electronics

MDPI

Article

Architecture for Smart Buildings Based on Fuzzy Logic and the OpenFog Standard

Imanol Martín Toral [1,*], Isidro Calvo [1,*], Jani Xenakis [2], Eneko Artetxe [1] and Oscar Barambones [1]

[1] Department of Systems Engineering and Automatic Control, Faculty of Engineering of Vitoria-Gasteiz, University of the Basque Country (UPV/EHU), 01006 Vitoria-Gasteiz, Spain; eneko.artetxe@ehu.eus (E.A.); oscar.barambones@ehu.eus (O.B.)

[2] Department of Materials Engineering, Faculty of Engineering Technology, KU Leuven, Ghent Campus, 9000 Ghent, Belgium; jani.xenakis@kuleuven.be

* Correspondence: imanol.martint@ehu.eus (I.M.T.); isidro.calvo@ehu.eus (I.C.)

Abstract: The combination of Artificial Intelligence and IoT technologies, the so-called AIoT, is expected to contribute to the sustainability of public and private buildings, particularly in terms of energy management, indoor comfort, as well as in safety and security for the occupants. However, IoT systems deployed on modern buildings may generate big amounts of data that cannot be efficiently analyzed and stored in the Cloud. Fog computing has proven to be a suitable paradigm for distributing computing, storage control, and networking functions closer to the edge of the network along the Cloud-to-Things continuum, improving the efficiency of the IoT applications. Unfortunately, it can be complex to integrate all components to create interoperable AIoT applications. For this reason, it is necessary to introduce interoperable architectures, based on standard and universal frameworks, to distribute consistently the resources and the services of AIoT applications for smart buildings. Thus, the rationale for this study stems from the pressing need to introduce complex computing algorithms aimed at improving indoor comfort, safety, and environmental conditions while optimizing energy consumption in public and private buildings. This article proposes an open multilayer architecture aimed at smart buildings based on a standard framework, the OpenFog Reference Architecture (IEEE 1934–2018 standard). The proposed architecture was validated experimentally at the Faculty of Engineering of Vitoria-Gasteiz to improve indoor environmental quality using Fuzzy logic. Experimental results proved the viability and scalability of the proposed architecture.

Keywords: Internet of Things; IoT; edge/fog/cloud computing; artificial intelligence of things (AIoT); artificial intelligence (AI); control system; indoor environment quality (IEQ); OpenFog; fuzzy logic

check for updates

Citation: Martín Toral, I.; Calvo, I.; Xenakis, J.; Artetxe, E.; Barambones, O. Architecture for Smart Buildings Based on Fuzzy Logic and the OpenFog Standard. *Electronics* **2023**, *12*, 4889. https://doi.org/10.3390/electronics12244889

Academic Editors: Soochang Park and Hosung Park

Received: 10 November 2023
Revised: 29 November 2023
Accepted: 1 December 2023
Published: 5 December 2023

1. Introduction

IoT is expected to make a significant impact in building construction, operation, and management by facilitating high-class services, providing efficient functionalities, and moving towards sustainable development goals [1,2]. These works identified several opportunities for IoT in smart buildings, especially for the purposes of managing the energy, enhancing the indoor comfort, and improving safety and security. As discussed in [3], the introduction of IoT and wireless sensor networks in Indoor Air Quality (IAQ) monitoring systems may help to achieve adequate environmental conditions as well as a correct ventilation of the buildings while improving comfort and health for the occupants.

Several Artificial Intelligence techniques have been used for improving the operation of IAQ systems. As a matter of example, in [4], artificial neural networks (ANN) were used for modeling the indoor air carbon dioxide concentration. This work proposes an ANN-based model aimed at predicting indoor CO_2 concentration using temperature and relative humidity as inputs. Also, in [5], it is presented a novel synthetical index for monitoring the air quality by means of a controller that integrated ANN and genetic algorithms.

Other authors have used algorithms based on fuzzy logic for classifying environmental conditions [6] or designing environment IAQ control systems [7]. In [8], an algorithm based on fuzzy logic is presented for the automation and management of indoor temperature, humidity, and air quality. Other authors propose systems, based on fuzzy logic, aimed at reducing the risk of COVID-19 transmission [9]. Another study suggests soft computing techniques such as ANN, SVM, fuzzy logic, deep learning models, and other hybrid models for critical IAQ monitoring [10].

Modern IoT applications involve a big number of IoT devices requiring high processing capabilities, which are frequently provided via cloud services. However, cloud computing applications face numerous difficulties in terms of performance, security, latency, and network breakdown. The use of IoT architectures may provide the skeleton to ease the integration of diverse IoT devices with the cloud. Cloud computing is frequently used for centralized data processing and storage. However, according to various studies [11,12], the use of cloud services may introduce latency and security problems as well as integration or management issues. This affects the performance, quality, and reliability of these services.

For this reason, the Edge/Fog/Cloud computing paradigms were introduced [13]. The Fog paradigm locates some processing of IoT applications at the cloud's edge. Fog computing uses a hierarchy of fog nodes placed near the physical plants. Fog nodes provide high quality services to sensor and actuator devices with short response times, which may reach a few milliseconds, using inexpensive local networking facilities. Also, the fog layer improves the security of critical applications, reducing hacker threats. Thus, the fog becomes an intermediate layer between IoT devices (sensors and actuators) and the cloud data centers, providing a superior control system that moves most of the decision-making functions closer to the plants, only contacting the cloud occasionally to report status or receive commands [14]. Several features of fog computing, as well as common architectures and algorithms, are reviewed in [15]. The fog computing paradigm is expected to help researchers to push artificial intelligence techniques to the edge.

Some authors have proposed fog solutions as applied to smart buildings [16]. In [17], it is presented an intelligent decision system which provides an efficient management of residential applications based on IoT devices that measure presence, humidity, temperature, and luminosity. Also, in [18], it is presented a scalable IoT architecture for monitoring the Indoor Environmental Quality parameters in public buildings. It uses low-cost devices for measuring temperature, humidity, eCO_2, and TVOCs.

In this scenario, it is necessary to introduce interoperable architectures aimed at corporative buildings which ease the introduction of advanced services in IAQ systems. These services, increasingly based on artificial intelligence techniques, are aimed at monitoring the IAQ conditions in real time and introduce sophisticated analysis of the environmental conditions. In addition, new services must be developed to control the environmental conditions of the buildings. These services are expected to improve the comfort and safety for the occupants while optimizing the energy consumption. However, IAQ systems require connecting heterogeneous devices. For this reason, it is necessary to provide architectures, preferably based on standards, in order to promote interoperability among devices, especially as the number of IoT devices increases. Although research on this topic is still incipient, there are some existing fog computing frameworks for IoT applications. One of the most known initiatives is OpenFog Reference Architecture (RA), promoted by the OpenFog Consortium and adopted as IEEE 1934–2018 standard [19], as well as by the Industy IoT Consortium [20], supported by the OMG. OpenFog RA outlines eight pillars for Fog applications: Security; Scalability; Open; Autonomy; Programmability; RAS (Reliability, Availability, and Serviceability); Agility; and Hierarchy. It also incorporates a glossary for fog computing terms. OpenFog RA defines Fog Computing as a system-level flat framework that divides storage, resources, computing services, and networking from every place, along with the range from Cloud to Things [19].

The OpenFog RA standard identifies several domains of special applicability, with smart buildings being one of them. These applications may contain thousands of sensors to measure various parameters, such as temperature, humidity, occupancy, air quality, or the state of doors/windows (open or close). Sensors may capture these data and transmit it to local storage servers in the fog for further analysis purpose. In addition, fog services may implement control algorithms to adjust indoor conditions if necessary, ensuring short response times. However, although smart buildings have been identified as a possible application domain for OpenFog RA, it is unlikely to find examples in the literature describing how to implement OpenFog on smart buildings.

This work proposes a multi-layer architecture based on the OpenFog architecture aimed at corporative buildings. The proposed architecture tries to guide developers to design new applications that may include complex algorithms based on AI techniques. The presented architecture implements the Edge, Fog, and Cloud layers by means of diverse types of devices. In particular, the Fog layer is responsible for collecting the data from the Edge nodes and executing the control algorithms for ensuring adequate indoor environmental conditions. Fuzzy logic was used for this task, since it is an intuitive technique that has been proven in monitoring IAQ systems. The architecture also allows for the integration of advanced services available as cloud services. The proposed architecture was validated experimentally by means of a prototype operating at the Faculty of Engineering of Vitoria-Gasteiz. The prototype included several IoT nodes at the edge, designed ad hoc to measure environmental parameters of the building and operating actuators. For validation purposes, a Fog node was responsible for providing several services, including (1) storing the data acquired from the IoT devices; (2) executing a control algorithm to improve the environmental air quality, based on Fuzzy logic techniques; and (3) connect to cloud services for knowing outdoors conditions and weather forecast. The prototype collected the information acquired via several IoT nodes deployed in a room, measuring IEQ parameters, namely temperature, humidity, luminosity, and CO_2 concentration, and operating a mechanical window by means of a Fuzzy logic algorithm that also considered outside information obtained from cloud services. This algorithmic model is commonly used for its straightforward implementation and acceptable results [7]. Experimental results proved the validity of the presented approach, paving the way to introduce complex algorithms, based on AI techniques, which may successfully combine fog and edge services.

The layout of the article is as follows: Section 2 summarizes OpenFog RA and analyzes some selected works in the subject. Section 3 introduces the generic architecture, aimed at smart buildings, based on OpenFog RA. Section 4 describes the experimental prototype used for validation purposes and presents the experimental results obtained. Finally, Section 5 draws some conclusions.

2. State of the Art

2.1. OpenFog Consortium: Standardizing Fog Architecture for AIoT

The OpenFog Consortium, established in 2015 with founding members such as ARM, Cisco, Dell, Intel, Microsoft, and Princeton University, aims to standardize the implementation of fog and edge computing technologies. Their approach focuses on enabling efficient communication between Fog–Fog and Fog–Cloud Tiers. They highlight the benefits of this approach, including improved Security, Cognitive capabilities, Agility, reduced Latency, and enhanced Efficiency, which they collectively refer using the SCALE acronym. The IEEE 1934–2018 OpenFog standard emerged as a result of this initiative.

The OpenFog Consortium collaborates with various IoT and technology industry alliance groups, including the Industrial Internet Consortium (IIC), ETSI-MEC, OPC-UA, Open Connectivity Foundation (OCF), OpenNFV, and others. This collaboration is aimed at avoiding the duplication of efforts and market confusion while working on optimizing specific application spaces. Thanks to this effort, the possibility for companies and factories of different sizes to implement AIoT architectures increases [21]. Also, the standardization of fog services via OpenFog RA represents a solution to issues such as the lack of a generalized and flexible reconfigurable framework, as mentioned in [22].

Its long-term vision is to foster greater convergence in the IoT industry by creating a shared perspective on edge and fog architectures, especially in areas that have not been adequately addressed via existing initiatives. This has led to interoperability issues between IoT devices, resulting in critical problems such as dependence on a single vendor and difficulty in developing applications that work across multiple platforms, hindering the widespread adoption of IoT technology [23]. The OpenFog Consortium's mission aligns with the goal of promoting standardized solutions that embrace IoT, AIoT, and paradigms such as fog and edge computing for enterprises, developers, and stakeholders [19].

OpenFog is based on eight main pillars that define the architecture. These pillars represent the key attributes that must be satisfied to embody the OpenFog definition of a horizontal, system-level architecture that provides the distribution of computing, storage, control, and networking functions closer to the data source along the cloud-to-thing continuum [19]. These pillars and their objectives are as shown in Figure 1.

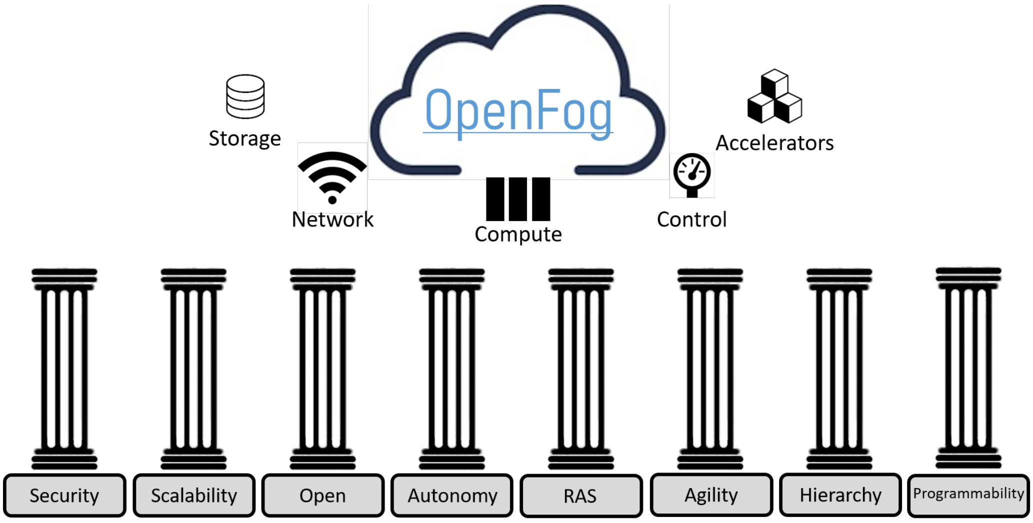


Figure 1. Pillars of OpenFog.

1. **Security**: OpenFog RA prioritizes security, offering adaptable measures for privacy, integrity, and trust. Compliance ensures end-to-end security, with hardware-based roots of trust;
2. **Scalability**: The architecture scales internally, within networks, and in a demand-driven elastic environment. It adapts to varying fog application needs, resizing resources as necessary;
3. **Openness**: Openness encourages diversity and innovation in the fog ecosystem via interoperability, versatile fog nodes, and location transparency;
4. **Autonomy**: OpenFog's autonomy pillar reduces reliance on centralized cloud control, enabling context-aware decisions and efficient data transmission, following the DIKW model. AI implementations like in Muneeb (2021) are emerging;
5. **RAS (Reliability, Availability, Serviceability)**: The RAS pillar ensures uninterrupted functionality, covering reliable hardware operation, fault detection, redundancy, and automation;
6. **Agility**: The agility pillar fosters data-driven IoT decisions and dynamic fog deployments, reducing network dependencies for optimized application placement;
7. **Hierarchy**: OpenFog's hierarchical computing resources ensure scalability and flexibility for IoT needs. Fog nodes operate autonomously, enabling uninterrupted management. Figure 2 shows four types of hierarchy proposed by OpenFog, but there can be more like Figure 4 in [24] where it shows a hierarchy with five tiers;
8. **Programmability**: Programmability in OpenFog RA allows for automated function reassignment, offering flexibility, resource efficiency, multi-tenancy support, cost-effective operations, and enhanced security [19].

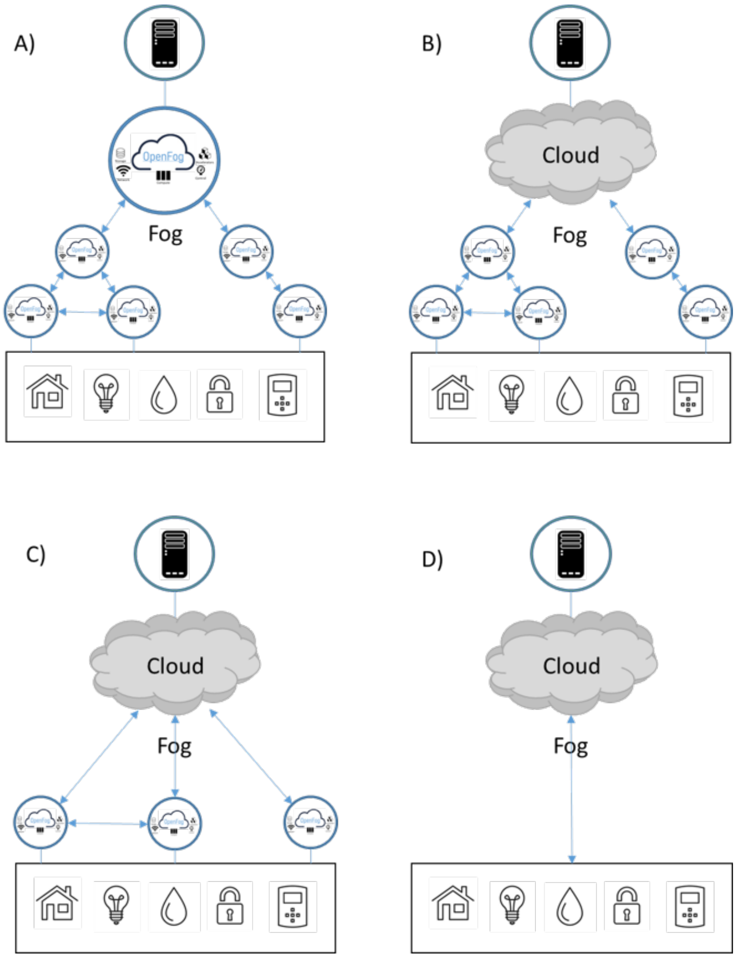


Figure 2. Types of hierarchy according to IEEE 1934 OpenFog standard [19].

The OpenFog framework is established on fundamental principles. It defines an architecture, which serves as a crucial reference for projects, enabling low latency and reliable operation, while reducing the need for constant connectivity to the cloud. However, as noted in [25], OpenFog emphasizes the migration of applications through the fog implementation tiers, but lacks specific implementation details. Some of these applications include the communication framework that is applied to security, control, resource management, and risk prevention systems, such as monitoring energy use and expenditure in household appliances [26]. It is also used to analyze environmental conditions to improve wellbeing, prevent machinery or infrastructure failures [27], and assess adverse weather conditions or natural disasters that may directly affect the building [28].

OpenFog shows the software and hardware structure on which the above concepts are supported. They can be defined in three concepts:

- **Software View:** This perspective is depicted in the top three layers of the architectural description, covering Application Services and Support, Node Management In Band (IB), and the Software Backplane;
- **System View:** This perspective is represented in the intermediate layers of the architectural description, spanning from Hardware Virtualization to Hardware Platform Infrastructure;
- **Node View:** This perspective is represented in the two lower layers and includes the Protocol Abstraction Layer and Sensors, Actuators, and Control.

2.2. Related Work

The implementation of fog services via the application of the standard is an expanding field, and therefore, as mentioned in this study [29], many articles are case studies or prototypes. That is why they provide a review that identifies and describes five fog computing models that are compatible with the adoption of OpenFog according to the IEEE 1934 standard. A model compatible with this standard can be [30], which focuses on enhancing data analysis in IoT using a multi-layered architecture involving cloud and fog computing in real time. Another project that is compatible with the standard targets fog computing solutions for IoT in industrial environments, prioritizing low latency and predictability and aligning with fog computing principles [31]. Other studies about the application of OpenFog are [32,33].

Other examples like this [34] propose a multi-layered architecture based on the OpenFog architecture. Also, it models and simulates fog computing environments and shows the benefits in terms of average latency and energy consumption. In [35], the authors explore how performance and management influence the architecture to enhance application orchestration in Kubernetes environments. In [36], the authors propose a unified architecture and taxonomy for the comparison of proposals, with the aim of meeting the requirements of IoT applications and guiding future research in this field. A universal cloud-edge orchestration platform aligned with OpenFog RA for advanced orchestration is proposed in this study [37].

OpenFog is not the only existing model for building an automatic control architecture for intelligent buildings. One of the first protocols created to introduce concepts such as those advocated in this article comes from KNX, founded in 1999 by the associations EIBA, EHSA, and BCI. Its aim is the development and promotion of an international communications standard for Home and Building Automation. For example, in [38,39], the implementation of KNX in smart homes is being developed. In addition to domestic use, KNX is used in many areas of industry for the control of plants, but also for corporate use, such as the control of environmental conditions in an office, as in [40]. KNX is known as a low-cost open technology. Although it is commonly used, it does not focus especially on Wireless connection. Although, there are examples in the last decade of the integration of IoT technologies or web services [41,42], which is something that OpenFog has contemplated from the beginning, giving solutions to problems of flexibility and scalability to the structure.

Another group seeking to standardize architectures that integrate services such as IIoT and AI is EdgeX, which is a solution provider vendor. They use end-to-end structures based on edge-cloud architecture, one of the most commercially available to date. Another standard to mention is Industrial Internet Reference Architecture (IIRA). It was created by members of the Industry Internet Consortium (IIC) and other members in 2019. Its motivation is to create a fast, efficient, and autonomous wireless edge computing architecture that facilitates the control and operability in the industrial sector. In Ref. [21], IIRA is compared with other similar standards. The review demonstrates its good performance in Industry 4.0 as being one of the most widely used and versatile standards. Ultimately, the Open Edge Computing Initiative drives the global convergence of edge computing platforms, providing compelling applications, operating the Living Edge Lab as a test centre, and promoting adoption with vendors and operators.

It should be noted that the aforementioned methods and protocols are based on edge computing, which provides highly marketable structures, favoring above all their application in the industrial area due to their versatility. However, the OpenFog consortium mentions the need to evolve towards a structure that integrates a fog layer to avoid the security and latency problems that can occur in the cloud. It also improves real-time responsiveness, as the analysis and control processes would be closer to the edge.

3. Generic Architecture

Following the OpenFog Reference Architecture (RA) guidelines, this article proposes an AIoT architecture to be deployed in smart buildings. This architecture must achieve the terms under the SCALE acronym (Security, Cognition, Agility, Latency, and Efficiency) in smart building applications, involving enhancing indoor air quality, optimizing energy efficiency, ensuring occupant safety, and enabling smart building automation. This approach enables its continuous integration into any type of corporative building, whether public or private.

The underlying premise of this architecture is its versatility in addressing various service areas. The first step in designing an architecture is considering the scope of its application. In the case of smart buildings, the objectives to consider are

- Implementation of measure and actuation IoT devices in selected areas;
- Acquisition and processing of information using local control devices tailored to different zones of the building;
- Integration of complex cloud services for general building control and analysis;
- Hierarchical control of different buildings or areas being subordinated to a supervision central control device.

The essential pillars that are indispensable for the architecture represent guiding principles that shape the framework and underlying infrastructure, which play a vital role in achieving the architectural design objectives. They include

- **Security**: Ensuring the integrity of systems in smart buildings and protecting user privacy. This includes mechanisms to ensure data integrity, device authentication, protection against cyber threats, and the implementation of security in device communications at all levels of the architecture;
- **Scalability**: Allowing the architecture to adapt to buildings of different sizes. Also, it must let the system to grow and evolve with the changing needs of the building. This requires a flexible software infrastructure that supports software management, updates, and service orchestration;
- **Openness**: Embracing open-source and free software technologies (FOSS) to encourage interoperability and collaboration. This avoids dependence on proprietary solutions and limitations to diversity and innovation;
- **Autonomy**: Granting fog nodes the ability for autonomous building management, minimizing the need for human intervention. This involves integrating Artificial Intelligence techniques that enable the system to learn from data and make intelligent decisions;
- **RAS (Reliability, Availability, and Service)**: Ensuring uninterrupted functionality in both normal and adverse conditions. This entails selecting and implementing robust and reliable hardware for sensors, actuators, and devices, as well as ensuring that fog nodes can assume the responsibilities of failed nodes. Also, proper communication technologies must be used to integrate the services that form the applications;
- **Hierarchy**: Partitioning the architecture into different layers, each with a unique role. This allows for rapid deployment and reconfiguration of services, adapting to changing user needs and environmental conditions;
- **Latency**: Minimizing delays in data transmission and processing, ensuring real-time communication and control. This is achieved by distributing the workload among nodes and performing data analysis at the edge or fog level.

3.1. Architecture

The architecture designed for application in intelligent buildings is shown in Figure 3. Despite the availability of other alternatives, as shown in Figure 2, choosing a multi-fog architecture, Figure 2B, along with the cloud is the most suitable option for targeting smart buildings and achieving the objectives described above. The foundation of this structure consists of devices in the Edge Tier, such as sensors, actuators, or mobile devices. The local control devices or Fog nodes correspond to the fog Tier being responsible for processing and analyzing information in the areas where they have been deployed, as well as hosting the algorithms that act as the decision-making authority at this level. In this case, there are two types of local control: one for specific areas, providing services for controlling and analyzing devices in the Edge Tier, and one for the entire building, overseeing the area controls. The number of tiers of a fog architecture depends on the requirements the system may need (e.g.the quantity of sensors, the amount of work, the capacities of nodes at every tier, the inter-node and sensor-to-actuation latency, and the reliability and availability of nodes). The general controllers will also play the role of bridges between the cloud and the area fog controllers to transmit information. These fog nodes could assume a variety of responsibilities, including

- Performing emergency monitoring and response functions;
- Overseeing building security functions;
- Managing climate and lighting;
- The general fog nodes take control of the sensor nodes if, for any reason, the area fog nodes are lost.

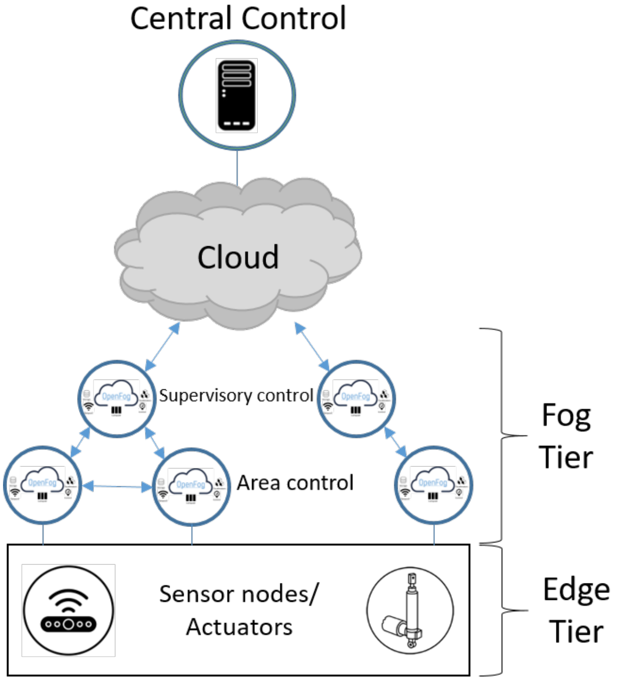


Figure 3. Generic architecture end-to-end example.

The information collected via the cloud will be sent to the central control, which will be in charge of managing all the structures that belong to the architecture.

The following shows the perspectives with which the construction of a generic OpenFog RA for smart buildings is managed.

3.1.1. Node View for Smart Buildings' Architecture

The architecture introduces two different types of nodes to the system: Edge Nodes and Fog Nodes. These nodes form the core elements of each tier and represent the essential components that drive the framework and underlying infrastructure as shown in Figure 4. Their unique characteristics and functionalities play a fundamental role in shaping the architecture and achieving its design goals. Although there is also the concept of a cloud node, this type of node fulfills a function more related to the application of a service provided via Internet being analyzed in Section 3.1.3.

Common elements exist for both types of nodes, such as

- **Storage:** This element provides the necessary capacity to store substantial data related to smart building operations;
- **Compute:** Each node is equipped to process data efficiently, optimizing performance and reducing computational costs;
- **Servers:** These components support the broader system, enhancing data processing and communication capabilities.

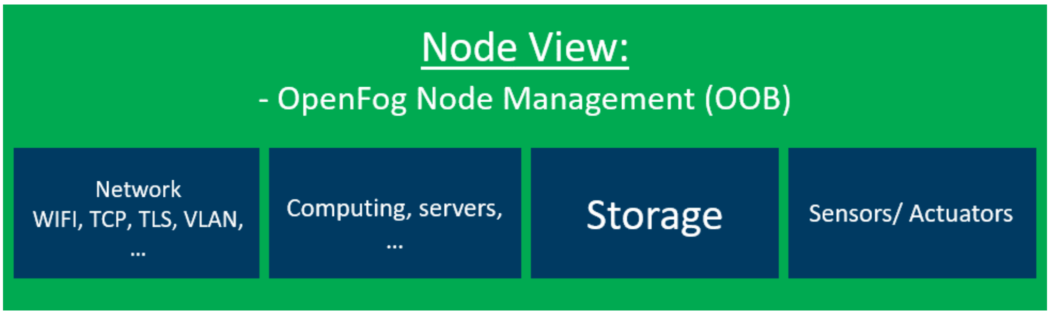


Figure 4. Node view for smart buildings. The idea of this figure is inspired by [19].

A description of the two node types, along with the distinguishing elements, is as follows:

- **Edge Node:** These nodes comprise sensors and actuators serving as fundamental devices in the Edge Tier. They play a central role in the system by collecting data related to building operations. These data are subsequently transmitted to fog devices, which act as intermediaries for relaying the information to the next architectural level. Edge devices encompass various options, including security sensors, safety sensors, and actuators as can be seen in Table 1.

Table 1. Examples of Edge Devices.

Device/Sensor Category	Illustrative Cases
Security sensors and dispositives	- Security cameras - Motion sensors - Access control systems - Intruder detectors - Fire detection systems
Safety sensors	- Carbon monoxide (CO) detectors - Smoke sensors - Air quality sensors - Temperature and humidity sensors - Water level sensors
Actuators	- Electronic locks - Automatic fire extinguishing systems - Motorized blinds - Intelligent Climate Control Systems - Automated lighting systems

- **Fog Node:** They include Fog Node Management Out of Band (OOB), which specifically handles control management within the system based in a human–machine interaction. It plays a pivotal role in ensuring scalability and reliability. This type of node facilitates efficient firmware and software upgrades for both sensor/actuator nodes and local Fog devices, enabling remote designation for updates and system maintenance. In cases of node disconnection, a recovery procedure can be employed to restore connectivity.

3.1.2. System View for Smart Buildings' Architecture

The system view analyzes the dynamic interactions and relationships among all architecture devices, providing a deep understanding of how they collaborate to achieve smart building objectives. It is the layer that manages the nodes within the system (see Figure 5).

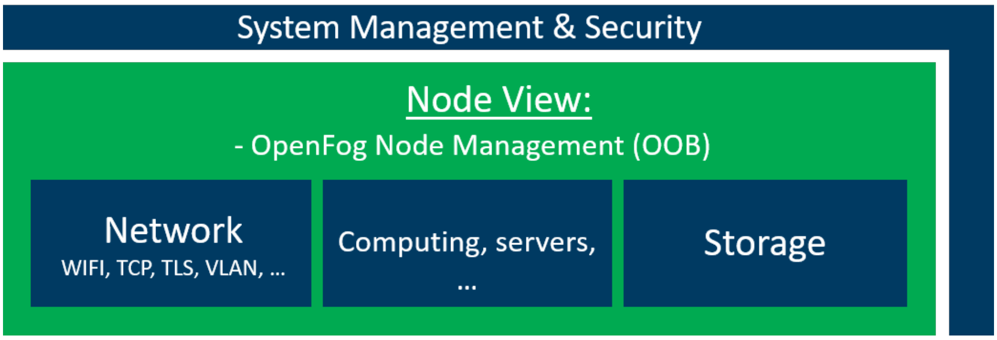

System Management & Security
Node View:
- OpenFog Node Management (OOB)
Network
WIFI, TCP, TLS, VLAN, ...
Computing, servers, ...
Storage

Figure 5. System view for smart buildings. The idea of this figure is inspired by [19].

- **System management:** At the system level, management includes end-to-end operations, from installation to repair, all autonomously executed by the system itself. This aligns with the OpenFog Reference Architecture (RA), streamlining procedures and enabling dynamic element assignments based on real-time needs. The system management allows for both human–machine and machine–machine communication when necessary, especially in scenarios requiring human intervention. It includes the management of specialized services like In Band (IB) within a holistic framework. Hierarchy is crucial, with higher-level hardware overseeing the system and lower-level nodes supervised by higher-level counterparts. The system ensures continuity by reassigning low-level nodes if a supervisor fails. Figure 6 illustrates an example of this.

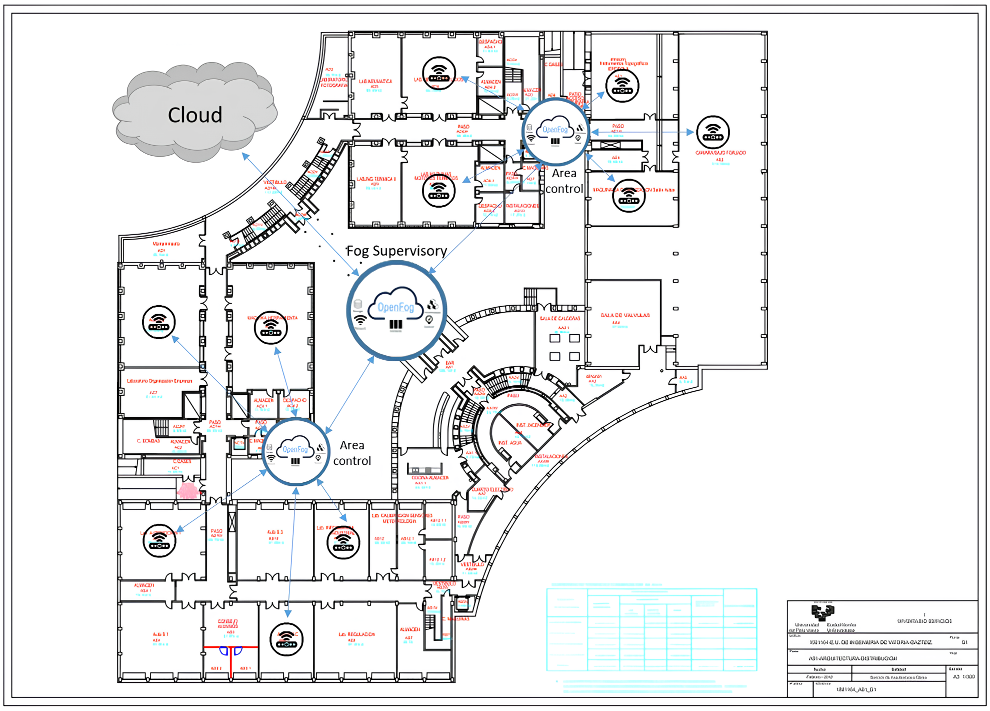


Figure 6. Simplified example of node distribution.

- **System security:** To safeguard the architecture from attacks, all nodes across levels must collaborate for autonomous defense. Higher-level nodes, like the fog nodes, manage security at access points and networks, ensuring lower-level nodes' safety. For this purpose, work must be performed on enhancing the security mechanisms of the devices. It is proposed that both low-level and high-level fog nodes should be adequately shielded against remote and hardware interactions. To achieve this, Single Board Computers (SBC) like Raspberry Pi, especially for the lower-tier nodes, are well suited. However, they can also be effectively utilized in higher-tier nodes. Additionally, Personal Computers (PC) are suitable for this purpose. These SBC and PCs offer the advantage of configuring user accounts and passwords, setting up firewalls, and installing antivirus software. Furthermore, both SBC and PC possess greater capacity for network filtering to prevent attacks or network misuse while ensuring data encryption.

When it comes to edge nodes, the primary focus of their protection should be at the hardware level, as it is the most vulnerable component. While safeguarding them remotely via the use of user names, passwords, and certificates is essential, equivalent measures must also be implemented at the hardware level. This dual approach aims to prevent unauthorized access to the software.

- **Network of the system:** The system's network has been designed with a specific focus on smart buildings. In the case of both public and private buildings, especially those within a corporate domain such as industry, corporate networks are utilized. These corporate networks, unlike typical domestic networks, are equipped with enhanced security measures and advanced features.

  Communication links between nodes and the network can be established via either wired (e.g., Ethernet) or wireless connections, with the choice being dependent on specific working conditions. When opting for a wireless connection in this architectural context, two key groups come into consideration: WLAN (Wireless Local Area Network) and WPAN (Wireless Personal Area Network). Protocols like Z-Wave or ZigBee can be viable options for network implementation. Additionally, the use of networks such as WiFi (a form of WLAN) is a common choice due to its affordability and widespread use in corporate networks.

3.1.3. Software View for Smart Buildings Architecture

Software View describes the software that runs on the nodes and provides the fog services. These services range from resource management and security to environmental optimization and data collection. The software component is organized in two layers, as shown in Figure 7: the first layer being Application Services and the second layer being Node Management and Software Backplane. The Node Management and Software Backplane layer not only manages the services, but also provides them and is used by the system to operate the rest of the architecture efficiently.

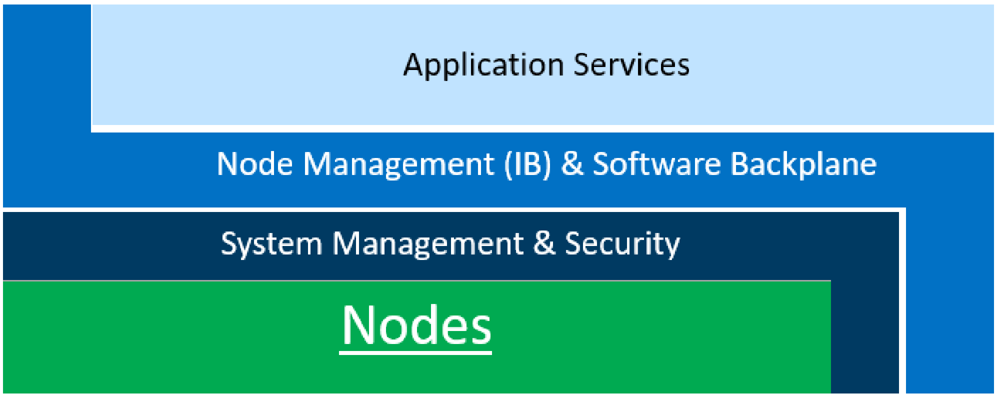

Application Services
Node Management (IB) & Software Backplane
System Management & Security
Nodes

Figure 7. Software view for smart buildings. The idea of this figure is inspired by [19].

In the context of the architecture, two key layers can be identified, each with its own functionality:

- **Node Management and Software Backplane**: At the software level, it is responsible for providing the platform necessary to run any software on the node and to facilitate node-to-node communications. In smart buildings, these components are essential to ensure that each node operates according to its desired state, guaranteeing the necessary availability, resilience, and performance (e.g., security management protects nodes and data via keys, cryptography, and policies; capacity management adjusts resources according to demand; and availability management ensures failover, maintaining sufficient capacity to meet Service Level Agreements (SLA) in smart buildings).

- **Application Services**: application services comprise a suite of tightly integrated micro-services that collectively form the backbone of fog computing applications. The proposed architectural framework relies upon several pivotal services to ensure the efficient operation of the system. Within this structure, fog connector services are tasked with interfacing with IoT devices and sensors, facilitating data translation into the required format. Simultaneously, core services are responsible for the aggregation of data from edge devices, making it accessible to higher-level applications and systems, including cloud platforms or cloud nodes. Another feature of the control services offered supports the ability to redirect the services managed by one fog node to another in case of a problem. Figure 8 shows a schematic of how the application is laid out.

  In these high tiers, cloud services are required to expand the scope of services offered by the architecture. Cloud environments can perform long-term and heavy resources operations, that can be associated to Big Data but easily extendable to any IoT applications [43]. Also, it can provide processing, networking, and storage capabilities closer to users [44,45]. However, cloud computing faces IoT challenges in real-time responsiveness, security, privacy, and energy efficiency, which current transport technologies struggle to address [46]. In the lower-level sections of the fog nodes, supporting services handle routine application tasks such as service registration and data maintenance. Supervisor fog nodes host analytics services, introducing AI capabilities into the architecture. These services provide both reactive and predictive functionalities, encompassing critical event analysis, anomaly detection, machine learning for forecasting, and more. While conventional data analysis remains an option, the integration of AI services is becoming increasingly prevalent due to their autonomy and expedited decision making. Integration services, on the other hand, have a significant impact on the scalability of the architecture by allowing for external fog nodes to register for specific data. Finally, user interface Services focus on the presentation of data, service status, analysis results, and system management operations.

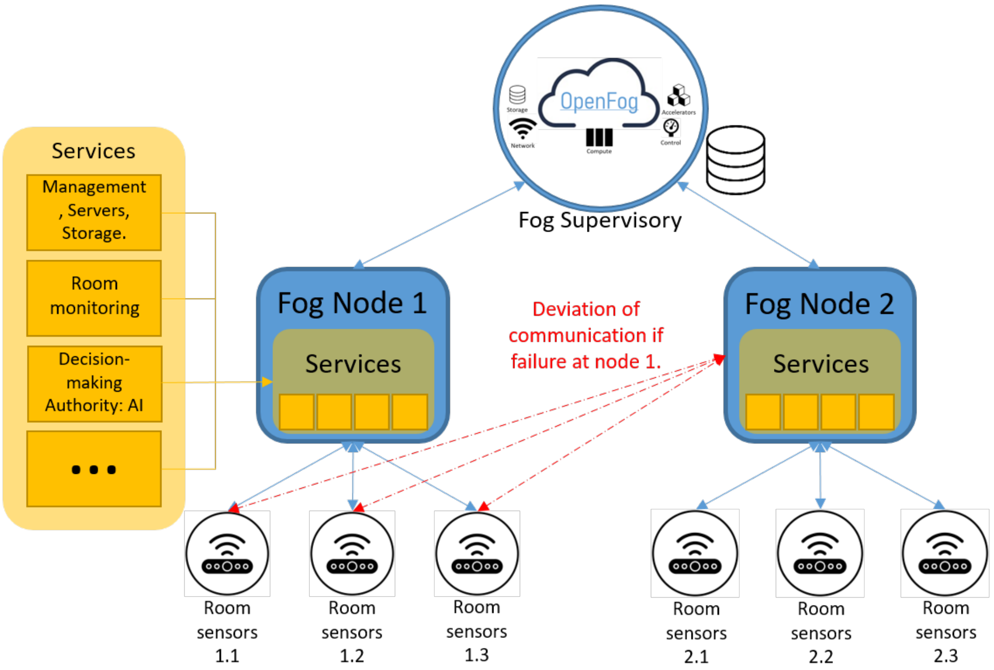


Figure 8. Generic deployment software view.

4. Case Study

To substantiate the potential advantages of implementing the proposed AIoT architecture based on OpenFog IEEE 1934 for smart buildings, this article includes an experimental prototype demonstrating the practical application of this structure in a real-world scenario.

An IEQ system was designed for validation purposes. This system is aimed at improving the indoor environment quality and higher energy efficiency in selected areas of the Faculty of Engineering of Vitoria-Gasteiz. A prototype of this system is currently in operation.

To solve some of the problems previously mentioned, the following ideas have been put forward:

- Design and implementation of smart sensors aimed at measuring several variables such as CO_2, temperature, and humidity, as well as window actuators;
- Creation of a fog communication service that collects the parameters measured via the Edge devices and sends the actuation signals by means of the MQTT protocol;
- Implementation of a fog service for deciding whether the windows should be open or not to ensure the air quality of the room. The algorithm used Fuzzy logic to infer a decision on the basis of the measured parameters;
- Coordinate the fog node with the cloud to access temperature and humidity data outside the building;
- Create Fog services, programmed in Python language, to analyze how the parameters evolve over time.

4.1. Prototype Architecture

The prototype shown in Figure 9 represents a simplified version of the view in Figure 3. For the case study, the fog supervisory node and the area fog have come together, since the system is only applied for the control of one room. Therefore, the edge tier is kept the same while only one node is available in the fog, which represents the remote control of the room which is connected to the cloud.

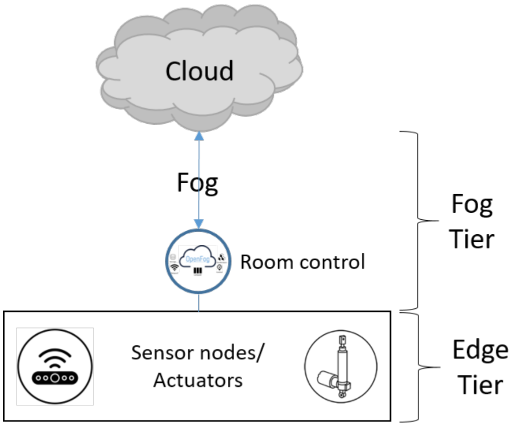


Figure 9. Prototype Architecture.

4.1.1. Experimental Nodes

Figures 10 and 11 portrays the experimental room used for the IEQ control prototype. The system structure consists of two groups of sensor nodes distributed throughout the room where the IEQ conditions are analyzed. These nodes communicate with a fog node via an already established IP network. The fog node is located in another room, from where the data provided by the edge nodes are monitored, analyzed, and processed.

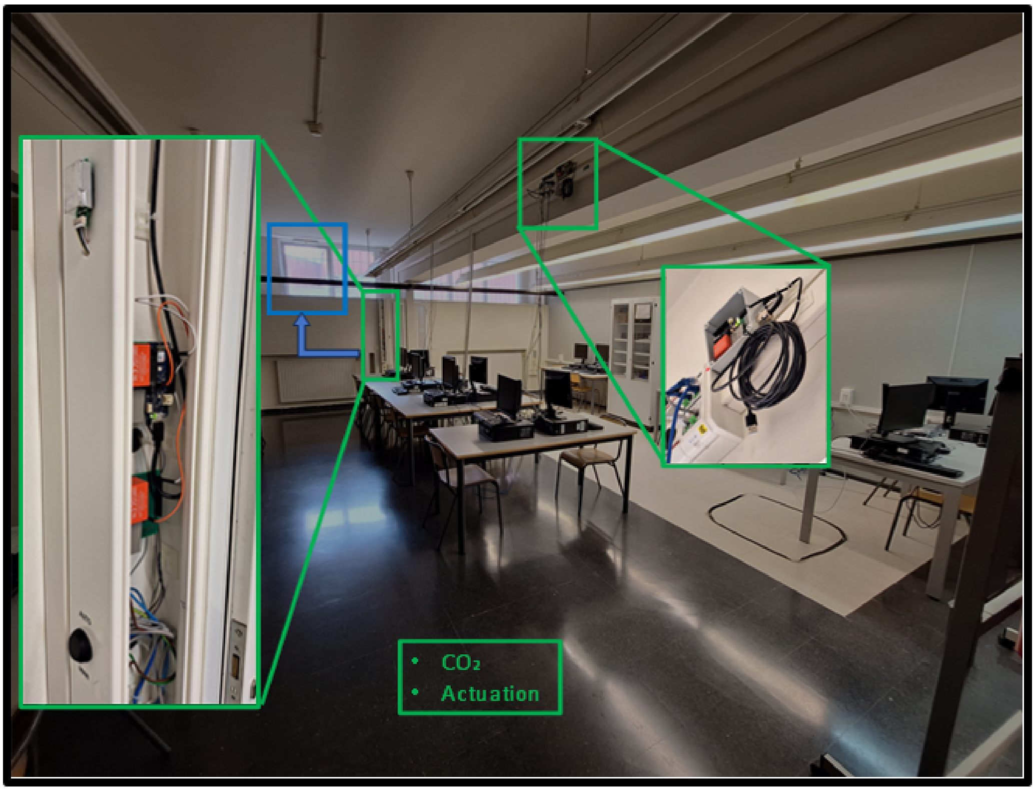


Figure 10. Distribution of the sensor nodes. CO_2 and actuators nodes.

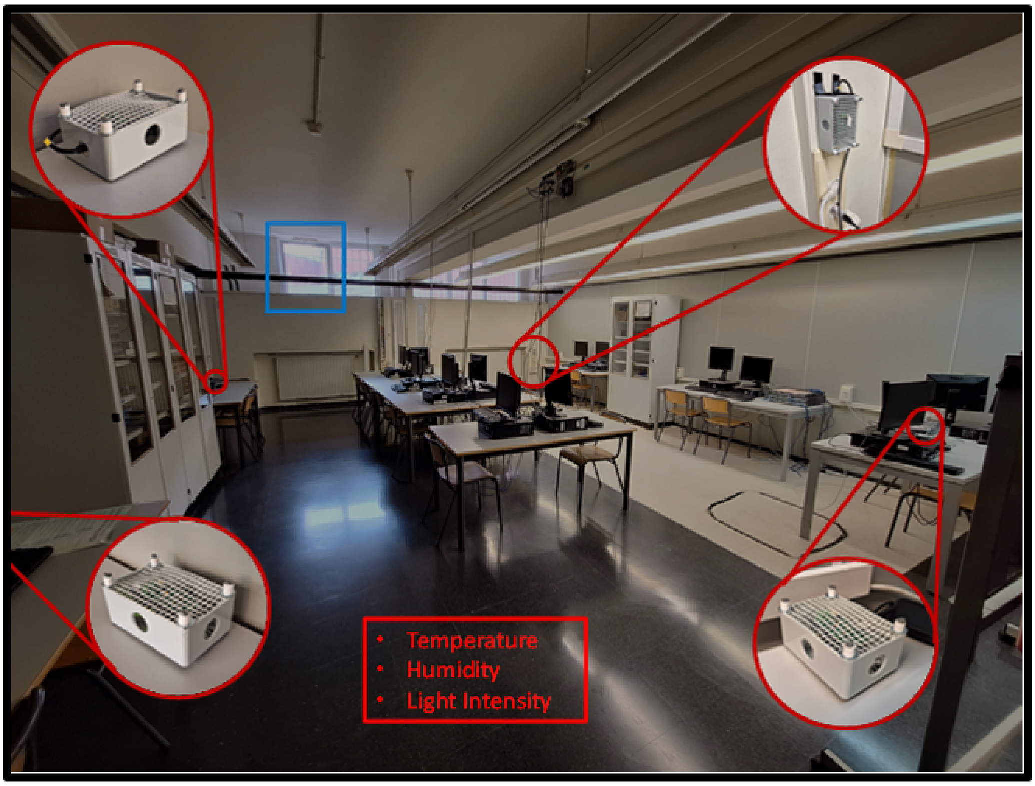


Figure 11. Distribution of the sensor nodes. Temperature, humidity, and light intensity nodes.

The fog and edge nodes designed for this prototype differ from each other because of the different criteria used to perform their functions. However, they share the same computational perspective, as both types of nodes are designed to perform their processes in the most efficient way. To illustrate this, the node designs are shown below:

- **Fog Node:** In this case, the area IEQ control fog node is hosted on a desktop computer running Windows. A Python application was designed to acquire the IoT data and execute the control algorithm. This node communicates with the edge nodes and hosts the AI decision-making authority, based on fuzzy logic. This node not only processes the information, but also stores it in a register. The fog node can also request responses from the edge nodes in case one of them becomes disconnected, a preliminary work tested on an OpenFog OBB mechanism that allowed for the dynamic connection of the IoT devices [18].

  The PCBs used in this prototype is expected to be replaced by a Single Board Computer, such as a Raspberry Pi, since this is a more economical and simple option and provides enough computing capabilities for executing the algorithms.
- **Edge Node:** The edge nodes, also called sensor nodes, are at the core of the IEQ architecture. They acquire the environmental data of the room and feature the control of the window actuator. The vast market of components facilitates the possibility of implementing simple, efficient, and functional low-cost hardware. In the case of the sensor nodes, the microcontroller boards that have been used are Arduinos MKR 1010, which stands out for its versatility. These nodes offer flexibility, low consumption, and also has the ability to connect to WiFi, which allows it to be used for IoT applications [47].

  The models designed for the respective groups use the same Arduino board. However, two different Printed Circuit Boards (PCBs) types were designed, with different integrated sensors. The characteristics of the two PCBs are as follows:
  - **Group A:** This PCB was used in nodes N2, N3, N4, and N5. All these nodes take measurements of the same IEQ parameters (e.g., temperature, humidity, and light intensity). For this, the sensors that have been integrated are BME680, KPS-3227-SP1C. Figure 12 shows the PCB of the group A model with its main components;
  - **Group B:** This PCB was used in nodes N0 and N1. Both nodes take CO_2 measurements using the MH-Z19B sensor. The N0 node is responsible for controlling the opening of the window. Figure 12 shows the PCB of this model with the main components.

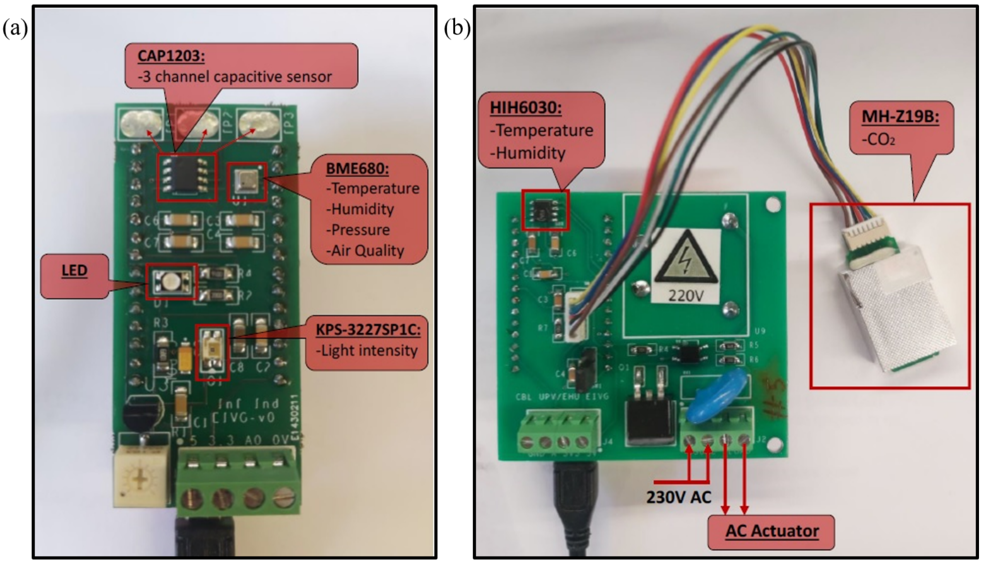


Figure 12. (**a**) Temperature, humidity, and ligth sensor node. (**b**) CO_2 and actuator control node.

4.1.2. System Overview

Despite its architectural simplicity, this is an end-to-end structure capable of future evolution by adding the tiers and services illustrated in Figure 3. The system ensures a continuous flow of data from sensor nodes (the edge nodes) to the fog node or supervisor node for processing. In case any sensor node disconnects or fails to send information on time, the system employs a protocol to re-establish communication, thus preventing data loss from the environment. If these attempts are unsuccessful, the system sends a notification indicating which node is experiencing issues. This process serves as an example of how the system manages the IB service.

Security is an important element for this type of architecture and, although it has not been implemented in this case, previous studies and experiments have been carried out to analyze the possibilities of using these technologies. To ensure system security, the primary focus has been on the nodes, as they constitute the fundamental structures of the system:

1. **Edge nodes:** As mentioned earlier, all edge or sensor nodes utilize Arduino MKR 1010 boards, which support the incorporation of security features. While the connection between the edge node and the fog node can be secured with a username, password, and certificates, there remains a potential vulnerability if someone gains physical access to the microcontroller and manipulates the code. To address this concern, MKR 1010 Arduinos come equipped with an integrated ATECC508A cryptochip. This cryptochip offers secure storage for up to 16 keys, certificates, various data types (read/write, read-only, or secret), consumption logging, and security configurations. Access to specific memory sections can be restricted in various ways and configurations can be locked to prevent unauthorized changes.

   Currently, encryption of the information transmitted over the network has been successfully achieved by programming certificates, usernames, and passwords in an ESP32-S2-Saola-1V1.2 boards, which is similar to the MKR. Future adoption of this model is not excluded.
2. **Fog node:** In the prototype, the Fog node is implemented over PC running Windows, connected to the UPV/EHU coorporative IP network. This node is configured according to the University rules, including an antivirus and a firewall configuration. Access to this node is restricted with passwords to avoid access by non-allowed users.

The system takes advantage of the University of the Basque Country's (UPV/EHU) corporate network, structured as a Virtual Local Area Network (VLAN). In simple terms, a VLAN is like a digital network segmentation that allows administrators to divide a single physical network into multiple logical networks. Each segment groups devices based on their function or permissions, making it easier to organize devices and isolate them into groups. This provides the system with the ability to operate similarly to a conventional physical LAN, but with the advantage of logical network grouping. In summary, a VLAN creates digital segments that enable devices to communicate as if they were on the same local network, even if they are physically located in different places. This digital division ensures isolation and enhances network security by restricting access only to authorized personnel within their respective segments in UPV/EHU's larger corporate network.

To establish communication between the Edge and Fog Tiers, the MQTT protocol is employed. This protocol is highly versatile and performs well in real-time communication, making it one of the most popular choices for machine-to-machine (M2M) communications [48,49]. In Figure 13 is presented the prototype connection architecture used. Most MQTT implementations introduce some security mechanisms that may reduce hacker attacks, if properly configured.

The data transport operation is straightforward. The control program of the fog node, hosting the MQTT broker to which all sensor nodes connect, sends commands to the nodes periodically, in the range of several minutes, via topics. The "take" topic is the command to request the data collected by all connected sensors at a specific instant. These will be sent in JSON format. Upon receiving the command, the nodes will send back the parameter values by means of the topic "values/n#", where "n#" identifies a specific node. If any node has disconnected and reconnected, the control device reissues the "take" topic to prevent data loss. If a node fails to reconnect, the control program reports the missing node and only registers those currently connected to the network.

Communication with cloud services, to collect external temperature and humidity, is carried out by means of the HTTPS protocol.

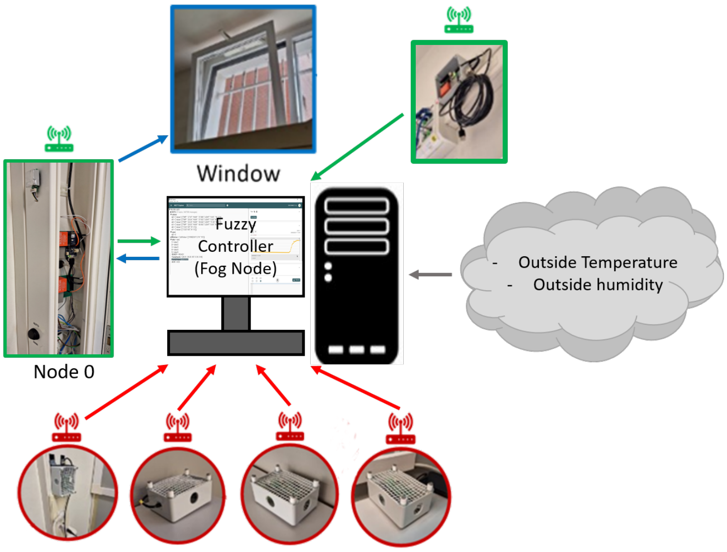


Figure 13. Prototype connection architecture. In red: node group A; in green: node group B; and in blue: window plus actuator.

4.2. Fog Services

The software responsible for executing data communication, collection, analysis, processing, and storage in the fog node was developed using the Python programming language. It handles all these functions and plays a crucial role in managing sensor node failures. In the event of disconnection, it attempts to reestablish contact with the nodes, and if unsuccessful, it issues an alert to notify of the problem. This functionality aligns with what was earlier referred to as "IB management". Additionally, the software is in charge of sending data collection and window control commands within their specified timeframes.

In Figure 14, it illustrates several aspects of the architecture. Firstly, the arrival of data is monitored following the transmission of the "take" topic message by the concentrator. The system provides notifications about the receipt of data from the edge nodes. The data is then displayed in a format translated from JSON. Once the data retrieval is complete, data processing begins. The information is processed via the controller algorithm, based on Fuzzy logic, to make decisions on the basis of the values obtained, as can be seen in Figure 15.

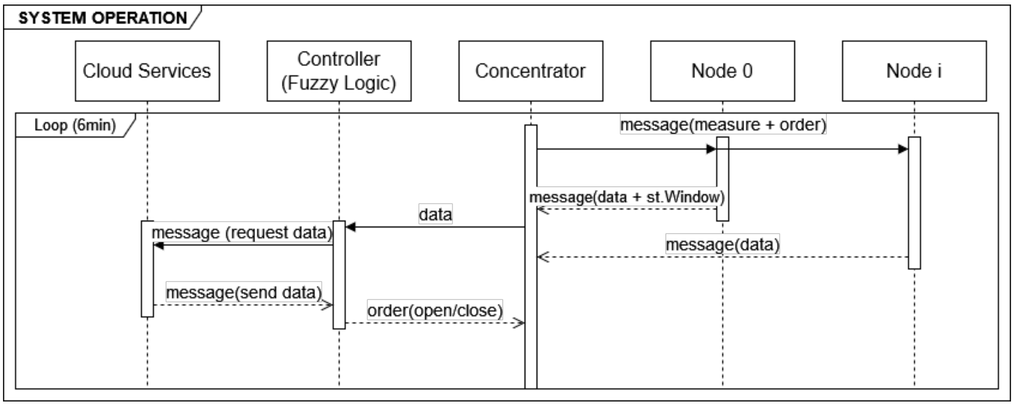


Figure 14. System operation flowchart.

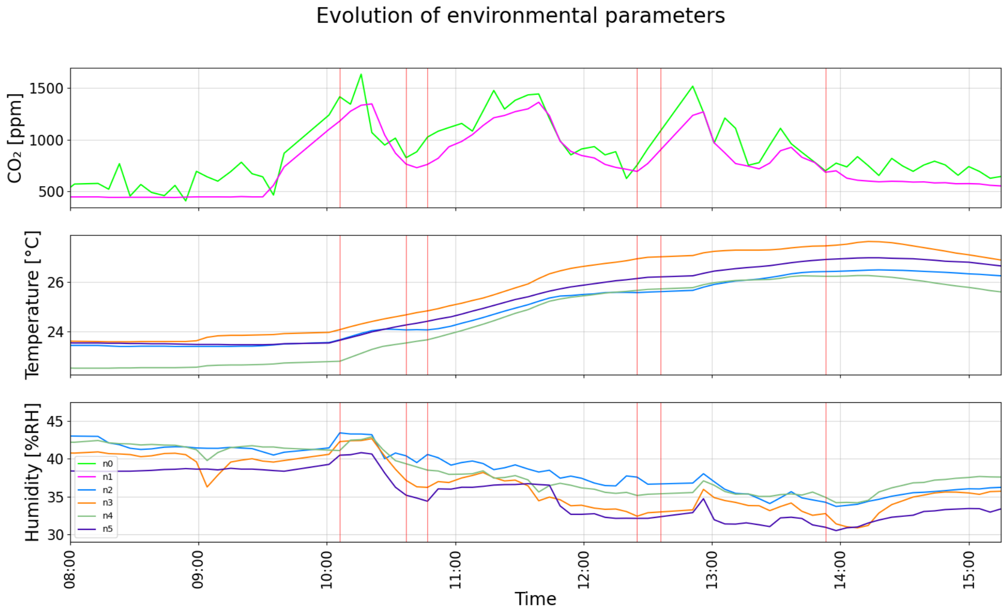


Figure 15. This graph is a representation of the operation of the control system. The red lines represent the times when the actuator has been operated.

In order to obtain real-time data for the decision-making authority to manipulate the window, cloud-hosted services have been used. These services are designed to offer widespread, user-friendly access to a shared pool of adaptable computing resources [50], as in this case, to access a website where humidity and temperature information from outside is saved. These data help to support monitoring and control services using data stored in the cloud, without the necessity of having a local store and system to take those values.

One of the primary services within the architecture is AI-driven decision making using fuzzy logic. To emulate human perception, a predefined set of categories is established. Depending on the input values, the algorithm infers decisions that align with the current values of the measured parameters. The choice of Fuzzy logic over other algorithms was due to its ease of use, good reliability, and ease for users to check what is happening in the calculations. Also, Fuzzy logic algorithms allows for easily combining the values of several input parameters.

In Figure 14, the control cycle is repeated every six minutes, using the data acquired from the IoT edge devices as well as the values obtained from the cloud services. When the Fuzzy Logic algorithm determines that the window should be open or closed, the control service, executed at the fog node, sends a message to command node 0, which opens or closes the window, depending on the value of the environmental conditions.

Table 2 summarizes the rules implemented in the Fuzzy logic control algorithm. Figure 16 shows the Mamdani method for generating the membership functions by which the fuzzy logics will be driven.

Table 2. Fuzzy rules to operate the classroom window.

Rules for the Fuzzy Logic:
IF temperature is cold AND outside temperature is warmer THEN window is open
IF humidity is wet AND outside humidity is lower THEN window is open
IF humidity is dry AND outside humidity is higher THEN window is open
IF temperature is hot and outside temperature is lower THEN window is open
IF CO_2 is high THEN window is open
IF temperature is comfortable AND CO_2 is good THEN window is closed
IF temperature is hot AND outside temperature is higher THEN window is closed
IF temperature is cold AND outside temperature is lower THEN window is closed
IF humidity is dry AND outside humidity is lower THEN window is closed
IF humidity is wet AND outside humidity is higher THEN window is closed

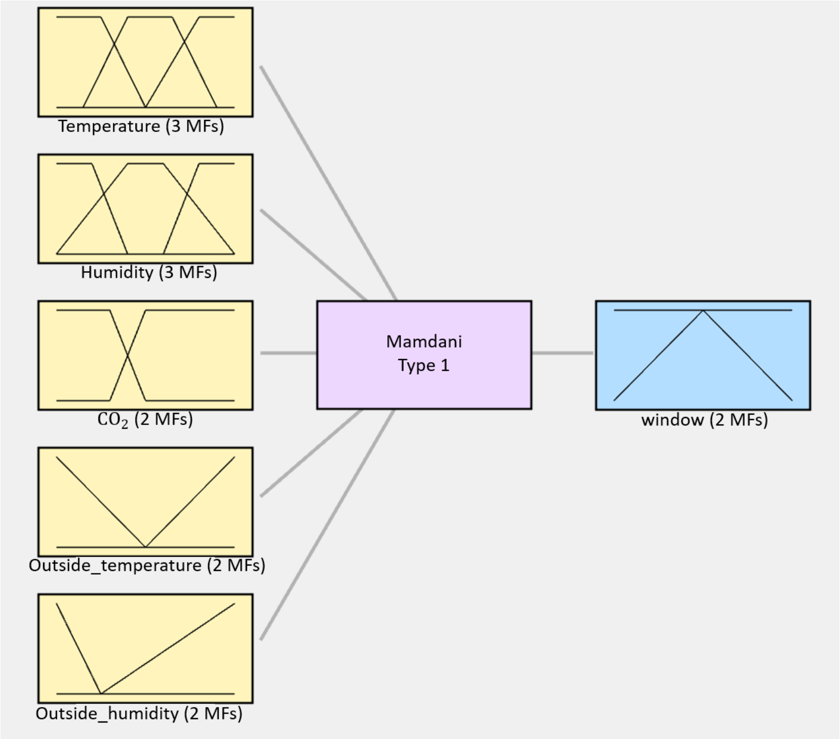


Figure 16. Representation of the general Fuzzy Logic System used for this project.

4.3. Experimental Measurements

The experimental process was divided into two phases. The initial phase involved testing the performance of the measurement devices in an unoccupied room. The second phase involved the analysis of indoor environmental quality (IEQ) during an academic term.

During the initial phase, experiments were conducted without the presence of individuals to record sensor measurements. This enabled the team to observe variations among the sensors. Figure 17 illustrates these temperature differences, with the largest observed variance being 1 °C. This variance aligns with the BME680 sensor datasheet [51], which specifies an accuracy of approximately ±5 °C.

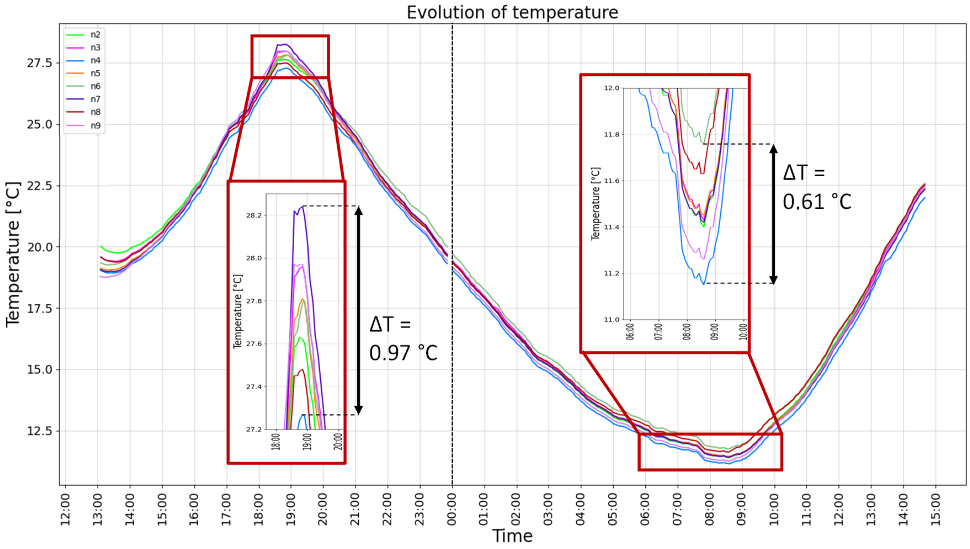


Figure 17. Evolution of measured temperatures by different sensors at the same location.

Similar procedures were applied to the humidity sensors, resulting in variances of approximately ±6%RH. This variance is slightly higher than the ±3%RH accuracy indicated in the datasheet [51].

Temperature and humidity values should be properly calibrated using precise and expensive sensors. As shown in Figures 17 and 18, they show the evolution of the measured parameters, taken at different locations of the room. It can be appreciated that all sensors present the same trends for every physical magnitude.

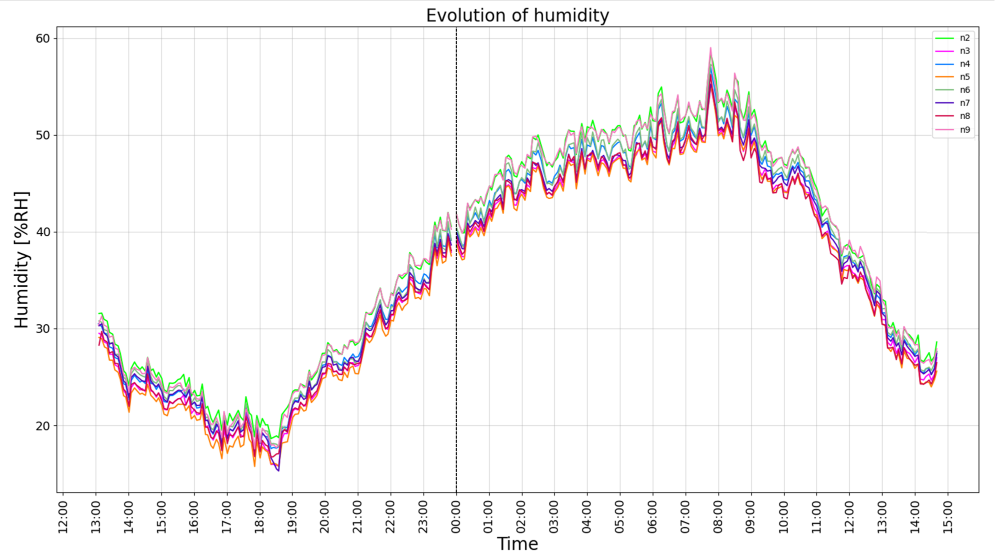


Figure 18. Evolution of measured humidity by different sensors at the same location.

The second experimental phase involves testing the prototype in a more realistic scenario. The goal of this second phase was to observe the system's response to environmental changes in a classroom during class sessions. In this experiment, the IoT system collected the IEQ parameters in a laboratory session taken from 9:30 to 13:30. There were approximately 15 people inside the laboratory. Figures 19–21 resulted in noticeable changes in IEQ, as evidenced by the collected data. The graph shown in Figure 19 shows a temperature increase while the laboratory session goes on. Starting at 9:30, when the laboratory session begins, it rises until it finishes at 13:30, where the temperature stops increasing. The temperature decreases when the laboratory session finishes.

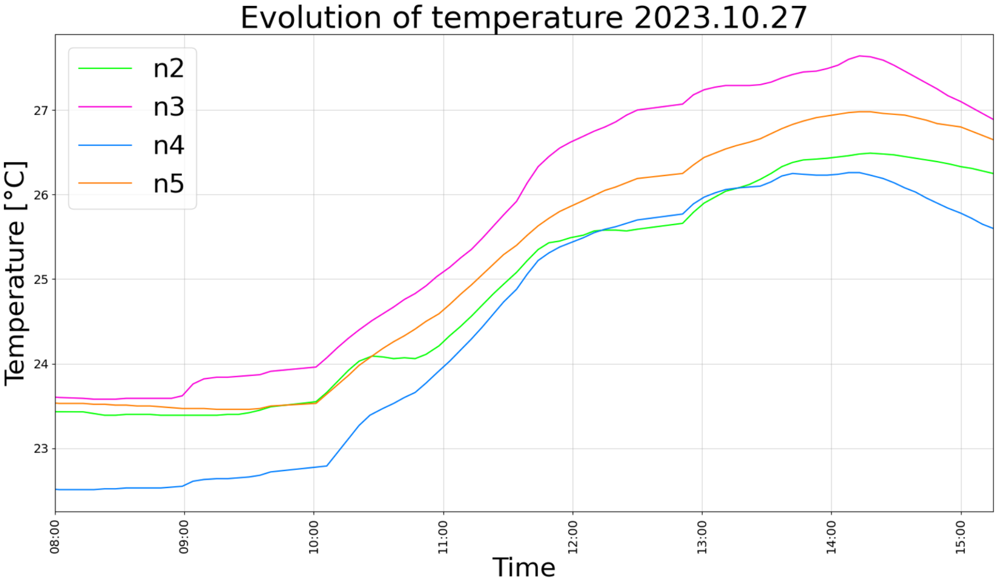


Figure 19. Evolution of temperature while the classroom is occupied.

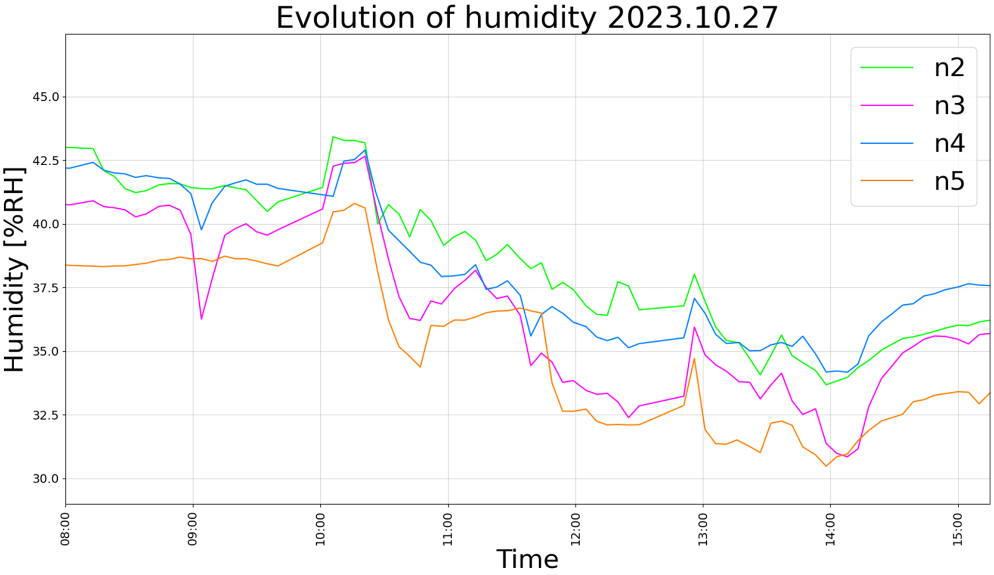


Figure 20. Evolution of humidity while the classroom is occupied.

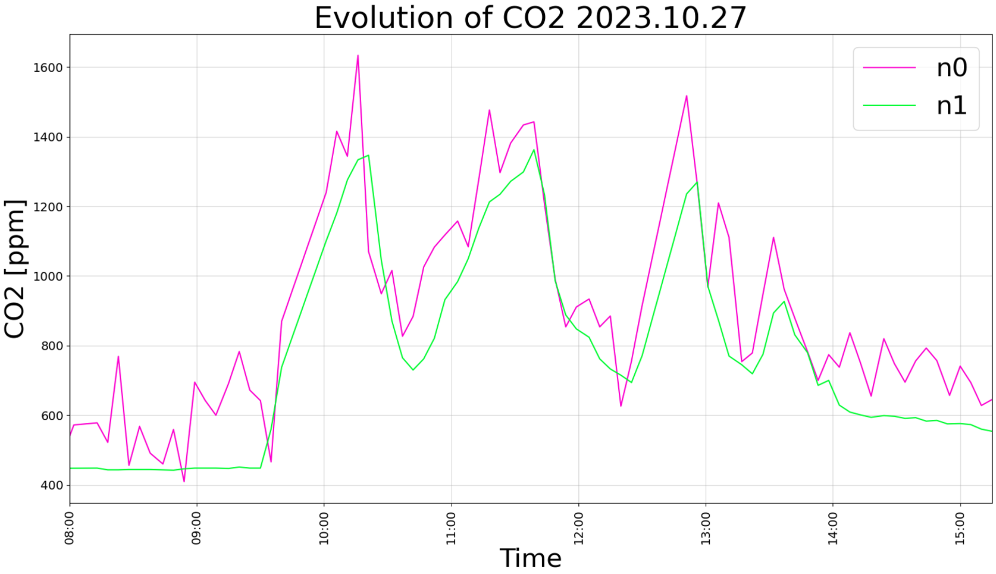


Figure 21. Evolution of CO_2 while the classroom is occupied.

On the humidity graph, it can be observed both more distinct variations and a progressive decline over time, as shown in Figure 20. In contrast to the temperature, humidity values will rise again once the class is over.

The graph of CO_2 in Figure 21 provides more information about what is happening in the room. The highest CO_2 peaks are recorded at 10:20, 11:40, 12:50, and 13:30. After reaching these peaks, the Fuzzy algorithm decides to open the window, and consequently, the level of CO_2 decreases rapidly due to the ventilation of the room. As expected, opening the window helps to reduce the CO_2 concentration, see Figure 14.

In Figure 22, Fuzzy Logic graphs are portrayed, where the system makes the decision to open the window. This specific moment is the execution of the window opening at 10:20, when the Fuzzy Logic value exceeded the threshold set at a value of 0.5. The high CO_2 concentration was mainly responsible for this. This causes the red line in the “Window” subplot to shift to the “open window” position.

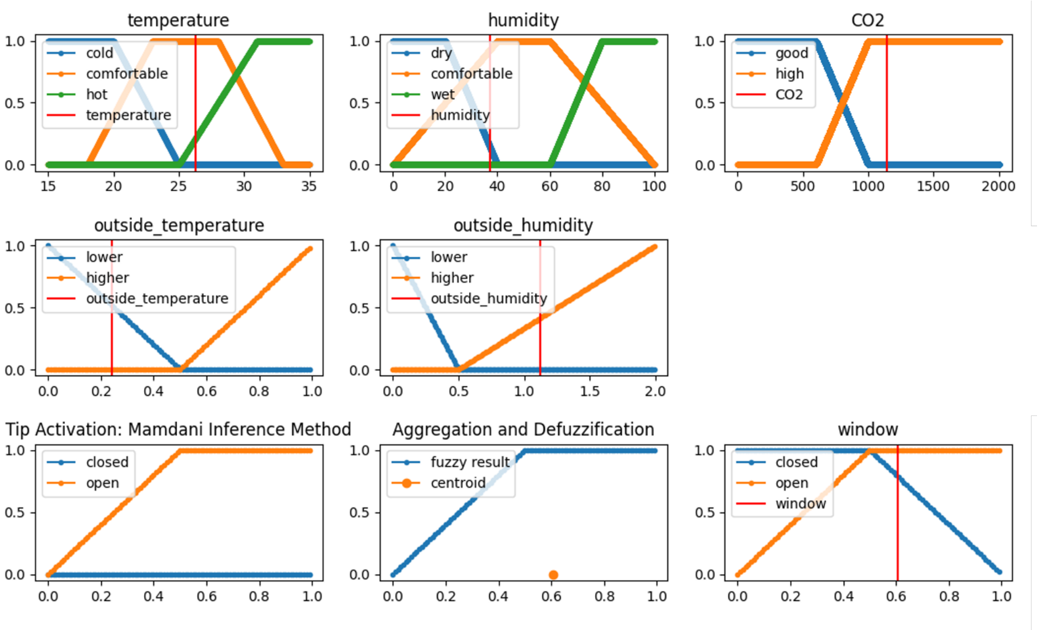
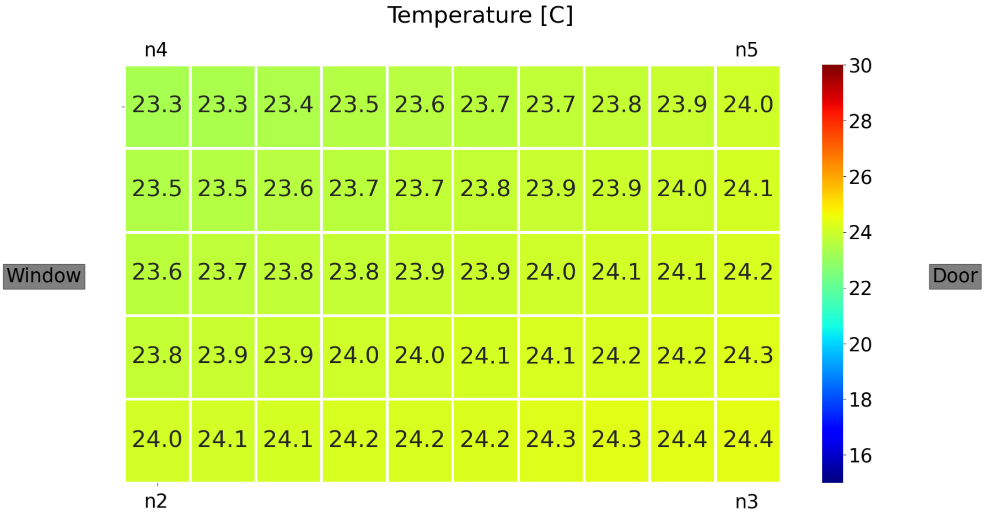


Figure 22. Fuzzy Logic visualization at 10:20, when the window was opened for the first time.

In order to better analyze the situation in the classroom described in Figure 22 at 10:20, the fog node provides services to show parameter maps that estimate the values of the collected parameters, i.e., temperature and humidity, from the sensor readings in this instant. As seen in Figure 23, the temperature variation around the room varies only by a few degrees. In Figure 24, a clear difference is shown between the area of nodes 4 and 5 and the area of nodes 2 and 3. This is due to the opening of the window closer to node 2 and the effect of the airflow from the window to the door near node 3.



23.3	23.3	23.4	23.5	23.6	23.7	23.7	23.8	23.9	24.0
23.5	23.5	23.6	23.7	23.7	23.8	23.9	23.9	24.0	24.1
23.6	23.7	23.8	23.8	23.9	23.9	24.0	24.1	24.1	24.2
23.8	23.9	23.9	24.0	24.0	24.1	24.1	24.2	24.2	24.3
24.0	24.1	24.1	24.2	24.2	24.2	24.3	24.3	24.4	24.4

Figure 23. Distribution of the temperature around the classroom.

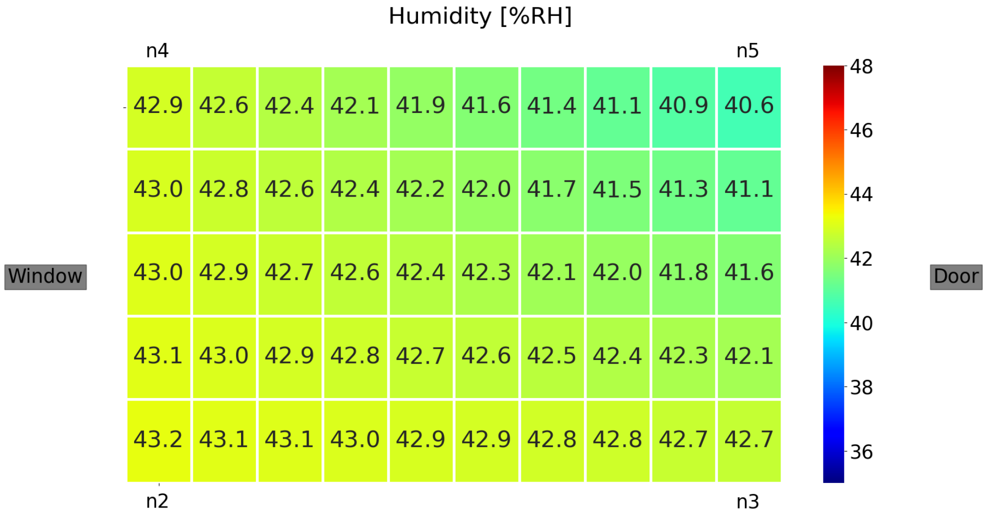


Figure 24. Distribution of the humidity around the classroom.

5. Conclusions

The introduction of IoT technologies in corporate buildings may help to obtain a more up-to-date and precise information about the status of the buildings, since they allow for connecting to every point of the building. Also, the use of complex algorithms, based on Artificial Intelligence techniques, may ease the analysis of the huge amounts of captured data and improve the operation and optimization of smart buildings. Although the combination of the Edge, Fog, and Cloud paradigms introduces new possibilities in IoT systems, it requires connecting heterogeneous devices, which may cause interoperability issues. For this reason, it is necessary to introduce interoperable architectures, based on standards and universal frameworks, to distribute consistently the resources and the services of AIoT applications for smart buildings, while ensuring interoperability. Although research of this topic is still incipient, there are some existing standards to introduce fog computing frameworks in IoT applications. One of the most known initiatives is OpenFog, promoted by the OpenFog Consortium and adopted as the IEEE 1934-2018 standard.

This work proposes a multi-layer architecture based on the OpenFog Reference Architecture aimed at corporative buildings. This architecture tries to guide developers to create complex applications aimed at improving the optimization and control of smart buildings by means of complex algorithms. The presented architecture implements the Edge, Fogm and Cloud layers by means of diverse types of devices. In particular, the Fog layer is responsible for collecting the data from the Edge nodes and executing the control algorithms for ensuring adequate indoor environmental conditions. Fuzzy logic was used for this task, since it is an intuitive technique that has been proven in monitoring IAQ systems. The architecture also allows for the integration of advanced services available as cloud services. MQTT is used for integration purposes, since it is quite easy to use, efficient, and provides built-on security mechanisms, which must be configured properly. A prototype, deployed at the Faculty of Engineering of Vitoria-Gasteiz, was used to validate the architecture. It included several IoT nodes at the edge, aimed at collecting environmental parameters of the building (temperature, humidity, luminosity, and CO_2 concentration) and operating actuators (mechanical windows). Several services were deployed at a Fog node: (1) IoT data storage; (2) control algorithms for operation of the actuators, based on Fuzzy logic techniques; and (3) connectivity to cloud services that were used in the control algorithm. Experimental results proved the validity of the presented approach, allowing for the introduction of complex algorithms in AIoT smart building applications.

Author Contributions: Conceptualization, I.C.; methodology, I.C., I.M.T. and J.X.; validation, J.X., I.M.T. and E.A.; investigation, J.X., I.M.T. and E.A.; software, J.X. and I.C.; writing—original draft preparation, I.M.T. and I.C.; writing—review and editing, I.M.T. and I.C.; visualization, I.M.T. and J.X.; supervision, I.C.; project administration, I.C. and O.B.; funding acquisition, I.C. and O.B. All authors have read and agreed to the published version of the manuscript.

Funding: The authors wish to express their gratitude to the Basque Government, through the project EKOHEGAZ II; to the Diputación Foral de Álava (DFA), through the project CONAVANTER; to the UPV/EHU, through the projects GIU20/063 and CBL 22APIN; and to the MobilityLab Foundation (CONV23/12), for supporting this work.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors wish to express their gratitude to Jose Luis Ledesma, research technician.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
AIoT	Artificial Intelligence of Things
ANN	Artificial Neural Network
BCI	BatiBUS Club International
eCO_2	Estimated Concentration of Carbon Dioxide
EIBA	European Installation Bus Association
EHSA	European Home Systems Association
IAQ	Indoor Air Quality
IB	In Band
IIC	Industrial IoT Consortium
IIRA	Industrial Internet Reference Architecture
IoT	Internet of Things
OOB	Out of Band
PCB	Printed Circuit Board
RA	Reference Architecture
SVM	Support Vector Machine
TVOCs	Total Volatile Organic Compounds

References

1. Jia, M.; Komeily, A.; Wang, Y.; Srinivasan, R.S. Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications. *Autom. Constr.* **2019**, *101*, 111–126. [CrossRef]
2. Starace, G.; Tiwari, A.; Colangelo, G.; Massaro, A. Advanced Data Systems for Energy Consumption Optimization and Air Quality Control in Smart Public Buildings Using a Versatile Open Source Approach. *Electronics* **2022**, *11*, 3904. [CrossRef]
3. Li, S. Review of Engineering Controls for Indoor Air Quality: A Systems Design Perspective. *Sustainability* **2023**, *15*, 14232. [CrossRef]
4. Khazaei, B.; Shiehbeigi, A.; Kani, A.R.H.M.A. Modeling indoor air carbon dioxide concentration using artificial neural network. *Int. J. Environ. Sci. Technol.* **2019**, *16*, 729–736. [CrossRef]
5. Zhang, T.; Li, X.; Zhao, Q.; Rao, Y. Control of a novel synthetical index for the local indoor air quality by the artificial neural network and genetic algorithm. *Sustain. Cities Soc.* **2019**, *51*, 101714. [CrossRef]
6. Dionova, B.W.; Mohammed, M.N.; Al-Zubaidi, S.; Yusuf, E. Environment indoor air quality assessment using fuzzy inference system. *ICT Express* **2020**, *6*, 185–194. [CrossRef]
7. Erozan, İ.; Özel, E.; Erozan, D. A two-stage system proposal based on a type-2 fuzzy logic system for ergonomic control of classrooms and offices. *Eng. Appl. Artif. Intell.* **2023**, *120*, 105854. [CrossRef]
8. Bushnag, A. An improved air quality and climate control monitoring system using fuzzy logic for enclosed areas. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 6339–6347. [CrossRef]
9. Hishamuddin, M.I.; Mansor, H.; Zahaba, M.; Yusoff, N.M.; Gunawan, T.S. Fuzzy Logic Controller of Indoor Air Quality Monitoring and Control System for Risk Reduction of COVID-19 Transmission. In Proceedings of the 8th IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA 2022), Melaka, Malaysia, 26–28 September 2022; pp. 313–317. [CrossRef]

10. Rahman, M.M.; Shafiullah, M.; Rahman, S.M.; Khondaker, A.N.; Amao, A.; Zahir, M.H. Soft Computing Applications in Air Quality Modeling: Past, Present, and Future. *Sustainability* **2020**, *12*, 4045. [CrossRef]
11. Alawlaqi, L.; Aldawod, A.; Alfowzan, R.; Albraheem, L. The Requirements of Fog/Edge Computing-Based IoT Architecture. In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2021, New York, NY, USA, 1–4 December 2021; pp. 51–57. [CrossRef]
12. Chen, Y.F.; Huang, D.H.; Huang, C.F.; Lin, Y.K. Reliability Evaluation for a Cloud Computer Network with Fog Computing. In Proceedings of the Companion of the 2020 IEEE 20th International Conference on Software Quality, Reliability, and Security, QRS-C 2020, Macau, China, 11–14 December 2020; pp. 682–683. [CrossRef]
13. Linthicum, D. Responsive Data Architecture for the Internet of Things. *Computer* **2016**, *49*, 72–75. [CrossRef]
14. Yousefpour, A.; Fung, C.; Nguyen, T.; Kadiyala, K.; Jalali, F.; Niakanlahiji, A.; Kong, J.; Jue, J.P. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *J. Syst. Archit.* **2019**, *98*, 289–330. [CrossRef]
15. Sabireen, H.; Neelanarayanan, V. A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges. *ICT Express* **2021**, *7*, 162–176. [CrossRef]
16. Dhaou, I.B. Design and Implementation of an Internet-of-Things-Enabled Smart Meter and Smart Plug for Home-Energy-Management System. *Electronics* **2023**, *12*, 4041. [CrossRef]
17. Filho, G.P.; Meneguette, R.I.; Maia, G.; Pessin, G.; Gonçalves, V.P.; Weigang, L.; Ueyama, J.; Villas, L.A. A fog-enabled smart home solution for decision-making using smart objects. *Future Gener. Comput. Syst.* **2020**, *103*, 18–27. [CrossRef]
18. Calvo, I.; Espin, A.; Gil-García, J.M.; Bustamante, P.F.; Barambones, O.; Apiñaniz, E. Scalable IoT Architecture for Monitoring IEQ Conditions in Public and Private Buildings. *Energies* **2022**, *15*, 2270. [CrossRef]
19. *IEEE Standard 1934–2018*; Adoption of OpenFog Reference Architecture for Fog Computing. IEEE Communications Society: New York, NY, USA, 2018. Available online: https://ieeexplore.ieee.org/document/8423800 (accessed on 4 October 2023).
20. OpenFog Reference Architecture for Fog Computing. 2017. Available online: https://www.iiconsortium.org/category/openfog/ (accessed on 24 July 2023).
21. Nakagawa, E.Y.; Antonino, P.O.; Schnicke, F.; Capilla, R.; Kuhn, T.; Liggesmeyer, P. Industry 4.0 reference architectures: State of the art and future trends. *Comput. Ind. Eng.* **2021**, *156*, 107241. [CrossRef]
22. Christou, I.T.; Kefalakis, N.; Soldatos, J.K.; Despotopoulou, A.M. End-to-end industrial IoT platform for Quality 4.0 applications. *Comput. Ind.* **2022**, *137*, 103591. [CrossRef]
23. Noura, M.; Atiquzzaman, M.; Gaedke, M. Interoperability in Internet of Things: Taxonomies and Open Challenges. *Mob. Netw. Appl.* **2019**, *24*, 796–809. [CrossRef]
24. Omoniwa, B.; Hussain, R.; Javed, M.A.; Bouk, S.H.; Malik, S.A. Fog/Edge Computing-Based IoT (FECIoT): Architecture, Applications, and Research Issues. *IEEE Internet Things J.* **2019**, *6*, 4118–4149. [CrossRef]
25. Rejiba, Z.; Masip-Bruin, X.; Marín-Tordera, E. A Survey on Mobility-Induced Service Mi-gration in the Fog, Edge, and Related Computing Paradigms. *ACM Comput. Surv.* **2019**, *52*, 33. [CrossRef]
26. Reka, S.S.; Venugopal, P.; Ravi, V.; Dragicevic, T. Privacy-Based Demand Response Modeling for Residential Consumers Using Machine Learning with a Cloud–Fog-Based Smart Grid Environment. *Energies* **2023**, *16*, 1655. [CrossRef]
27. Popović, I.; Rakić, A.; Petruševski, I.D. Multi-Agent Real-Time Advanced Metering Infrastructure Based on Fog Computing. *Energies* **2022**, *15*, 373. [CrossRef]
28. Tsipis, A.; Papamichail, A.; Angelis, I.; Koufoudakis, G.; Tsoumanis, G.; Oikonomou, K. An Alertness-Adjustable Cloud/Fog IoT Solution for Timely Environmental Monitoring Based on Wildfire Risk Forecasting. *Energies* **2020**, *13*, 3693. [CrossRef]
29. Ostrowski, K.; Małecki, K.; Dziurzański, P.; Singh, A.K. Mobility-aware fog computing in dynamic networks with mobile nodes: A survey. *J. Netw. Comput. Appl.* **2023**, *219*, 103724. [CrossRef]
30. Puliafito, C.; Mingozzi, E.; Longo, F.; Puliafito, A.; Rana, O. Fog computing for the Internet of Things: A survey. *ACM Trans. Internet Technol.* **2019**, *19*, 1–41. [CrossRef]
31. Ungurean, I.; Gaitan, N.C. Software Architecture of a Fog Computing Node for Industrial Internet of Things. *Sensors* **2021**, *21*, 3715. [CrossRef]
32. Gebremichael, T.; Ledwaba, L.P.; Eldefrawy, M.H.; Hancke, G.P.; Pereira, N.; Gidlund, M.; Akerberg, J. Security and Privacy in the Industrial Internet of Things: Current Standards and Future Challenges. *IEEE Access* **2020**, *8*, 152351–152366. [CrossRef]
33. Beraldi, R.; Alnuweiri, H. Distributed fair randomized (DFR): A resource sharing protocol for fog providers. In Proceedings of the 2019 4th International Conference on Fog and Mobile Edge Computing, FMEC 2019, Rome, Italy, 10–14 June 2019; pp. 29–36. [CrossRef]
34. Muneeb, M.; Ko, K.M.; Park, Y.H. A Fog Computing Architecture with Multi-Layer for Computing-Intensive IoT Applications. *Appl. Sci.* **2021**, *11*, 11585. [CrossRef]
35. Cuadra, J.; Hurtado, E.; Pérez, F.; Casquero, O.; Armentia, A. OpenFog-Compliant Application-Aware Platform: A Kubernetes Extension. *Appl. Sci.* **2023**, *13*, 8363. [CrossRef]
36. Bellavista, P.; Berrocal, J.; Corradi, A.; Das, S.K.; Foschini, L.; Zanni, A. A survey on fog computing for the Internet of Things. *Pervasive Mob. Comput.* **2019**, *52*, 71–99. [CrossRef]
37. Böhm, S.; Wirtz, G. PULCEO—A Novel Architecture for Universal and Lightweight Cloud-Edge Orchestration. In Proceedings of the 17th IEEE International Conference on Service-Oriented System Engineering, SOSE 2023, Athens, Greece, 17–20 July 2023; pp. 37–47. [CrossRef]

38. Lee, W.S.; Hong, S.H. KNX-zigbee gateway for home automation. In Proceedings of the 4th IEEE Conference on Automation Science and Engineering (CASE 2008), Kyoto, Japan, 25–28 May 2008; pp. 750–755. [CrossRef]
39. Luca, G.D.; Lillo, P.; Mainetti, L.; Mighali, V.; Patrono, L.; Sergi, I. The use of NFC and Android technologies to enable a KNX-based smart home. In Proceedings of the 2013 21st International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2013), Split, Croatia, 18–20 September 2013. [CrossRef]
40. Bujdei, C.; Moraru, S.A. Ensuring comfort in office buildings: Designing a KNX monitoring and control system. In Proceedings of the 2011 7th International Conference on Intelligent Environments (IE 2011), Nottingham, UK, 25–28 July 2011; pp. 222–229. [CrossRef]
41. Vanus, J.; Gorjani, O.M.; Bilik, P. Novel Proposal for Prediction of CO_2 Course and Occupancy Recognition in Intelligent Buildings within IoT. *Energies* **2019**, *12*, 4541. [CrossRef]
42. Asensio, J.A.; Criado, J.; Padilla, N.; Iribarne, L. Emulating home automation installations through component-based web technology. *Future Gener. Comput. Syst.* **2019**, *93*, 777–791. [CrossRef]
43. Alonso, J.; Orue-Echevarria, L.; Casola, V.; Torre, A.I.; Huarte, M.; Osaba, E.; Lobo, J.L. Understanding the challenges and novel architectural models of multi-cloud native applications—A systematic literature review. *J. Cloud Comput.* **2023**, *12*, 6. [CrossRef]
44. Angel, N.A.; Ravindran, D.; Vincent, P.M.D.R.; Srinivasan, K.; Hu, Y.-C. Recent Advances in Evolving Computing Paradigms: Cloud, Edge, and Fog Technologies. *Sensors* **2022**, *22*, 196. [CrossRef] [PubMed]
45. Zhang, J.; Tao, D. Empowering Things with Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things. *IEEE Internet Things J.* **2021**, *8*, 7789–7817. [CrossRef]
46. Orive, A.; Agirre, A.; Truong, H.L.; Sarachaga, I.; Marcos, M. Quality of Service Aware Orchestration for Cloud–Edge Continuum Applications. *Sensors* **2022**, *22*, 1755. [CrossRef]
47. Arduino® MKR WiFi 1010. Available online: https://docs.arduino.cc/resources/datasheets/ABX00023-datasheet.pdf (accessed on 20 September 2023).
48. Mishra, B.; Kertesz, A. The use of MQTT in M2M and IoT systems: A survey. *IEEE Access* **2020**, *8*, 201071–201086. [CrossRef]
49. Jove, E.; Aveleira-Mata, J.; Alaiz-Moretón, H.; Casteleiro-Roca, J.L.; Blanco, D.Y.M.D.; Zayas-Gato, F.; Quintián, H.; Calvo-Rolle, J.L. Intelligent One-Class Classifiers for the Development of an Intrusion Detection System: The MQTT Case Study. *Electronics* **2022**, *11*, 422. [CrossRef]
50. Donno, M.D.; Tange, K.; Dragoni, N. Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog. *IEEE Access* **2019**, *7*, 150936–150948. [CrossRef]
51. BME6xy: Handling, Soldering and Mounting Instructions. Available online: https://www.bosch-sensortec.com/products/environmental-sensors/gas-sensors/bme680/ (accessed on 13 October 2023).

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.