

Article

A Scalable Microservices Architecture for Condition Monitoring and State-of-Health Tracking in Power Conversion Systems

José M. García-Campos ^{1,*}, Abraham M. Alcaide ^{2,3}, A. Letrado-Castellanos ², Ramon Portillo ²
and Jose I. Leon ^{2,3}

¹ Department of Telematics Engineering, University of Sevilla, 41092 Sevilla, Spain

² Department of Electronic Engineering, University of Sevilla, 41092 Sevilla, Spain

³ ENGREEN Laboratory of Engineering for Energy and Environmental Sustainability, University of Sevilla, 41092 Sevilla, Spain

* Correspondence: jgarcia111@us.es

Abstract

The role of power converters in modern electrical infrastructure (such as electric vehicle charging stations, battery energy storage systems and photovoltaic energy systems) has become critical. Given the high reliability required by these converters, continuous condition monitoring for predictive maintenance is mandatory. Traditional SCADA and HMI systems often face scalability bottlenecks and lack the flexibility in data aggregation and storage scalability required for long-term predictive maintenance. This paper proposes a scalable, containerized microservices-based architecture for degradation tracking and State-of-Health (SoH) monitoring in power conversion systems. The architecture features a decoupled four-layer structure, utilizing dedicated UDP servers for low-latency data ingestion, RabbitMQ (AMQP) for robust message routing, and a NoSQL (MongoDB) storage layer with a FastAPI interface. The proposed system was validated using a Hardware-in-the-Loop (HiL) setup with a Typhoon HIL606 simulator monitoring an Active Neutral Point Clamped (ANPC) power converter. Experimental stress tests demonstrated a Packet Delivery Ratio (PDR) of 1.0 at ingestion rates up to 100 messages per second (msgs/s) per node. The system exhibits transmission and processing overheads consistently below 5 ms, ensuring timely data availability for tracking thermal dynamics and parametric aging trends. This operational performance significantly exceeds the nominal requirement of 2 msgs/s for condition monitoring, ensuring robust data integrity. Finally, this modular approach provides the horizontal scalability necessary for Industry 4.0 integration, offering a high-performance framework for long-term health monitoring in modern power electronics.

Keywords: power converters; condition monitoring; predictive maintenance (PdM); microservices; scalability; Industry 4.0; Hardware-in-the-Loop; State-of-Health (SoH)



Academic Editors: Alin Dragomir
and Cristian Andriesei

Received: 7 January 2026

Revised: 5 February 2026

Accepted: 13 February 2026

Published: 16 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Societal energy demand is continuously growing with the massive adoption of electric vehicles, the proliferation of artificial intelligence agents and data-centers, among others. In this context, power converters are fundamental components in these modern electrical systems, serving as the interface for regulating and transferring power energy across a wide range of energy sources including renewable energy systems [1].

As demand grows for efficient power electronics systems, the reliability and availability of power converters become paramount [2]. Power converters are susceptible to

several operational failures, making them critical factors for the operation of the entire system [3]. This becomes particularly important in some specific applications where the power converter availability factor is very high (above 99%), such as medical equipment or aerospace systems [4,5].

Numerous studies have identified power devices and capacitors as the most critical failure-prone components within power electronic systems [6]. The degradation of these devices can lead to system failures, raising critical safety concerns and incurring significant economic consequences. Thermo-mechanical stresses induced by thermal cycling are a major cause of these failures and greatly increase the likelihood of wear-out mechanisms such as bond-wire fatigue and solder degradation [7,8]. Conversely, in power capacitors, the operating temperature is directly correlated with their Remaining Useful Lifetime (RUL).

Traditional maintenance strategies relied on reliability data provided by manufacturers, such as those found in standard reference manuals. These models often assume constant failure rates, overlooking the wear-out mechanisms typically observed in power devices [9]. However, degrading phenomena are much more complex; consequently, these maintenance plans suffer from poor accuracy, often yielding unreliable RUL estimates.

Therefore, to improve maintenance plans, it is crucial to consider the impact of various stress factors such as temperature fluctuations that can accelerate degradation and lead to premature failures for power devices and capacitors [10]. To achieve this, key thermo-electrical magnitudes should be captured and post-processed to identify failure modes and degradation phenomena in critical components at the converter level.

Traditionally, Human–Machine Interface (HMI) and Supervisory Control and Data Acquisition (SCADA) systems have been the dominant frameworks for operating power conversion systems, owing to their maturity and widespread industrial adoption [11]. While these systems provide essential supervisory functions and long-term data persistence through dedicated logging systems, they are primarily optimized for steady-state, plant-level monitoring. Their architectures are inherently centralized and rely on polling-based communication cycles [12], which often employ aggressive data compression and downsampling to minimize storage overhead. Consequently, they are ill-suited for capturing the high-resolution, high-frequency transient electrical signatures required to analyze the complex wear-out mechanisms and State-of-Health (SoH) of power converters [13]. Furthermore, the increasing density of power units in modern distributed grids creates significant scalability bottlenecks. These centralized architectures struggle to manage concurrent high-throughput data streams without compromising data integrity or increasing latency [14,15]. This landscape is changing with the emergence of Industry 4.0, the Internet-of-Things (IoT), and edge-computing paradigms.

Industry 4.0 and IoT have become widely adopted paradigms for the development of monitoring systems in multiple application domains [16], including precision agriculture [17], healthcare [18], industrial environments [19], transportation [20], among others [21]. While these applications span diverse sectors, the need for real-time, intelligent monitoring has also gained prominence in the electrical domain [22], particularly as power systems become increasingly complex and data-driven. Consequently, supervisory and monitoring systems have become essential for electric power systems [23], with numerous studies highlighting their importance for energy-related platforms. Representative examples include IoT-based monitoring of photovoltaic facilities, focusing on the acquisition and visualization of key variables such as power, current, and irradiance [24], supervision and control of microgrid operation [25], and monitoring and fault detection in wind turbines, enabling both condition assessment [26] and predictive maintenance strategies [27].

As these systems expand in scale in size and complexity, the volume of data exchanged increases, imposing stricter requirements on data integrity. Therefore, the choice of application-layer communication protocol [28] (which governs how information is formatted, transported, and processed) must be carefully considered to optimize hardware resource usage, energy consumption, connection stability, and overall data reliability.

Common application-layer protocols in IoT monitoring ecosystems include MQTT (Message Queuing Telemetry Transport) [29], a lightweight publish/subscribe protocol suitable for low-power sensor networks and machine-to-machine communication; CoAP (Constrained Application Protocol) [30], designed for resource-constrained devices with efficient request/response communication over UDP; and AMQP (Advanced Message Queuing Protocol) [31], which provides robust message delivery, flexible routing, and reliable TCP-based transport, making it well-suited for complex and high-throughput systems.

Consequently, selecting an appropriate protocol is critical for modern power electronics monitoring systems, particularly where reliable and efficient data collection is paramount. While MQTT and CoAP offer lightweight solutions for resource-constrained devices, their message delivery guarantees and routing capabilities often prove insufficient in complex, high-throughput environments. In contrast, AMQP provides robust message delivery guarantees, flexible routing through exchanges and queues, and reliable TCP-based transport, thereby guaranteeing data integrity. Its support for both point-to-point and publish/subscribe models, combined with security features, makes AMQP especially suitable for supervisory and monitoring systems in power electronics, enabling real-time analysis, predictive maintenance, and accurate tracking of critical components.

However, selecting an efficient communication protocol addresses only the transport layer of the monitoring challenge. As the number of monitored assets increases, traditional monolithic software architectures—where data acquisition, processing, and storage are tightly coupled in a single executable—often become a bottleneck [32]. These centralized structures face limitations regarding horizontal scalability and often lack the flexibility to integrate heterogeneous devices or isolate faults effectively. Consequently, leveraging Industry 4.0 capabilities requires adopting a modular software architecture. A microservices-based approach, deployed via containerization, provides the decoupling to ensure that high-frequency data ingestion does not compromise the performance of data processing or storage performance, thereby facilitating true horizontal scalability in distributed power electronic systems [33].

Despite recent advancements in communication protocols, a significant gap remains between general purpose monitoring architectures and the stringent temporal resolution requirements of power electronics condition monitoring. This limitation becomes evident when analyzing the shortcomings of contemporary solutions:

- **Next-Generation SCADA:** Modern SCADA systems have evolved towards cloud connectivity, yet their core design remains optimized for supervisory control and steady-state data acquisition (typically 1 Hz to 0.1 Hz) [34]. They are ideal for grid stability management but lack the temporal resolution to capture the sub-millisecond electro-thermal dynamics required for assessing the SoH of power converters [35].
- **Standard Edge IoT:** While generic IoT platforms leverage protocols like MQTT or HTTP for broad interoperability, they are designed for discrete messaging rather than continuous data streaming. Relying on these TCP-based protocols for direct waveform acquisition introduces timing irregularities that challenge the capability to sustain gapless high-speed data acquisition. These solutions are well-suited for environmental monitoring but struggle to sustain the continuous streaming of high-frequency waveforms needed for transient fault diagnosis [36].

Therefore, this paper proposes a scalable, containerized microservices-based architecture designed specifically for the real-time monitoring and fault diagnosis of distributed power conversion systems. Unlike traditional centralized approaches, the proposed framework employs a decoupled four-layer structure to isolate data ingestion from processing and storage tasks. The system integrates a low-latency UDP-based ingestion layer with an AMQP message broker, the system ensures reliable high-frequency data acquisition while maintaining the flexibility required to integrate heterogeneous sensors and advanced diagnostic algorithms without altering the core infrastructure.

The main contributions of this work can be summarized as follows:

- A novel decentralized monitoring architecture: A modular software framework based on containerized microservices that overcomes the scalability bottlenecks of monolithic SCADA systems, allowing for the seamless integration of new power assets is proposed.
- High-frequency ingestion strategy: A specific implementation combining UDP for low-latency data transmission with RabbitMQ (AMQP) for reliable asynchronous buffering, ensuring zero data loss even under high-load conditions typical of power electronics switching frequencies is considered.
- Experimental characterization of system limits: A performance benchmark using synthetic software-emulated nodes to stress-test the architecture, analyzing the PDR and delay to define the maximum throughput boundaries and scalability saturation points prior to hardware integration has been conducted.
- Experimental validation via HiL: The proposed system is validated using a Typhoon HIL606 real-time simulator modeling an ANPC converter, demonstrating the system's capability to handle ingestion rates up to 100 msgs/s per node with a Packet Delivery Ratio (PDR) of 1.0.

The remainder of this paper is organized as follows. Section 2 analyzes the current challenges in data observation and monitoring for power conversion systems, emphasizing the necessity of high-frequency data capture to accurately determine the SoH of critical components. Section 3 details the proposed modular software architecture. Section 4 presents an experimental characterization of the system's performance limits using synthetic software-emulated nodes to evaluate horizontal scalability and resource saturation boundaries. Section 5 validates the complete architecture in a realistic HiL environment, demonstrating its operation with an ANPC converter under real-time conditions. Finally, Section 6 summarizes the main contributions and outlines future research directions.

2. Data Observation and Monitoring Issues

Studying the behavior of a physical system by observing its operational characteristics allows for proactive maintenance planning, which helps prevent significant deterioration or failure, a process known as condition monitoring (CM) [37]. As previously noted, power devices and capacitors in power converters are susceptible to failure due to thermal cycling (55% of cases), humidity or moisture (19%), contamination or dust (6%), and vibration (20%) [38]. These statistics underscore the importance of CM techniques for detecting early signs of degradation and scheduling maintenance accordingly.

In power converters the most critical components are power devices and capacitors. In both cases, their operational temperature is a key factor. In Figure 1, a cross-section of a power device module is shown highlighting critical components such as bond wires.

For power devices, there is a relationship between the number of thermal cycles of their junction temperature can undergo. This relationship is given by:

$$N_f = a\Delta T^{-\alpha} e^{\left(\frac{E_a}{k_B T_{jm}}\right)} \quad (1)$$

where the coefficients a , α , and E_a are obtained by experimental fitting; k_B is the Boltzmann's constant; ΔT is the thermal cycle amplitude (defined as the thermal swing); and T_{jm} is the average junction temperature. These effects are illustrated in Figure 2, using a Semikron power electronics module as an example [39].

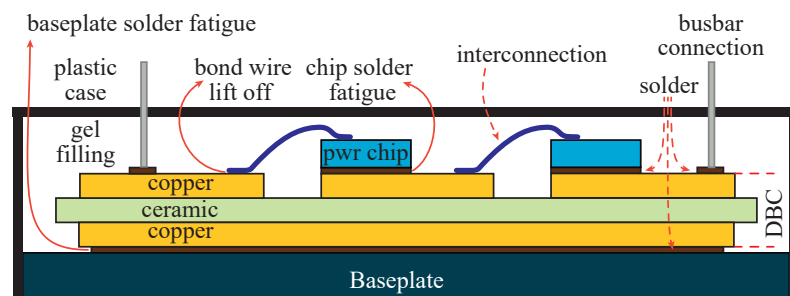


Figure 1. Cross-section of a conventional power module, and typical faults and their location.

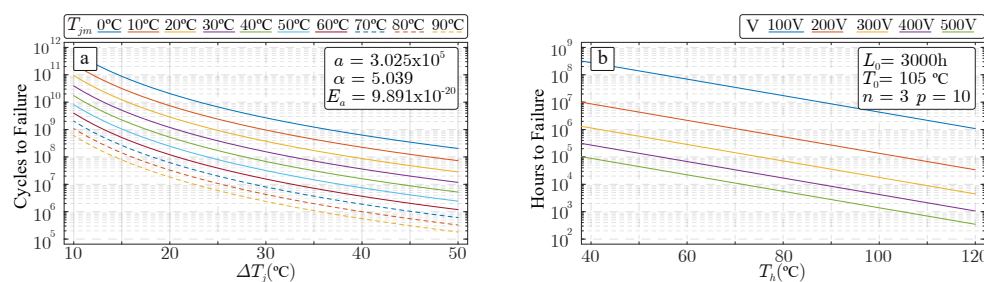


Figure 2. (a) Temperature cycle capability characteristic N_f for a standard Semikron IGBT module. (b) Reliability model for electrolytic capacitor.

Directly measuring junction temperature in operational power devices within power converters is challenging; consequently, estimation techniques must be employed to monitor its behavior [40,41]. This estimation can be performed by using their conduction and switching loss to subsequently calculate its T_j temperature by using Cauer or Foster thermal networks (see Figure 2b) [42,43]. Alternatively, T_j temperature can be obtained by observing the evolution of several thermo-sensitive power device parameters [40,44].

Similarly, power capacitors' lifetime is closely related to their operational hot-spot temperature (see Figure 2b). In this regard, the Capacitor's RUL can be determined by:

$$L_c = L_0 \left(\frac{V}{V_0} \right)^{-n} 2^{\frac{T_0 - T_h}{\rho}} \quad (2)$$

where parameters L_0 , T_0 and V_0 are the nominal lifetime, temperature and voltage under the test conditions. Temperature T_h is the hot-spot temperature of the device and coefficients n and ρ depend on the capacitor technology. For instance, for electrolytic capacitors n is defined in the range of [3–5] and ρ is equal to 10 [45,46].

In this case, T_h temperature can be estimated by considering the i_{RMS} thorough the device.

Therefore, it is clear that most relevant electrical magnitudes should be captured for further analysis and post-processing in order to determine the current SoH of the

power converter and refining the maintenance scheduling of most critical components in power converters.

3. Proposed Architecture

3.1. System Overview

This work proposes a scalable, experiment-oriented architecture for the long-term condition monitoring and electrical analysis of power electronic converters. Built on a containerized microservices paradigm to ensure portability and fault isolation, the system is organized into four hierarchical layers:

1. Data acquisition and protocol adaptation: Interfaces directly with power converters to capture and packetize raw high-frequency data.
2. Message-oriented middleware: Acting as the communication backbone, this layer utilizes a message broker to decouple data producers from consumers.
3. Data processing and subscription services: Composed of independent subscriber microservices that consume data streams for processing and persistence, ensuring horizontal scalability.
4. Data storage and access interfaces: Manages the storage of heterogeneous data in a NoSQL database and provides a RESTful API for standardized retrieval and integration.

Figure 3 depicts the architecture's conceptual diagram and information flow. This decoupled design ensures that each stage operates independently, facilitating efficient data processing and storage.

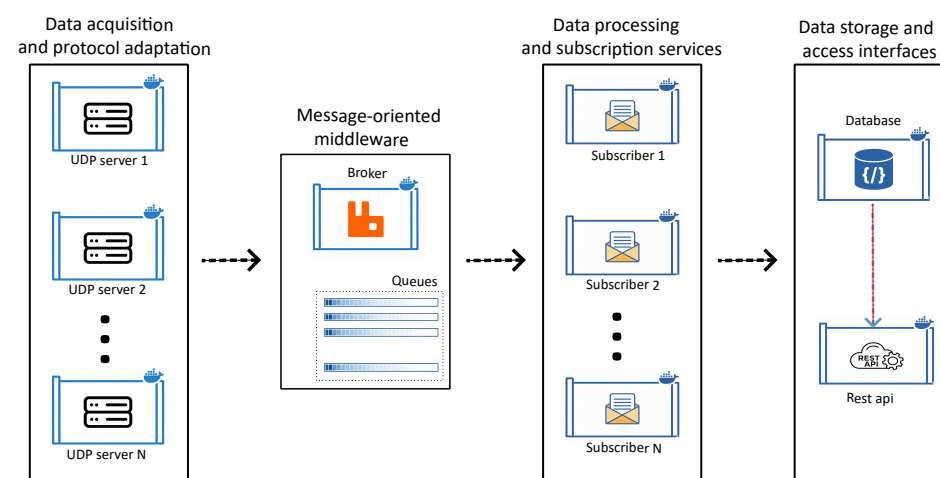


Figure 3. Proposed architecture scheme.

Each power converter under observation is treated as an independent data source, while experiments are defined as logical entities associated with a specific converter and operating condition. This abstraction enables the simultaneous execution of multiple experiments across different converters without introducing tight coupling between system components. Furthermore, a key feature of the proposed architecture is its ease of deployment, which is centralized through a single configuration file that registers all target converters and experiments; this approach minimizes manual setup and supports rapid scaling across multiple devices. To ensure reproducibility and facilitate the practical application of this architecture, the complete source code is publicly available in [47].

3.2. Data Acquisition and Protocol Adaptation Layer

This layer bridges the physical converters and the digital ecosystem, transforming raw high-frequency data for asynchronous processing. A dedicated, containerized UDP

server is instantiated for each monitored converter [48]. The selection of UDP is based on the inherent capabilities of the converter hardware and for its minimal overhead and high throughput, which are critical for constrained micro-controllers in high-frequency telemetry [49]. To mitigate UDP's lack of delivery guarantees, the layer implements an application-level verification process to ensure data consistency before propagation. While UDP is used here to meet specific timing constraints, the architecture remains protocol-agnostic, allowing the adoption of alternatives like TCP where hardware resources permit [50].

3.2.1. Message Reception and Validation

Incoming messages are received as raw byte streams. Upon reception, the data undergo a sequence of processing steps to ensure structural integrity and semantic consistency before being forwarded to subsequent stages. Messages transmitted by the power converters are required to conform to the predefined format illustrated in Figure 4, which serves as the basis for the validation checks applied during reception.

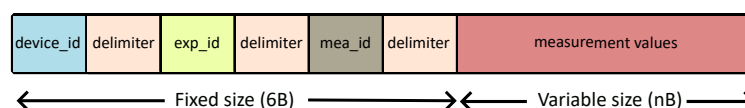


Figure 4. Raw message format.

The format illustrated in Figure 4 provides a general overview of the structure of the messages sent by the power converters. To better understand the composition of each message, Table 1 presents a detailed description of the individual fields, including their size and intended purpose.

Table 1. Raw message.

Field	Size (Bytes)	Description
device_id	1	Power converter identifier
exp_id	1	Identifier of the experiment
mea_id	1	Identifier of the measured quantity
measurement values	4xn	Values of the measurement
delimiter	1	Field separator

Message reception involves the following validation steps:

- (i) Message length validation: The total message length, in bytes, is verified against the expected structure (see Figure 4). The first part of the message is fixed at 6 bytes, with each field occupying 1 byte, while the measurement values field is variable in size, depending on the type of measurement being performed. The validation ensures that the total length (see Equation (3)) equals 6 plus a multiple of 4 bytes, since the measurement values field consists of one or more values, each encoded as a 32-bit floating-point number

$$L_{\text{total}} = L_{\text{fixed}} + L_{\text{variable}}, \quad \text{with} \quad L_{\text{fixed}} = 6 \text{ bytes}, \quad L_{\text{variable}} = 4 \cdot n, \quad n \in \mathbb{N} \quad (3)$$

where n is the number of measurement values.

- (ii) Message decoding: The byte stream is decoded into structured values by interpreting the fixed-size identifier fields (device_id, exp_id, mea_id and delimiter in Figure 4) as single-byte numerical values, while the measurement values (see Figure 4) field is decoded by converting each group of four bytes into a 32-bit floating-point number.

- (iii) Measurement ID validation: The system defines a finite set of supported measurement values, representing all possible types of measurements that can be generated by the converters. The decoded measurement identifier (mea_id in Figure 4) is checked against this set.

This list should be updated according to the specific power converter topology and/or industry application. As an example, Appendix A lists all supported measurement types, representing the finite set of values that can be generated by the power converters and validated by the system. Notably, vector magnitudes length is a function of the fundamental frequency of the power system and the operating frequency of the converter. For instance a power converter connected to the grid (50 Hz) operated at 1 kHz generates vectors of $1000/50 = 20$ elements. However, this length depends on the computational capabilities of the control platform of the power converter.

3.2.2. Message Reconstruction and Broker Integration

Validated data is encapsulated into a structured header–payload message (see Figure 5) for middleware transmission. The header manages routing context, while the payload contains the measurement data defined in Table 1, appended with an acquisition timestamp.

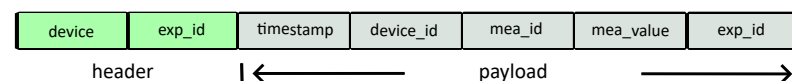


Figure 5. Reconstructed message format with header-payload structure.

3.3. Message-Oriented Middleware Layer

Serving as the asynchronous communication backbone, this layer is implemented using AMQP via a single containerized RabbitMQ broker. AMQP was chosen for its strong delivery guarantees and buffering, while RabbitMQ was selected for its maturity and proven performance. To ensure logical isolation, a dedicated headers exchange is assigned to each power converter, acting as the specific entry point for its generated data. Messages published to these exchanges—comprising header fields and payload (see Figure 5)—are routed to experiment-specific queues based on attributes such as the exp_id. This setup ensures logical isolation between converters, preventing cross-device data interference, and allows multiple experiments to run concurrently on the same converter while maintaining strict data separation.

Table 2 summarizes the key technical characteristics and configuration parameters of the middleware layer.

Table 2. Key characteristics of the message-oriented middleware layer.

Feature	Description	Purpose
Protocol	AMQP	Standardized communication
Broker	RabbitMQ	AMQP implementation
Exchange	Headers	Header-based routing
Bindings	Based on exp_id and device	Experiment and device separation
Scalability	Horizontal	System expansion

To illustrate the operation of the message-oriented middleware layer, Figure 6 presents a representative example involving two power converters. In this example, each data acquisition service associated with a given converter (deviceName1 and deviceNameN in Figure 6) acts as a producer and publishes measurement data to the containerized RabbitMQ broker using AMQP.

Within the broker, each power converter is associated with its own headers exchange (e.g., deviceName1 or deviceNameN), which receives all messages originating from that device. Each message contains a payload with the measured values, among other data, and a set of header fields, including the power converter name and the experiment identifier. In the example illustrated in Figure 5, the message header fields take the values device = deviceName1 and exp_id = exp_2, which determines the routing of the message to the corresponding experiment-specific queue. Regarding the payload, the timestamp is 1,766,508,591.4, the device_id is 1, the mea_id is 5, the mea_value is [1, 2, 3]. Finally, the exp_id is exp_2. Eventually, the payload of the message will be stored in its corresponding queue (deviceName1_exp_2) until the appropriate consumer retrieves it.

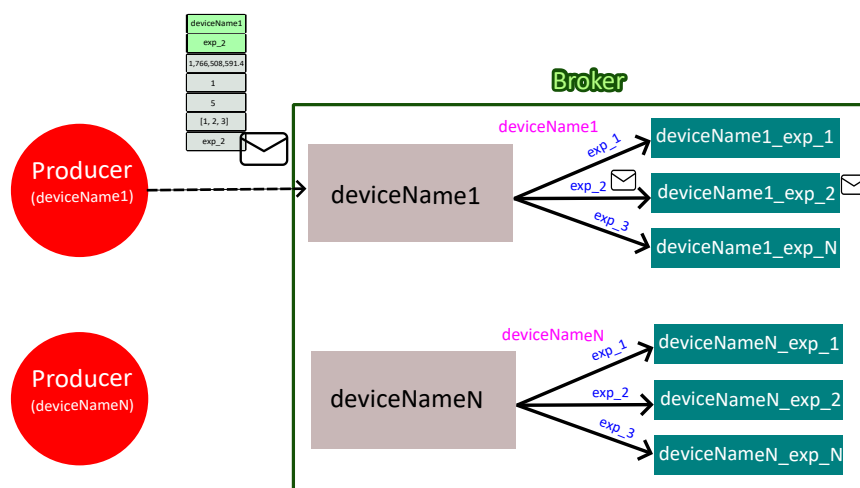


Figure 6. Information flow and routing within the message-oriented middleware.

3.4. Data Processing and Subscription Services

The processing layer consists of independent subscriber microservices, each deployed as an isolated container responsible for persisting data from a specific experiment's queue of a given power converter. This one-to-one mapping ensures fault isolation and facilitates horizontal scaling. Data reliability is guaranteed through RabbitMQ acknowledgments, where messages are removed from the queue only after a confirmed database write.

Upon consumption, each subscriber executes a two-stage pipeline to guarantee data integrity:

1. **Experiment confirmation:** Messages are cross-referenced with the active experiment's metadata based on the exp_id header. This secondary validation acts as a safeguard against routing errors, filtering out mismatched or malformed packets to ensure strict consistency.
2. **Storage in database:** Validated data is immediately written to the corresponding storage collection. This ensures near real-time persistence, preserving chronological integrity for subsequent retrieval.

Figure 7 depicts the specific data processing and subscription workflow described above. In this diagram, two distinct groups of subscribers can be identified: those corresponding to the queues of deviceName1 and those assigned to the queues of deviceNameN. As shown, the total number of subscriber instances, represented by pentagonal blocks in Figure 7, matches the number of active queues (six in this example).

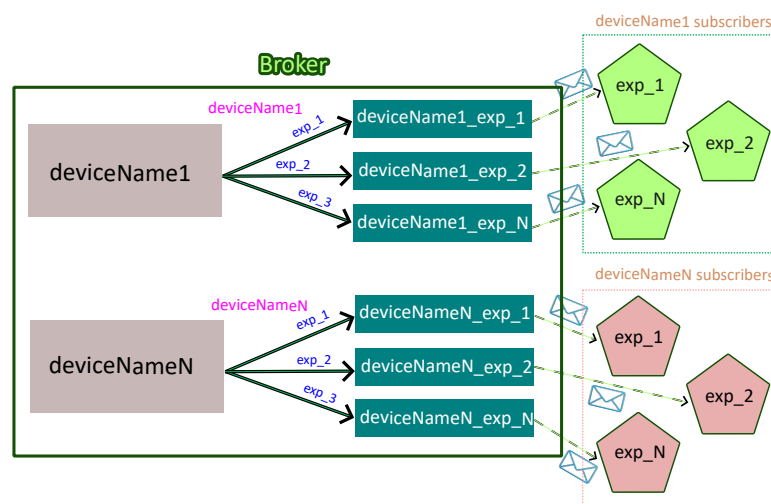


Figure 7. Architecture of the data processing and subscription layer.

3.5. Data Storage and Access Interfaces

This layer ensures long-term persistence and structured access to subscriber-processed measurements using a non-relational MongoDB database [51]. MongoDB was selected for its native support of complex document structures, avoiding the rigidity of relational schemas [52] and the inefficient data flattening required by standard TSDBs (e.g., InfluxDB) for high-dimensional vectors [53]. This enables the atomic storage of vectors, preserving the semantic integrity of heterogeneous datasets within a single record [54]. Comparative research on IoT workloads [55] validates this choice. While InfluxDB suffers from CPU spikes exceeding 600% under high concurrency, MongoDB demonstrates superior stability with moderate CPU growth ($\approx 145\%$), ensuring scalable performance in containerized environments.

The database is structured by power converter, using distinct collections for each experiment to ensure isolation and scalability. To guarantee persistence, Docker volumes are employed to decouple the storage engine from the container lifecycle, preventing data loss during system updates or restarts. To maximize ingestion speed, subscribers perform direct write operations into MongoDB [56]. Bypassing intermediate layers minimizes communication overhead and prevents bottlenecks during high-frequency sampling.

A read-only RESTful API, built on the FastAPI framework [57], provides structured access to stored data. FastAPI was selected for its asynchronous execution, ensuring high performance under concurrent high-frequency requests [58]. The interface abstracts the underlying schema, delivering JSON payloads filtered by experiment, converter, or measurement identifiers. This interface also facilitates the integration of advanced diagnostic algorithms based on artificial intelligence or machine learning. By providing programmatic access to historical datasets, external AI agents can retrieve training data or perform offline validation of predictive maintenance models (e.g., RUL estimation) without interfering with the real-time data acquisition process. Furthermore, it facilitates aggregating data from multiple converters to train unsupervised learning models that identify irregular operating patterns across a fleet.

Security [59] is currently managed through private network isolation, suitable for a controlled laboratory, thus omitting public authentication. However, the architecture is fully compatible with industrial standards like OAuth2 and TLS [57], ensuring seamless adaptation for production environments where these protocols are mandatory.

Both the database and API operate as independent containerized services. This decoupling allows them to scale via replication in response to increased load without disrupting the upstream acquisition and processing layers.

3.6. System Deployment and Configuration

The deployment of the proposed monitoring system is designed to be modular and automated, ensuring the horizontal scalability and fault isolation principles discussed in Section 4.1. This process is managed by an orchestration layer by using a shell-based script that parses a centralized YAML configuration file to dynamically generate the infrastructure via Docker Compose.

The deployment workflow is executed in three sequential phases:

1. **Network Infrastructure and Service Discovery:** The orchestrator establishes a dedicated Docker bridge network to facilitate internal service discovery via logical hostnames, eliminating reliance on volatile static IPs.
2. **Core Services Deployment:** The script initializes the foundational, singleton components of the architecture: the MongoDB instance, the RabbitMQ message broker, and the RESTful API for data access. These services constitute the system's backbone and are deployed using a static Docker Compose configuration.
3. **Dynamic Service Instantiation and Specialization:** For scalable components, the orchestrator performs dynamic instantiation using standardized Docker images. Individual container "specialization" is achieved via runtime parameter injection during the generation of specific Docker Compose files:
 - **UDP servers:** A unique container is instantiated per power converter. Shared images are parameterized with specific listening ports and converter identifiers, which determine the target RabbitMQ exchange for message publication.
 - **Subscribers:** Similarly, independent containers are instantiated per active experiment. Runtime parameters inject specific RabbitMQ routing details (exchange/queue), experiment identifiers, and MongoDB connection details (IP and port).

Finally, the orchestrator executes the `docker-compose up` command for all dynamically generated configurations. This automated approach minimizes manual intervention, mitigates configuration drift, and enables the system to scale from a single converter to a high-density experimental environment in seconds.

An example of this configuration file is shown in Appendix B.

4. Experimental Characterization of System Limits

The performance of the proposed monitoring architecture has been evaluated by characterizing its operational limits in terms of data reliability and delivery latency point of view. For this purpose, two primary metrics [60] are considered: the Packet Delivery Ratio (PDR), which measures the system's capacity to reliably transport measurement data, and the delivery latency, defined as the time elapsed between the generation of a measurement and its successful persistence within storage layer.

The PDR figure of merit is defined as the ratio of successfully received messages to the total number of transmitted messages as shown in (4), expressed as a percentage:

$$PDR = \frac{N_{\text{received}}}{N_{\text{sent}}} \times 100\% \quad (4)$$

where N_{received} is the number of messages successfully received by the system, and N_{sent} is the total number of messages transmitted. In this context, a sent message refers to a message generated and transmitted by a (synthetic) power converter at a given point in time, whereas a received message is one that has been successfully stored in the database, ensuring it is available for subsequent processing and analysis.

Similarly, the architectural delay figure of merit is calculated as the time difference between the dispatch of the data packet and its final storage (5):

$$delay = t_{received} - t_{sent} \quad (5)$$

where t_{sent} and $t_{received}$ represent the timestamps at which a message is sent and received, respectively. It is worth noting that this metric isolates the communication and processing overhead introduced by the proposed architecture, excluding the intrinsic physical acquisition window (≈ 20 ms for a 50 Hz fundamental cycle) required to construct the data vector.

The experimental analysis comprises two main phases. First, the maximum throughput that a single power converter can sustain without compromising data integrity or system responsiveness is determined (Section 4.1). Second, the system's horizontal scalability is assessed by identifying the maximum number of concurrent converters that can be supported under high-load conditions (Section 4.2). These experiments yield insights into the architectural bottlenecks and offer practical guidelines for real-world deployments.

To validate the high efficiency and resource optimization of the proposed architecture, all experimental tests were conducted on a Raspberry Pi 4 Model B (4 GB RAM) running the Raspberry Pi OS operating system [61]. This hardware selection underscores the system's lightweight design, demonstrating its ability to operate effectively on cost-efficient, resource-constrained edge computing devices without relying on high-performance server infrastructure.

Furthermore, it must be emphasized that the quantitative results presented are intrinsically linked to the specific hardware and software environment employed and should not be directly extrapolated to other architectures. While this study provides a clear baseline for the system's behavior, a dedicated experimental characterization would be necessary for any alternative deployment to identify the specific operational boundaries and limitations of the corresponding architecture.

4.1. Impact of Message Rate on Reliability and Delay

Adhering to the methodology described above, this section evaluates the performance of a single monitoring node under varying ingestion rates. The primary objective is to determine the maximum throughput that the architecture can handle prior to compromising service quality in terms of reliability and latency.

To achieve this, the characterization is structured into five distinct experimental scenarios (see Table 3). Each experiment corresponds to a fixed message generation rate, from 1 to 10,000 messages per second (msg/s). To ensure statistical significance and account for potential outliers arising from inherent system and network variability, each scenario was independently repeated for 15 iterations. Each iteration lasted 10 min, providing a robust data sample to calculate consistent average values for both PDR and delay.

The PDR results (see Table 3) reveal a clear operational boundary for the proposed architecture on the selected hardware. While the system maintains perfect reliability (PDR = 1.0) for ingestion rates up to 100 msg/s (Experiments 1–3), a significant throughput saturation is observed as the rate scales further. In Experiment 4 (1000 msg/s), the PDR drops to 0.17, indicating that the system can no longer process the incoming message bursts in real-time, leading to buffer overflows or message rejection at the broker level. This trend is exacerbated in Experiment 5 (10,000 msg/s), where the PDR reaches 0.034. These results suggest that the effective processing capacity of a single-node Raspberry Pi 4 deployment peaks at approximately 340 messages per second under the current configuration.

The results for the delivery delay are summarized in Table 3. Remarkably, the system maintains a sub-5 millisecond delay across all experimental scenarios, with values ranging from 2.85 ms to 3.95 ms.

In the first three experiments (1 to 100 msg/s), the delay shows a slight downward trend, stabilizing at approximately 2.85 ms. This behavior is typical in message-oriented middleware, where the initial overhead of processing sporadic packets is amortized as the ingestion rate increases and the subscriber microservices reach a steady-state execution flow [62].

Table 3. Message rate—features and results.

Experiment	Messages/s	PDR	Delay (ms)
1	1	1.0	3.95
2	10	1.0	3.58
3	100	1.0	2.85
4	1000	0.174	3.05
5	10,000	0.034	3.27

Importantly, these experiments utilize synthetic message streams rather than physical converters. Therefore, the results reflect the limits of the monitoring infrastructure—including the message broker, data processing, and storage layers—rather than the operational behavior of actual power converters. These findings provide practical guidance for defining operating parameters and identifying thresholds where system performance may degrade.

4.2. Scalability Analysis with Multiple Simulated Converters

Building upon the performance characterization established in the previous section, this subsection evaluates the horizontal scalability of the proposed architecture. The primary objective is to determine the operational limits of the system when transitioning from a single-node setup to a multi-device environment, thereby assessing its suitability for large-scale power conversion plants. Following the findings from Section 4.1, which identified 100 msg/s as the stability threshold for the current hardware configuration, this experimental phase maintains a constant aggregate throughput of 100 msg/s.

This workload is distributed across an increasing number of synthetic power converter instances (N). It is important to emphasize that these nodes do not consist of physical hardware; instead, they are synthetic entities emulated via software. This approach allows for an isolated assessment of the impact of concurrency and resource orchestration on system stability, ensuring that the performance evaluation focuses strictly on the architecture's ability to manage multiple simultaneous data streams.

The experiments are defined as follows:

To ensure statistical reliability and maintain consistency with the methodology established in Section 4.1, each experimental configuration was conducted over 15 iterations. The results obtained from this scalability characterization are summarized in Table 4, which present the average PDR and delivery delay for each experiment.

Table 4. Scalability—features and results.

Experiment	Device Number	PDR	Delay (ms)
1	2	1.00	3.14
2	4	1.00	8.93
3	5	0.92	24.85
4	6	0.78	27.65

These results reveal that the architecture maintains its operational integrity up to a limit of $N = 4$ concurrent devices. While for $N \leq 4$ a PDR of 1.0 and delays below 9 ms are guaranteed, from $N = 5$ onwards, a significant degradation occurs, with the PDR dropping to 0.92 and the delay increasing non-linearly to 24.85 ms. This behavior identifies $N = 4$ as the saturation threshold for the evaluated hardware, where instability does not derive from the total data volume—maintained constant at 100 msg/s—but from the overhead of managing multiple independent network streams and the resource contention among containerized instances for the system's physical resources.

4.3. Implications for System Deployment

The experimental characterization of the proposed architecture reveals a significant operational margin that is critical for industrial power conversion applications. The stress tests identified a maximum stable throughput of 100 msg/s per monitoring node before reaching saturation. However, the specific requirements for condition monitoring, focused on tracking slow-evolving degradation variables like thermal stress or electrolyte loss, are typically satisfied with a sampling rate of 2 msg/s per power converter.

Consequently, under nominal operating conditions, a single converter utilizes only 2% of the node's verified communication capacity. This extensive headroom is a deliberate architectural advantage that ensures two critical deployment capabilities:

1. **Vertical Scalability and Reliability:** Operating at a fraction of its total capacity allows a single low-cost edge device (e.g., Raspberry Pi 4) to concurrently monitor multiple power converters—theoretically up to 50 units at the target rate—without compromising stability. Furthermore, this 98% safety margin provides the necessary resilience to handle unpredictable data bursts or high-priority traffic during transient fault events, preventing packet loss and ensuring deterministic latency in critical situations.
2. **Edge Computing Readiness:** By maintaining a lightweight communication overhead, the system preserves the majority of the node's computational resources (CPU and memory) for the execution of the diagnostic layer. This enables the implementation of real-time health-state estimation algorithms, such as Fast Fourier Transform (FFT) analysis or machine learning-based classification, directly at the edge.

5. Use Case—Hardware-in-the-Loop Active Neutral Point Clamped Power Converter Real Time Simulation

To evaluate the proposed architecture under realistic conditions, this section presents a validation case study centered on the monitoring of a grid-tied Photovoltaic (PV) Power Plant. The aim is to validate the system's ability to acquire, process, and store high-frequency telemetry data from complex power electronics system without disrupting ongoing operations.

The monitoring architecture was validated using a Hardware-in-the-Loop (HiL) real-time simulator, specifically the HIL606 from Typhoon. Consistent with the characterization presented in Section 4, the acquisition system was deployed on a Raspberry Pi 4 Model B (4GB RAM) to replicate a resource-constrained edge environment. In this scenario, the HIL device emulates an Active Neutral Point Clamped (ANPC) converter serving as the electrical interface for the PV plant. An scheme of the system is shown in Figure 8 where the emulated power system architecture based on the ANPC power converter is also drawn.

The experimental setup is established within a controlled laboratory environment. Concerning the network topology, both the Typhoon HIL606 and the Raspberry Pi are connected to the same dedicated Local Area Network (LAN) via Gigabit Ethernet switch device. This configuration enables the data acquisition platform operates in an air-gapped

mode (i.e., isolated from the Internet), thereby preserving high-frequency telemetry integrity and preventing packet loss due to external network congestion. This aligns with standard practices in power electronic systems, where converters remain isolated from the Internet for safety, restricting external connectivity to administrative purposes.

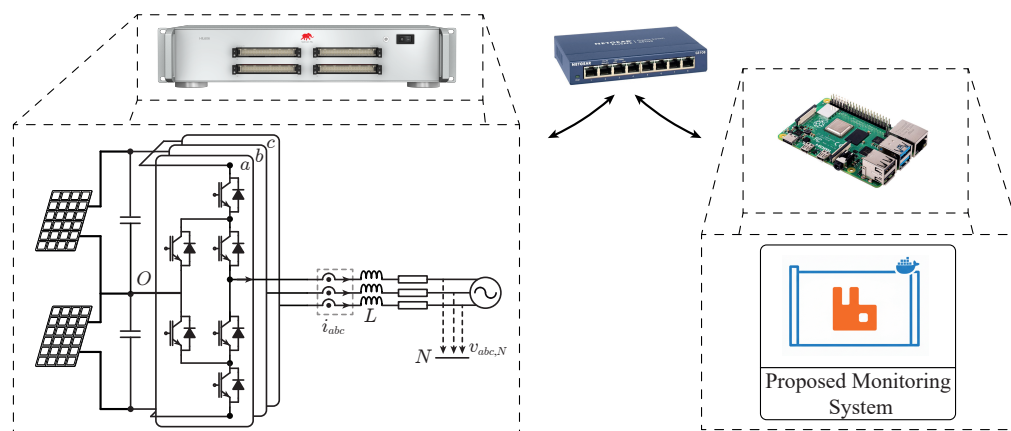


Figure 8. Experimental setup considering HIL system and the proposed monitoring system.

In this validation scenario, the Typhoon HIL simulator emulates the ANPC converter following a pre-defined mission profile (that means, imposing active and reactive power references (P^* , Q^*) for a PV application). The control scheme used in this scenario follows the conventional dq synchronous-frame based on linear controllers for the sake of simplicity and the modulation technique is based on conventional level-shifted pulse width modulation (LS-PWM). After starting up the system, the HIL606 is streaming critical telemetry data to the proposed acquisition platform. The data export adheres to the classification criterion reported in Table A1, encompassing measurements such as three-phase grid voltages, phase currents, and estimated IGBT conduction losses. Specifically, the transmission strategy follows a multi-rate approach: messages in the catalog from IDs 1 to 6 (corresponding to internal vector magnitudes) are dispatched periodically every 5 s, assuming a steady state (sending a vector containing information for one single period), whereas messages from IDs 10 to 15 (containing average or RMS metrics) are transmitted at 1 s intervals. This injection profile, which interleaves instantaneous values for transient analysis with downsampled metrics for steady-state supervision, effectively emulates the complex and heterogeneous data traffic patterns typical of real-world industrial converters.

As a result of the HIL606 operation, some data have been captured in the proposed monitoring system from emulated power converter and lately recovered from the database for presentation purposes in this manuscript. An example of internal data vector capturing is shown in Figure 9 where the grid voltage is shown.

As mentioned, the datalogging process has performed using a multi-rate process. To demonstrate this fact, RMS values of significant magnitudes have been capturing by message ID 10, 11 and 13 during a portion of time equals to 15 s. These trends are represented in Figure 10 and Figure 11, respectively. This feature is specially important to determine if the power system is correctly operated. In the case of conduction loss of the power devices of the phase a are internally sampled and forwarded to the monitoring system for later processing. It is worth to be mentioned the amount of transmitted data, specially for message ID = 13 where the average conduction loss of 12 devices are logged without data-loss, an example of the capture for data vector is shown. This functionality is very helpful for early fault detection and/or RUL observation because it is possible to capture correlated data within the power converter.

Listing 1 shows a raw data fragment from the MongoDB collection, illustrating how messages with different measurement IDs (id_mea) and vector lengths (mea_value) are chronologically persisted as they arrive from the microservices network.

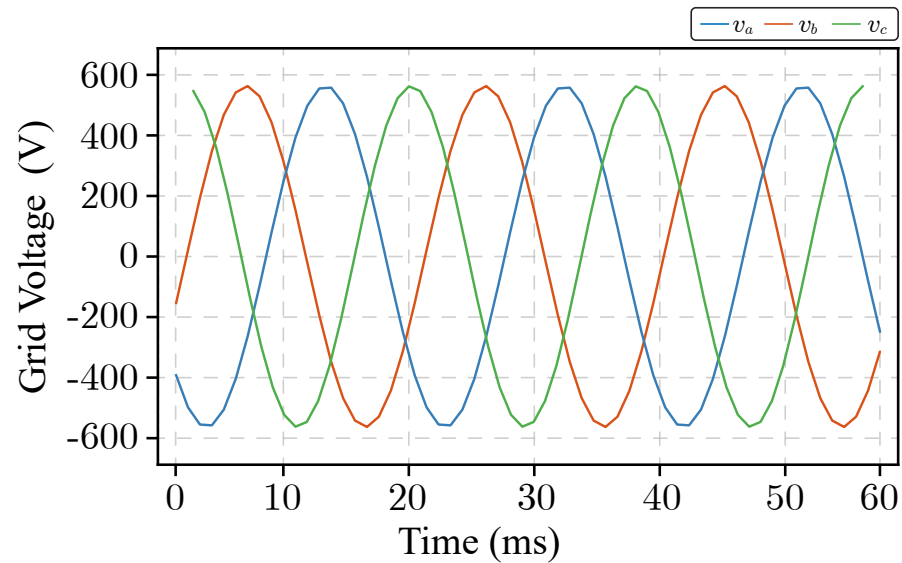


Figure 9. Grid voltage captured in the Typhoon HIL system and sent to the monitoring system.

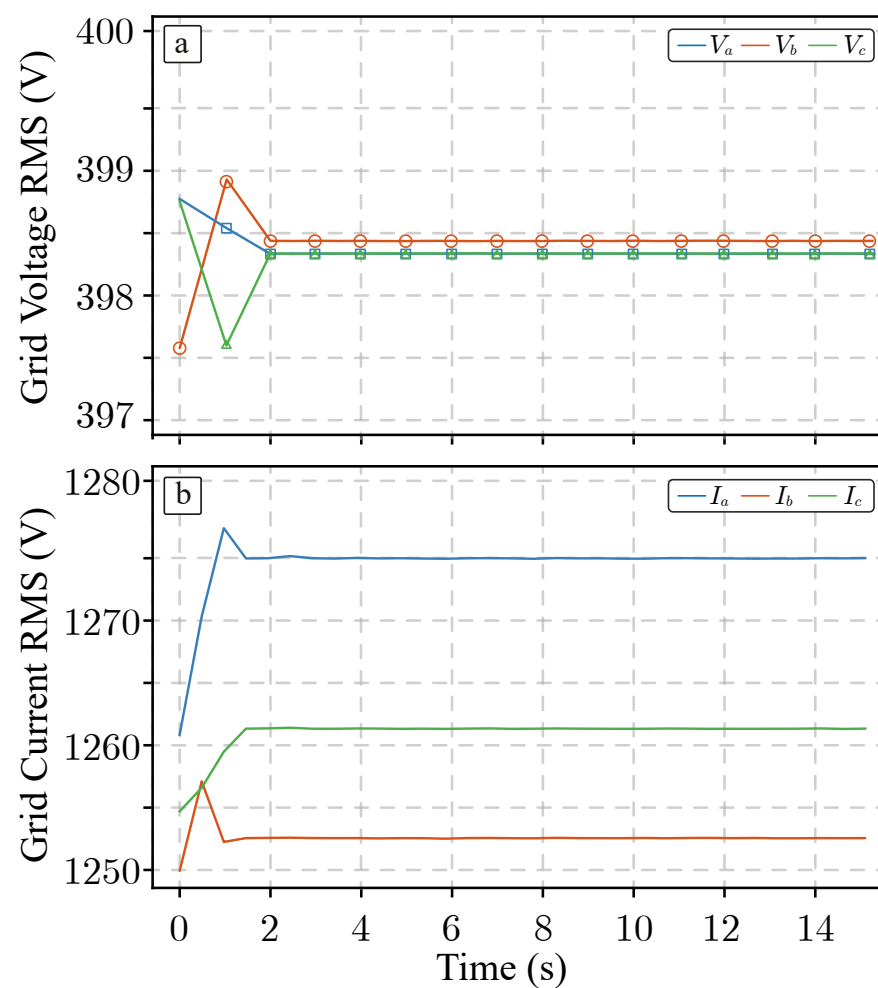


Figure 10. Magnitudes capture in the Typhoon system and sent to the monitoring system: (a) grid voltage RMS; (b) output current.

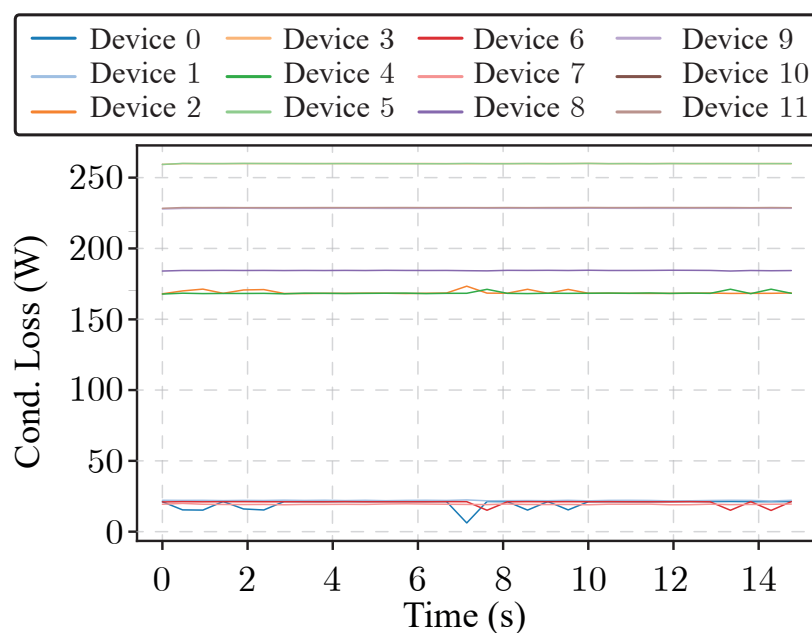


Figure 11. Conduction losses of the power device presents in phase a captured by the Typhoon HIL system and sent to the monitoring system.

Listing 1. Representative subset of data persisted in MongoDB, illustrating the interleaving of messages with varying measurement IDs and vector lengths.

```
{
  "_id": "695437fceca9cc949c9ec03a",
  "timestamp": 1767127033.4299982,
  "device": "typhoon",
  "id_mea": 11,
  "mea_value": Array (3) [
    0: 1251.0855712890625,
    1: 1261.38623046875,
    2: 1255.5899658203125
  ],
  "experiment_id": "experiment_3"
}
{
  "_id": "695437fceca9cc949c9ec03c",
  "timestamp": 1767127033.4384139,
  "device": "typhoon",
  "id_mea": 13,
  "mea_value": Array (12) [
    0: 21.117103576660156,
    1: 22.158349990844727,
    2: 167.96240234375,
    ...
    11: 228.3522186279297
  ],
  "experiment_id": "experiment_3"
}
{
  "_id": "695437fceca9cc949c9ec03e",
  "timestamp": 1767127033.446452,
  "device": "typhoon",
  "id_mea": 1,
  "mea_value": Array (180) [
    0: -154.28805541992188,
    1: 21.586299896240234,
    ...
    179: 562.0804443359375
  ],
  "experiment_id": "experiment_3"
}
```

The experimental results confirm that the proposed architecture, deployed on a standard Raspberry Pi 4, successfully handles the dual requirement of high-frequency waveform reconstruction and long-term trend logging. The system maintained data integrity under the emulated traffic load, demonstrating that a decoupled microservices approach is a viable and cost-effective solution for the real-time monitoring of power conversion assets.

6. Conclusions and Future Works

This study presents a scalable, containerized monitoring architecture designed to address the rigidity and throughput limitations of traditional SCADA systems in power electronics. By employing a decoupled four-layer framework based on independent microservices, the proposed solution supports the high-frequency telemetry ingestion while maintaining system stability and data integrity.

The validation process was conducted in two complementary phases. First, synthetic characterization established the system's maximum throughput boundaries. Subsequently, and most importantly, the architecture was assessed within a representative environment using a Hardware-in-the-Loop (HiL) setup. In this environment, a Typhoon HIL606 simulator emulated a grid-tied Photovoltaic Power Plant driven by a standard mission profile, replicating the dynamics of an ANPC converter in real-time.

Experimental results in representative scenario demonstrated that the system, deployed on a resource-constrained Raspberry Pi 4, effectively processes heterogeneous traffic patterns—combining instantaneous waveforms for fault diagnosis and RMS metrics for steady-state supervision. This versatility is facilitated by the non-relational (NoSQL) nature of the storage layer, which enables the system to ingest and persist diverse data structures independent of rigid schema limitations. Furthermore, employing containerization was instrumental in this deployment, providing the modularity necessary to support horizontal scalability and efficient resource orchestration at the edge. The architecture maintained a PDR of 1.0 at ingestion rates of up to 100 msgs/s per node, demonstrating the feasibility of implementing reliable monitoring using cost-effective hardware. To extend this scalability to massive deployments, the architecture is designed to evolve towards container orchestration platforms such as Kubernetes, enabling automated scaling, self-healing capabilities, and the centralized lifecycle management of the microservices fleet across extensive industrial clusters.

Building upon the successful HiL validation, future research targets the deployment on physical ANPC converters to assess the architecture's robustness under real-world electromagnetic interference and thermal constraints. To bridge the gap towards industrial adoption, the software framework will be enhanced by integrating digital twin interfaces and RUL estimation algorithms directly into dedicated containers. This enables real-time synchronization between the physical asset and its virtual model, facilitating predictive maintenance strategies to minimize unplanned downtime. Furthermore, the system's scalability will be extended to a hybrid edge cloud topology aimed at the centralized supervision of distributed energy resources, such as PV inverters, battery energy storage systems, and EV charging stations, allowing for the unified observability of extensive device fleets. Finally, to ensure compatibility with brownfield infrastructure, future microservices will bridge internal telemetry with standard industrial protocols such as IEC 61850 or Modbus TCP, facilitating seamless integration with existing SCADA monitoring systems.

Author Contributions: Conceptualization, J.M.G.-C., A.M.A. and J.I.L.; software, J.M.G.-C. and A.L.-C.; formal analysis, J.M.G.-C., A.M.A. and J.I.L.; research, J.M.G.-C., A.L.-C., A.M.A. and R.P.; resources, J.M.G.-C. and A.M.A.; writing—original draft preparation, J.M.G.-C. and A.M.A.; writing—review and editing, J.M.G.-C., A.M.A., J.I.L. and R.P.; visualization, J.M.G.-C. and A.M.A.

supervision, J.M.G.-C., A.M.A. and J.I.L.; project administration; funding acquisition, J.M.G.-C. and A.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by Ministerio de Ciencia e Innovación under the project PID2023-152292OB-I00 as well as “UE—Ministerio de Hacienda y Función Pública—Fondos Europeos—Junta de Andalucía—Consejería de Universidad, Investigación e Innovación project 2024-31824-1828102303”, Shanghai Partner Research Program under Grant 25HB2705900.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Measurement Data Dictionary

Table A1 provides the complete mapping of variable identifiers (IDs) to physical quantities for the 3L ANPC converter implementation. These identifiers are utilized by the UDP serialization logic and the MongoDB document structure to categorize incoming telemetry.

Table A1. Supported measurement types for a 3L ANPC Converter operated in a grid connection application and a 1 kHz control frequency.

ID	Description
1	Instantaneous three-phase voltages in vector format, 20 components
2	Instantaneous three-phase currents in vector format, 20 components
3	Instantaneous values of DC, voltages of the 2 capacitors, and input current to the converter, 20 components
4	Instantaneous conduction losses in the devices, 20 components each
5	Instantaneous switching losses in the devices, 20 components each
6	Instantaneous junction temperatures of the devices, 20 components each
10	RMS value of three-phase voltages
11	RMS value of three-phase currents
12	RMS value of the converter DC, voltages of the 2 capacitors, and input current to the converter
13	RMS value of conduction power losses in the devices
14	RMS value of switching power losses in the devices
15	RMS value of junction temperatures in the devices

Appendix B. Architectural Deployment

The deployment automation, as detailed in Section 3.6, is driven by a centralized configuration file that defines the system topology and service parameters. Listing A1 presents the standard YAML template used as input for the orchestration script, with its key components detailed below:

- **network:** Defines the container network to which all containers (API, database, broker and UDP servers) are attached, enabling inter-container communication.
- **api:** Specifies the IP address for the RESTful API. Only one API container is created, independent of the number of monitored converters.
- **services:** Lists all power converters to be monitored. For each device:
 - **device:** Human-readable name of the converter.
 - **id:** Unique numerical identifier used by the UDP server to route messages to the corresponding broker exchange.

- port: UDP port for incoming data. Each converter has a dedicated UDP server container.
- ipbroker: IP of the central RabbitMQ broker.
- experiments: List of experiments for the device, creating dedicated queues in the broker and subscriber services.

Listing A1. YAML template for architectural deployment.

```
network:
  name: container_network
api:
  ip: "API_IP"

services:
  - device: deviceName1
    id: 1
    port: UDP_port_1
    ipbroker: "Broker_IP"
    experiments:
      - exp_1
      - exp_2
      - exp_N

  - device: deviceNameN
    id: 2
    port: UDP_port_2
    ipbroker: "Broker_IP"
    experiments:
      - exp_1
      - exp_2
      - exp_N
```

References

1. Blaabjerg, F.; Yang, Y.; Kim, K.A.; Rodríguez, J.R.R. Power Electronics Technology for Large-Scale Renewable Energy Generation. *Proc. IEEE* **2023**, *111*, 335–355. [\[CrossRef\]](#)
2. Hosseinabadi, F.; Chakraborty, S.; Bhoi, S.K.; Prochart, G.; Hrvanovic, D.; Hegazy, O. A Comprehensive Overview of Reliability Assessment Strategies and Testing of Power Electronics Converters. *IEEE Open J. Power Electron.* **2024**, *5*, 473–512. [\[CrossRef\]](#)
3. Peyghami, S.; Palensky, P.; Blaabjerg, F. An Overview on the Reliability of Modern Power Electronic Based Power Systems. *IEEE Open J. Power Electron.* **2020**, *1*, 34–50. [\[CrossRef\]](#)
4. Harikumar, J.; Buticchi, G.; Migliazza, G.; Madonna, V.; Giangrande, P.; Costabeber, A.; Wheeler, P.; Galea, M. Failure Modes and Reliability Oriented System Design for Aerospace Power Electronic Converters. *IEEE Open J. Ind. Electron. Soc.* **2021**, *2*, 53–64. [\[CrossRef\]](#)
5. Liserre, M.; Buticchi, G.; Leon, J.I.; Marquez Alcaide, A.; Raveendran, V.; Ko, Y.; Andresen, M.; Monopoli, V.G.; Franquelo, L. Power Routing: A New Paradigm for Maintenance Scheduling. *IEEE Ind. Electron. Mag.* **2020**, *14*, 33–45. [\[CrossRef\]](#)
6. Wang, H.; Blaabjerg, F. Power Electronics Reliability: State of the Art and Outlook. *IEEE J. Emerg. Sel. Top. Power Electron.* **2021**, *9*, 6476–6493. [\[CrossRef\]](#)
7. Zhang, G.; Zhou, D.; Yang, J.; Blaabjerg, F. Fundamental-frequency and load-varying thermal cycles effects on lifetime estimation of DFIG power converter. *Microelectron. Reliab.* **2017**, *76–77*, 549–555. [\[CrossRef\]](#)
8. Ma, K.; Liserre, M.; Blaabjerg, F.; Kerekes, T. Thermal Loading and Lifetime Estimation for Power Device Considering Mission Profiles in Wind Power Converter. *IEEE Trans. Power Electron.* **2015**, *30*, 590–602. [\[CrossRef\]](#)
9. Kardan, F.; Shekhar, A.; Bauer, P. Impact of Parameter Uncertainties on Power Electronic Device Lifetime Predictions. *IEEE Access* **2025**, *13*, 100479–100491. [\[CrossRef\]](#)
10. Reigosa, P.D.; Wang, H.; Yang, Y.; Blaabjerg, F. Prediction of Bond Wire Fatigue of IGBTs in a PV Inverter Under a Long-Term Operation. *IEEE Trans. Power Electron.* **2016**, *31*, 7171–7182. [\[CrossRef\]](#)
11. Cheng, G.; Lin, Y.; Abur, A.; Gómez-Expósito, A.; Wu, W. A Survey of Power System State Estimation Using Multiple Data Sources: PMUs, SCADA, AMI, and Beyond. *IEEE Trans. Smart Grid* **2024**, *15*, 1129–1151. [\[CrossRef\]](#)

12. Nomandela, S.; Mnguni, M.E.S.; Raji, A.K. Adaptive Control and Interoperability Frameworks for Wind Power Plant Integration: A Comprehensive Review of Strategies, Standards, and Real-Time Validation. *Appl. Sci.* **2025**, *15*, 12729. [\[CrossRef\]](#)
13. Sánchez, L.; Couso, I.; Costa, N. Self-supervised spectral learning with an application to condition monitoring of tunnel ventilation systems under sparse data conditions. *Comput. Ind. Eng.* **2026**, *211*, 111634. [\[CrossRef\]](#)
14. El Akhdar, A.; Baidada, C.; Kartit, A.; Hanine, M.; García, C.O.; Lara, R.G.; Ashraf, I. Exploring the Potential of Microservices in Internet of Things: A Systematic Review of Security and Prospects. *Sensors* **2024**, *24*, 6771. [\[CrossRef\]](#)
15. Mirani, A.A.; Velasco-Hernandez, G.; Awasthi, A.; Walsh, J. Key Challenges and Emerging Technologies in Industrial IoT Architectures: A Review. *Sensors* **2022**, *22*, 5836. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Sobin, C.C. A Survey on Architecture, Protocols and Challenges in IoT. *Wirel. Pers. Commun.* **2020**, *112*, 1383–1429. [\[CrossRef\]](#)
17. Ahmed, B.; Shabbir, H.; Naqvi, S.R.; Peng, L. Smart Agriculture: Current State, Opportunities, and Challenges. *IEEE Access* **2024**, *12*, 144456–144478. [\[CrossRef\]](#)
18. Bollineni, C.; Sharma, M.; Hazra, A.; Kumari, P.; Manipriya, S.; Tomar, A. IoT for Next-Generation Smart Healthcare: A Comprehensive Survey. *IEEE Internet Things J.* **2025**, *12*, 32616–32639. [\[CrossRef\]](#)
19. Awaisi, K.S.; Ye, Q.; Sampalli, S. A Survey of Industrial AIoT: Opportunities, Challenges, and Directions. *IEEE Access* **2024**, *12*, 96946–96996. [\[CrossRef\]](#)
20. Dui, H.; Zhang, S.; Liu, M.; Dong, X.; Bai, G. IoT-Enabled Real-Time Traffic Monitoring and Control Management for Intelligent Transportation Systems. *IEEE Internet Things J.* **2024**, *11*, 15842–15854. [\[CrossRef\]](#)
21. Shehu Yalli, J.; Hilmi Hasan, M.; Abubakar Badawi, A. Internet of Things (IoT): Origins, Embedded Technologies, Smart Applications, and Its Growth in the Last Decade. *IEEE Access* **2024**, *12*, 91357–91382. [\[CrossRef\]](#)
22. Mirani, A.A.; Awasthi, A.; O'Mahony, N.; Walsh, J. Industrial IoT-Based Energy Monitoring System: Using Data Processing at Edge. *IoT* **2024**, *5*, 608–633. [\[CrossRef\]](#)
23. Vargas-Salgado, C.; Aguila-Leon, J.; Chiñas-Palacios, C.; Hurtado-Perez, E. Low-cost web-based Supervisory Control and Data Acquisition system for a microgrid testbed: A case study in design and implementation for academic and research applications. *Heliyon* **2019**, *5*, e02474. [\[CrossRef\]](#)
24. de Arquer Fernández, P.; Fernández, M.Á.F.; Candás, J.L.C.; Arboleya, P.A. An IoT open source platform for photovoltaic plants supervision. *Int. J. Electr. Power Energy Syst.* **2021**, *125*, 106540. [\[CrossRef\]](#)
25. Eltamaly, A.M.; Alotaibi, M.A.; Alolah, A.I.; Ahmed, M.A. IoT-based hybrid renewable energy system for smart campus. *Sustainability* **2021**, *13*, 8555. [\[CrossRef\]](#)
26. Kalyanraj, D.; Prakash, S.L.; Sabareswar, S. Wind turbine monitoring and control systems using Internet of Things. In Proceedings of the 2016 21st Century Energy Needs—Materials, Systems and Applications (ICTFCEN), Kharagpur, India, 17–19 November 2016; pp. 1–4.
27. Zhao, L.; Zhou, Y.; Matsuo, I.B.M.; Korkua, S.K.; Lee, W.J. The Design of a Remote Online Holistic Monitoring System for a Wind Turbine. *IEEE Trans. Ind. Appl.* **2020**, *56*, 14–21. [\[CrossRef\]](#)
28. Naik, N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. In Proceedings of the 2017 IEEE International Systems Engineering Symposium (ISSE), Vienna, Austria, 11–13 October 2017; pp. 1–7. [\[CrossRef\]](#)
29. The MQTT Protocol. 2012. Available online: <http://www.mqtt.org> (accessed on 26 December 2025).
30. CoAP Protocol. 2025. Available online: <https://coap.space/> (accessed on 26 December 2025).
31. OASIS. *Advanced Message Queuing Protocol (AMQP)*, Version 1.0; OASIS: Burlington, MA, USA, 2012.
32. Santos, L.C.d.; da Silva, M.L.P.; dos Santos Filho, S.G. Sustainability in Industry 4.0: Edge Computing Microservices as a New Approach. *Sustainability* **2024**, *16*, 11052. [\[CrossRef\]](#)
33. Pallewatta, S.; Kostakos, V.; Buyya, R. Placement of Microservices-based IoT Applications in Fog Computing: A Taxonomy and Future Directions. *ACM Comput. Surv.* **2023**, *55*, 321. [\[CrossRef\]](#)
34. Yan, Y.; Qian, Y.; Sharif, H.; Tipper, D. A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 5–20. [\[CrossRef\]](#)
35. Wang, H.; Liserre, M.; Blaabjerg, F. Toward Reliable Power Electronics: Challenges, Design Tools, and Opportunities. *IEEE Ind. Electron. Mag.* **2013**, *7*, 17–26. [\[CrossRef\]](#)
36. Profanter, S.; Tekat, A.; Dorofeev, K.; Rickert, M.; Knoll, A. OPC UA versus ROS, DDS, and MQTT: Performance Evaluation of Industry 4.0 Protocols. In Proceedings of the 2019 IEEE International Conference on Industrial Technology (ICIT), Melbourne, VIC, Australia, 13–15 February 2019; pp. 955–962. [\[CrossRef\]](#)
37. Yang, S.; Xiang, D.; Bryant, A.T.; Mawby, P.A.; Ran, L.; Tavner, P.J. Condition monitoring for device reliability in power electronic converters: A review. *IEEE Trans. Power Electron.* **2010**, *25*, 2734–2752. [\[CrossRef\]](#)
38. Manohar, S.S.; Sahoo, A.; Subramaniam, A.; Panda, S.K. Condition monitoring of power electronic converters in power plants—A review. In Proceedings of the 2017 20th International Conference on Electrical Machines and Systems (ICEMS), Sydney, NSW, Australia, 11–14 August 2017; pp. 1–5. [\[CrossRef\]](#)

39. Wintrich, A.; Nicolai, U.; Tursky, W.; Reimann, T. *Application Manual Power Semiconductors*; SEMIKRON International GmbH: Nuremberg, Germany, 2011.
40. Avenas, Y.; Dupont, L.; Khatir, Z. Temperature Measurement of Power Semiconductor Devices by Thermo-Sensitive Electrical Parameters—A Review. *IEEE Trans. Power Electron.* **2012**, *27*, 3081–3092. [\[CrossRef\]](#)
41. Alosa, C.; Alcaide, A.M.; Stowhas-Villa, A.; Berger, J.G.; Immovilli, F.; Rojas, C.A.; Lizana F, R.; Buticchi, G.; Kouro, S.; Leon, J.I. Computational-Efficient IGBT and Diode Thermal Modelling Methodology with High Accuracy. *IET Power Electron.* **2025**, *18*, e70143. [\[CrossRef\]](#)
42. Wang, H.; Zhou, Z.; Xu, Z.; Ge, X.; Yang, Y.; Zhang, Y.; Yao, B.; Xie, D. A Thermal Network Model for Multichip Power Modules Enabling to Characterize the Thermal Coupling Effects. *IEEE Trans. Power Electron.* **2024**, *39*, 6225–6245. [\[CrossRef\]](#)
43. Yang, X.; Heng, K.; Dai, X.; Wu, X.; Liu, G. A Temperature-Dependent Cauer Model Simulation of IGBT Module with Analytical Thermal Impedance Characterization. *IEEE J. Emerg. Sel. Top. Power Electron.* **2022**, *10*, 3055–3065. [\[CrossRef\]](#)
44. Xu, Z.; Xu, F.; Wang, F. Junction Temperature Measurement of IGBTs Using Short-Circuit Current as a Temperature-Sensitive Electrical Parameter for Converter Prototype Evaluation. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3419–3429. [\[CrossRef\]](#)
45. Wang, H.; Wang, H. Analytical Modeling and Design of Capacitor Bank Considering Thermal Coupling Effect. *IEEE Trans. Power Electron.* **2021**, *36*, 2629–2640. [\[CrossRef\]](#)
46. Wang, H.; Huang, S.; Kumar, D.; Wang, Q.; Deng, X.; Zhu, G.; Wang, H. Lifetime Prediction of DC-Link Capacitors in Multiple Drives System Based on Simplified Analytical Modeling. *IEEE Trans. Power Electron.* **2021**, *36*, 844–860. [\[CrossRef\]](#)
47. García-Campos, J.M.; Alcaide, A.M.; Letrado-Castellanos, A.; Portillo, R.; Leon, J.I. Microservices Architecture for Condition Monitoring and SoH Tracking in Power Conversion Systems. GitHub Repository. 2026. Available online: <https://github.com/jmgarcam/microservices-pcs-soh> (accessed on 2 February 2026).
48. Gu, Y.; Grossman, R.L. UDT: UDP-based data transfer for high-speed wide area networks. *Comput. Netw.* **2007**, *51*, 1777–1799. [\[CrossRef\]](#)
49. Masirap, M.; Amaran, M.H.; Yussoff, Y.M.; Rahman, R.A.; Hashim, H. Evaluation of reliable UDP-based transport protocols for Internet of Things (IoT). In Proceedings of the 2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 30–31 May 2016; pp. 200–205. [\[CrossRef\]](#)
50. Guzmán-Miranda, H.; Marquez Alcaide, A. Lynx: An Efficient and Flexible Communication Schema for On-Chip and Off-Chip Applications in Distributed Static Industrial Networks. *IEEE Trans. Ind. Inform.* **2024**, *20*, 12542–12553. [\[CrossRef\]](#)
51. MongoDB Inc. *The MongoDB Database Manual*; MongoDB Inc.: New York, NY, USA, 2025.
52. Makris, A.; Tserpes, K.; Spiliopoulos, G.; Zissis, D.; Anagnostopoulos, D. MongoDB Vs PostgreSQL: A comparative study on performance aspects. *GeoInformatica* **2021**, *25*, 243–268. [\[CrossRef\]](#)
53. InfluxData Inc. *InfluxDB: Time Series Data Platform*; InfluxData Inc.: San Francisco, CA, USA, 2025.
54. Wang, J.; Zhang, W.; Shi, Y.; Duan, S.; Liu, J. Industrial big data analytics: Challenges, methodologies, and applications. *arXiv* **2018**, arXiv:1807.01016. [\[CrossRef\]](#)
55. Tripathi, P.; Miraz, M.H.; Joshi, S. Comparative Analysis of MongoDB and InfluxDB for Time Series Data Management in IoT Environments: A Study on Performance, Scalability, and Concurrency. In Proceedings of the 2023 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA), Zagreb, Croatia, 14–15 December 2023; pp. 39–42. [\[CrossRef\]](#)
56. Al-Sakran, A.; Qattous, H.; Hijawi, M. A proposed performance evaluation of NoSQL databases in the field of IoT. In Proceedings of the 2018 8th International Conference on Computer Science and Information Technology (CSIT), Amman, Jordan, 11–12 July 2018; pp. 32–37. [\[CrossRef\]](#)
57. Ramírez, S. FastAPI: High Performance Web Framework for Building APIs with Python. 2025. Available online: <https://fastapi.tiangolo.com/> (accessed on 28 December 2025).
58. Bednarsz, B.; Miłosz, M. Benchmarking the performance of Python web frameworks. *J. Comput. Sci. Inst.* **2025**, *36*, 336–341. [\[CrossRef\]](#)
59. Wang, M.; Sun, Y.; Sun, H.; Zhang, B. Security Issues on Industrial Internet of Things: Overview and Challenges. *Computers* **2023**, *12*, 256. [\[CrossRef\]](#)
60. Bagaskara, A.E.; Setyorini, S.; Wardana, A.A. Performance Analysis of Message Broker for Communication in Fog Computing. In Proceedings of the 2020 12th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 6–8 October 2020; pp. 98–103. [\[CrossRef\]](#)
61. Foundation Risk Partners. *The Raspberry PI Foundation*; Foundation Risk Partners: Ormond Beach, FL, USA, 2025.
62. Dobbelaere, P.; Esmaili, K.S. Kafka versus RabbitMQ. *arXiv* **2017**, arXiv:1709.00333. [\[CrossRef\]](#)

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