Resilient IIoT Communication via Edge-Based Store-and-Forward and SLO Monitoring

José António Matias, Raimundo José de Araújo Macêdo
`matiasjose@ufba.br, macedo@ufba.br`

Distributed Systems Laboratory (LaSiD)
DCI/IC – Federal University of Bahia (UFBA)

Abstract. *Industrial Internet of Things (IIoT) systems integrate Operational Technology (OT) and Information Technology (IT) through distributed pipelines, but remain vulnerable to wide-area network (WAN) instability, which affects latency and reliability. This work-in-progress paper proposes an edge-based architecture that combines OPC UA–MQTT translation, a store-and-forward mechanism, and monitoring based on Service Level Objectives (SLO) to improve resilience. Preliminary simulation results show a reduction in latency from 23.21 ms (cloud, low load) to 3.77 ms at the edge, while maintaining system stability even under stress scenarios in which the cloud reached 961.97 ms. An increase in delivery reliability (94.3%) under failures is also observed, highlighting the benefits and trade-offs of edge buffering.*

1. Introduction

The transition to Industry 4.0 is characterized by the integration of cyber-physical systems and the increasing demand for scalable, data-driven industrial environments [Kagermann et al. 2013]. Traditional monolithic architectures, aligned with ISA-95, are increasingly inadequate for handling the growing volume of real-time industrial data [Newman 2015, Bigheti et al. 2019].

To address these challenges, modern systems adopt distributed paradigms based on cloud computing, edge computing, and microservices [Sharma et al. 2024]. Microservices enable modularity and horizontal scalability, while edge computing reduces latency by processing data closer to the source [Castilho and Kamienski 2018, Gaffurini et al. 2025].

However, integrating OT and IT layers introduces strong dependency on WAN connectivity, creating a critical point of failure [Usman et al. 2022, Jammes et al. 2014]. Network disruptions may result in data loss, latency degradation, and reduced system reliability [Hewa et al. 2022]. Furthermore, existing proposals often lack validation under degraded network conditions and rarely adopt standardized performance metrics or SLO-based governance [Bjørndal et al. 2021, Gaffurini et al. 2025]

This work proposes a resilient IIoT communication architecture that integrates edge computing, microservices, and protocol translation between OPC UA and MQTT [Hoppe et al. 2017]. It designs and evaluates a store-and-forward buffering mechanism to mitigate WAN instability and reduce data loss, and provides a quantitative analysis of latency and delivery reliability under network disruptions. The study also identifies trade-offs between buffer capacity, data generation rate, and system autonomy, and introduces an SLO-aware monitoring approach as a basis for future adaptive control mechanisms.

2. Background and Motivation

We have conducted an exploratory literature review across IEEE Xplore, Scopus, ScienceDirect, and Google Scholar, identifying 1,401 studies (2011–2025). After screening using defined inclusion criteria, 25 studies were selected for detailed analysis. The analysis reveals a strong trend toward microservices and edge computing: 64% of architectures adopt microservices, 56% incorporate edge processing, and 88% emphasize horizontal scalability [Sharma et al. 2024, Bigheti et al. 2019, Kompally 2025]. These approaches aim to reduce latency, improve modularity, and enable flexible deployment.

Despite these advances, important limitations remain. Approximately 72% of studies rely exclusively on simulation-based validation, often neglecting real-world network variability [Bjørndal et al. 2021]. In addition, 96% do not use standardized metrics, such as P95 latency or explicit SLO definitions, which limits reproducibility and comparability.

Interoperability also remains a challenge. Around 44% of studies report protocol fragmentation issues, and only a minority adequately address semantic integration across heterogeneous industrial systems [Al-Masri et al. 2020, Kyösti and Lindström 2022]. These gaps highlight the need for architectures that combine resilience, interoperability, and measurable performance guarantees.

3. Hybrid Resilient Architecture

The proposed architecture is structured into three layers: OT, Edge, and Cloud (Fig. 1). The design focuses on improving communication resilience through edge-based processing, protocol translation, and local buffering.

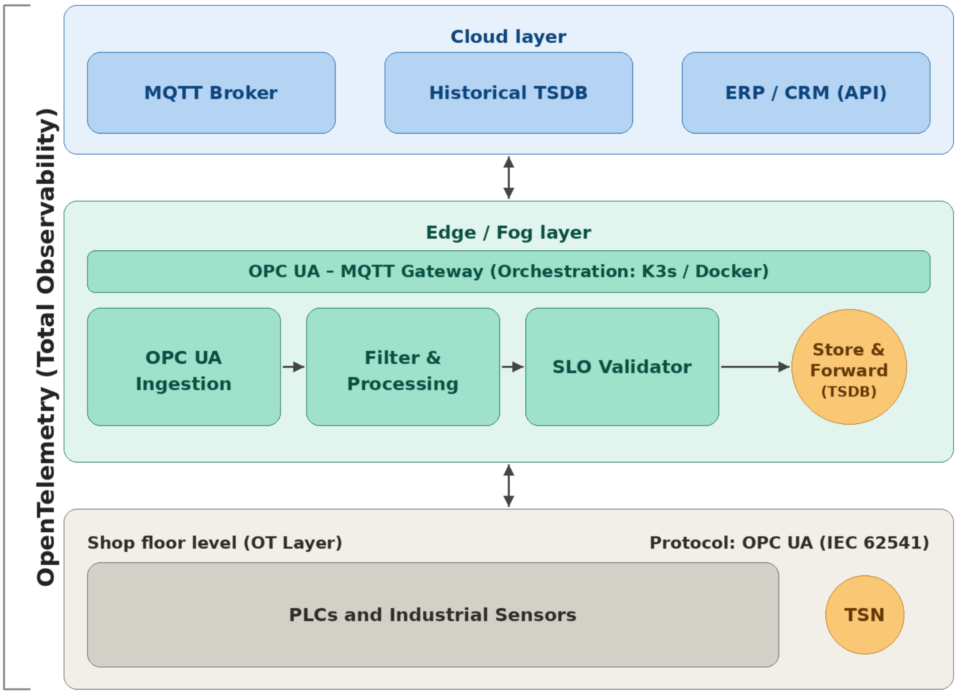

Figure 1. SLO-oriented hybrid architecture.

The OT layer comprises PLCs and industrial sensors exposing data via OPC UA (IEC 62541). The Edge layer acts as a communication gateway responsible for protocol translation, buffering, and resilience control. The Cloud layer hosts message brokering and integration services.

3.1. Edge Gateway and Resilience Mechanism

The core of the proposed solution is a containerized edge gateway implementing a resilience algorithm based on the store-and-forward pattern. The gateway was developed in Java and deployed using Docker containers orchestrated with Docker Compose, ensuring reproducibility and isolation.

The resilience mechanism adopts a hybrid buffering strategy that combines an in-memory FIFO buffer with configurable capacity (MAX_BUFFER_SIZE) and a persistent file-based storage layer to ensure data durability during prolonged failures. This hybrid design enables continued operation during connectivity disruptions while preserving data integrity.

At the current development stage, the system comprises four main functional modules: a Data Generation module that simulates industrial sensors by producing continuous telemetry messages; a Buffer Management module responsible for enqueue/dequeue operations and overflow control; a Persistence module that stores unsent messages on disk; and a Communication module that manages interaction with the cloud broker.

3.2. Communication and Cloud Integration

The edge gateway communicates with a cloud-based message broker implemented using RabbitMQ. Under normal conditions, messages are transmitted directly. During WAN failures, messages are buffered locally and later retransmitted in batches when connectivity is restored. This approach reduces dependency on continuous network availability and improves delivery reliability in unstable environments.

3.3. SLO-Aware Monitoring

System behavior is monitored using Service Level Indicators (SLIs), including latency, throughput, and delivery rate. In this work, SLOs are used for threshold-based evaluation of system performance. Adaptive control mechanisms based on SLO violations are considered as future work.

The behavior of the edge gateway is defined by guarded commands executed at each sampling interval ($\Delta t = 6$ s, for the simulation presented below). At each cycle, the gateway attempts to establish connectivity ($connected \leftarrow$ tryConnect()), after which the following conditions are evaluated: when connected and the buffer is not empty, buffered messages are transmitted; when connected and the buffer is empty, the current message is sent directly; when disconnected and the buffer is not full, the current message is stored locally; otherwise, if the buffer is full, the message is discarded.

3.4. Promising Outcomes: Latency and Resilience

Experimental validation utilized a dual approach: iFogSim2 [Mahmud et al. 2022] for performance scaling and a Java/Docker prototype for resilience. Performance tests with up to 200 sensors (20,000 msg/s) showed that edge processing maintained a 3.77 ms latency (P95: 5 ms), whereas cloud-centralized processing under stress spiked to 961.97 ms, exceeding the 150 ms SLO by over 600%. Resilience was evaluated via a 35-message cycle in a controlled laboratory setup using RabbitMQ, where WAN failures were injected to simulate industrial link instability. The system exhibited distinct phases: an initial successful transmission, followed by a Store phase in which a 10-message buffer absorbed transient failures. Under prolonged disruption, the buffer reached capacity, triggering FIFO-controlled drops (messages 21–22). Upon reconnection, a Forward phase prioritized historical data, recovering 83.3% of messages generated during downtime. Overall, the architecture achieved 94.3% delivery reliability, reducing data loss from 16 messages (no-buffer scenario) to just 2.

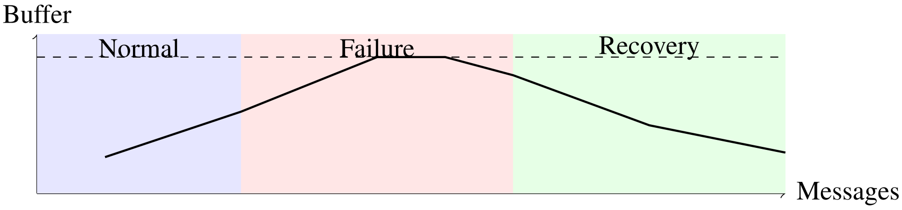

Figure 2. Buffer behavior under network instability.

As illustrated in Fig. 2, the buffer absorbs transient failures, saturates under prolonged outages, and is drained upon recovery through burst transmission.

3.5. Discussion of Trade-offs

The results highlight a fundamental trade-off between data generation rate and system autonomy. The required buffer memory can be approximated as:

$$M = T_{down} \times f \times S$$

where T_{down} is network downtime, f is message frequency, and S is message size.

Increasing data granularity improves observability but reduces the system's ability to operate during disconnections, as higher data rates accelerate buffer consumption.

3.6. Challenges and Trade-offs

Edge systems must balance sampling frequency, payload size, and storage capacity. Higher data rates improve observability but reduce resilience under disconnection. Additionally, reliance on a single gateway introduces potential single points of failure, indicating the need for distributed edge designs.

4. Conclusion

This work-in-progress presents a hybrid architecture for resilient industrial automation combining edge computing, microservices, and SLO-based governance. Preliminary results demonstrate latency improvements and resilience gains under network instability, while highlighting trade-offs between data granularity and system autonomy.

In this work, SLO usage is limited to monitoring and threshold-based evaluation, with adaptive control mechanisms left for future work. Future work also includes real-world validation, distributed edge deployments, and predictive adaptation strategies.

References

Al-Masri, E., Kalyanam, K. R., Batts, J., Kim, J., Singh, S., Vo, T., and Yan, C. (2020). Investigating messaging protocols for the internet of things (iot). *IEEE Access*, 8:94880–94911.

Bigheti, J. A., Fernandes, M. M., Risso, S. L., Arao, D. P., and Godoy, E. P. (2019). Controle de processos baseado em serviços para a indústria 4.0. In *Anais do I CONGRESSO BRASILEIRO DE INSTRUMENTAÇÃO, SISTEMAS E AUTOMAÇÃO (COBISA)*, volume 1, Brasil.

Bjørndal, N., de Araújo, L. J. P., Bucchiarone, A., Dragoni, N., Mazzara, M., and Dustdar, S. (2021). Benchmarks and performance metrics for assessing the migration to microservice-based architectures. *Journal of Object Technology*, 20(2):2:1–17.

Castilho, G. U. and Kamienski, C. (2018). Aplicação de computação em névoa na internet das coisas para cidades inteligentes: da teoria à prática. In *Anais do IX Workshop de Computação em Nuvem, Grade e Aplicações (WCGA)*, Brasil.

Gaffurini, M., Brandão, D., Rinaldi, S., Flammini, A., Sisinni, E., and Ferrari, P. (2025). Characterizing the Real-Time Communication Performance of Virtual PLC in Industrial Edge Platform. *IEEE Open Journal of Instrumentation and Measurement*, 4:1–11.

Hewa, T., Braeken, A., Liyanage, M., and Ylianttila, M. (2022). Fog computing and blockchain-based security service architecture for 5G industrial IoT-enabled cloud manufacturing. *IEEE Transactions on Industrial Informatics*, 18(10):7174–7185.

Hoppe, S., Graube, M., Iatrou, C., and Urbas, L. (2017). OPC UA meets MQTT – An industrial mapping concept. *at-Automatisierungstechnik*, 65(11):766–777.

Jammes, F., Karnouskos, S., Bony, B., et al. (2014). Promising technologies for SOA-based industrial automation systems. In Colombo, A. W., Bangemann, T., Karnouskos, S., Delsing, J., Stluka, P., Harrison, R., Jammes, F., and Lastra, J. L., editors, *Industrial Cloud-Based Cyber-Physical Systems*. Springer, Cham, Switzerland.

Kagermann, H., Wahlster, W., and Helbig, J. (2013). Recommendations for implementing the strategic initiative industrie 4.0: Final report of the industrie 4.0 working group. Technical report, acatech – National Academy of Science and Engineering, Berlin, Germany.

Kompally, V. S. (2025). A microservices-based hybrid cloud-edge architecture for real-time IIoT analytics. *Journal of Information Systems Engineering and Management*, 10(16s).

Kyösti, P. and Lindström, J. (2022). Integration of legacy systems with modern platforms using arrowhead framework. *Process Automation*.

Mahmud, R., Pallewatta, S., Goudarzi, M., and Buyya, R. (2022). ifogsim2: An extended ifogsim simulator for mobility, clustering, and microservice management in edge and fog computing environments. *J. Syst. Softw.*, 190(C).

Newman, S. (2015). *Building microservices*. O'Reilly Media.

Sharma, M., Tomar, A., and Hazra, A. (2024). Edge computing for industry 5.0: Fundamental, applications, and research challenges. *IEEE Internet of Things Journal*, 11(11):19070–19093.

Usman, M., Ferlin, S., Brunstrom, A., and Taheri, J. (2022). A survey on observability of distributed edge & container-based microservices. *IEEE Access*, 10:86904–86919.