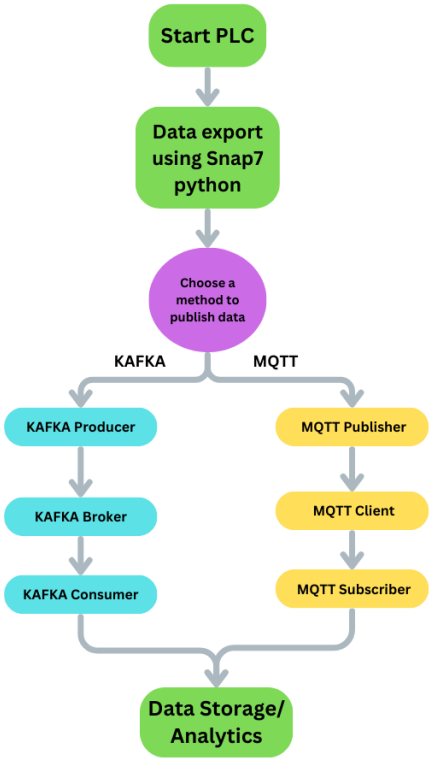


Figure 6. Comparison flow: Kafka vs. MQTT.

To transfer high volumes of data in Kafka, we introduced partitions, which enable data to be sent in parallel, ensuring distribution of the load across multiple consumers. Each Kafka topic is divided into multiple partitions, allowing producers to write data across them simultaneously. Consumers in a consumer group read from different partitions concurrently. It is to note that MQTT itself doesn't have a built-in concept of partitions for parallel processing.

IV. RESULTS

Table 1 presents a performance comparison between MQTT and Kafka (2 and 5 partitions), focusing on latency and throughput. The datasets streamed from the PLC were stored in CSV files (common format for tabular data) using Snap7 (Python). These incorporate discrete values, unique Identifications (ID), AMR status in text format, and integer-based parameters, matching the structure that is configured within the TIA Portal. The CSV file sizes varied based on workload scenarios: the data size of the workload scenario 1 reached 8 MegaBytes (MB), while it was 163 MB for workload scenario 2. Results observed lower latency and higher throughput with Kafka (2 partitions) compared to MQTT for both workload scenarios. Increasing Kafka partitions from 2 to 5 shows an improved latency and throughput, suggesting potential scalability advantages.

TABLE I. KAFKA VS MQTT RESULT METRICS

Data Transfer Technique	**Comparison Metrics**			
	Latency (second)		***Throughput (messages/second)***	
	Workload Scenario 1	***Workload Scenario 2***	***Workload Scenario 1***	***Workload Scenario 2***
MQTT	31	248	367	264
Kafka (2 partitions)	2	190	424	280
Kafka (5 partitions)	2	116	510	302

V. CONCLUSION AND FUTURE WORK

This study provides a preliminary evaluation of two real-time data streaming methods, Kafka and MQTT, in terms of latency and throughput. The results show that for the I-CPS laboratory testbed environment, Kafka with its distributive architecture and partitioning possibility emerged as a promising solution for enabling real-time data streaming and analysis, as it is providing lower latency and higher throughput compared to MQTT, under the same conditions.

Building on the comparative analysis of Kafka and MQTT for industrial data streaming, future research will conduct controlled experiments to further investigate the performance, scalability, and fault tolerance of different real-time data streaming transfer techniques, including Kafka and MQTT, under varying operational conditions. Furthermore, the approach will also integrate Retrieval-Augmented Generation (RAG)-based Large Language Models (LLM) with industrial systems to generate real-time alerts, provide contextual recommendations, and enable real-time analytics by processing streamed data and retrieving relevant historical insights. This approach aims to enhance anomaly detection, predictive maintenance, and operational decision-making, paving the way for a more intelligent and responsive industrial CPS.

ACKNOWLEDGMENT

We gratefully acknowledge the financial support of the Canada Foundation for Innovation (CFI) and the Government of Quebec, for project number 40115

REFERENCES

[1] J. Xu, W. Wan, L. Pan, W. Sun, and Y. Liu, "A Systematic Review of Real-Time Data Monitoring and Its Potential Application to Support Dynamic Life Cycle Inventories," *Environmental Management and Sustainable Development*, vol. 14, no. 1, pp. 1-20, 2025.

[2] F. Ye, Z. Xia, M. Dai, and Z. Zhang, "Real-Time Fault Detection and Process Control Based on Multi-channel Sensor Data Fusion," *arXiv preprint arXiv:2005.12585*, May 2020.

[3] Apache Software Foundation, "Apache Kafka Documentation," Jan.2025. [Online]. Available: https://kafka.apache.org/20/documentation.html. [Accessed: 10-Feb-2025]

[4] Kasumi Kato, Atsuko Takefusa, Hidemoto Nakada, and Masato Oguchi, "A Study of a Scalable Distributed Stream Processing Infrastructure Using Ray and Apache Kafka," IEEE International Conference on Big Data (Big Data), 2018.

[5] Sangil Park and Jun-Ho Huh, "A Study on Big Data Collecting and Utilizing Smart Factory Based Grid Networking Big Data Using Apache Kafka," IEEE Access, vol. 11, pp. 96131–96145, 2023.

[6] Nico Braunisch, Sven Schlesinger, and Robert Lehmann, "Adaptive Industrial IoT Gateway Using Kafka Streaming Platform," 2022 IEEE 20th International Conference on Industrial Informatics (INDIN), pp. 600–607, 2022.

[7] Filippo Bosi et al., "Cloud-enabled Smart Data Collection in Shop Floor Environments for Industry 4.0," European Union Conference on Industry 4.0, 2019.

[8] Bożena Małysiak-Mrozek et al., "Interpreting Industrial IoT Data Streams Through Fuzzy Querying with Hysteretic Fuzzy Sets on Apache Kafka," IEEE Transactions on Fuzzy Systems, vol. 32, no. 8, pp. 4671–4685, 2024.

[9] Gleb Radchenko, Ameer B. A. Alaasam, and Andrey Tchernykh, "Micro-Workflows: Kafka and Kepler Fusion to Support Digital Twins of Industrial Processes," IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), 2018.

[10] C. L. D. Ho, C.-H. Lung, and Z. Mao, "Comparative Analysis of Real-Time Data Processing Architectures: Kafka versus MQTT Broker in IoT," 2024 IEEE 4th International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB), pp. 1–6, 2024. doi: 10.1109/ICEIB61477.2024.10602689.

[11] MQTT, "MQTT: The standard for IoT messaging," MQTT.org, [Online]. Available: https://mqtt.org/. [Accessed: Feb. 14, 2025].

[12] Siemens, "TIA Portal: An overview of the most important documents and links (Controller)," Siemens Industry Support. [Online]. Available: https://support.industry.siemens.com/cs/document/65601780/tia-portal-an-overview-of-the-most-important-documents-and-links-controller?dti=0&lc=en-CA. [Accessed: Feb. 14, 2025].

[13] Python-Snap7, "Python-Snap7 Documentation," Read the Docs, [Online]. Available: https://python-snap7.readthedocs.io/en/latest/. [Accessed: Feb. 14, 2025].

[14] The Prometheus Authors, "Prometheus Documentation: Overview," Jan. 2025.[Online].Available:https://prometheus.io/docs/introduction/overview/. [Accessed: 10-Feb-2025].

[15] Grafana Labs, "Grafana Documentation," Jan. 2025. [Online]. Available: https://grafana.com/docs/. [Accessed: 10-Feb-2025