IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 July 2024, accepted 9 August 2024, date of publication 16 August 2024, date of current version 26 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3444471

RESEARCH ARTICLE

Design and Implementation of a Smart Campus Flexible Internet of Things Architecture on a Brazilian University

BRUNO OLIVEIRA ZARPELLON, LUIS DE ORO ARENAS, EDUARDO PACIÊNCIA GODOY,
FERNANDO PINHABEL MARAFÃO, (Member, IEEE),
AND HELMO K. MORALES PAREDES, (Senior Member, IEEE)
Institute of Science and Technology, São Paulo State University (UNESP), Sorocaba 18087-180, Brazil

Corresponding author: Bruno Oliveira Zarpellon (bruno.o.zarpellon@unesp.br)

This work was supported in part by the São Paulo Research Foundation (FAPESP) under Grant 2022/15423-3, in part by the National Council for Scientific and Technological Development (CNPq) under Grant 309297/2021-4, and in part by the Coordination for the Improvement of Higher Education Personnel (CAPES) under Grant 001.

ABSTRACT This work proposes a flexible and comprehensive Internet of Things (IoT) architecture designed for application on smart campuses to build their smart infrastructure and facilitate their transition to becoming smart. The concept of a smart campus is derived from the concept of a smart city, which was developed to demonstrate how urban areas were addressing their new and dynamic challenges by integrating technology and data-driven decision-making. Although the concept of the smart campus was initially conceived to study the smart city in a less complex setting, over time, both concepts have evolved into distinct areas of inquiry, exhibiting unique characteristics and impacts. In general, a smart campus is a university campus where information and communication technologies and IoT are applied to some or all the campus processes, thereby making these processes more efficient, cost-effective and environmentally sustainable for the institution, its members and the surrounding community. Additionally to presenting the proposal for a flexible and comprehensive architecture for a smart campus, this paper also demonstrates its interpretation for implementation on a Brazilian campus transitioning to become smart, the Institute of Science and Technology, Sorocaba (ICTS) - campus of the São Paulo State University (Unesp) in Sorocaba, Brazil. To the best of the authors' knowledge, there is a scarcity of existing works which address this level of detail when proposing an IoT architecture for a smart campus. The implementation of this architecture has demonstrated that it can be successfully deployed using only open-source technologies. Furthermore, it has been shown that anyone with access to the campus website can access most of the data collected and stored on the system.

INDEX TERMS Data integration, open-source technologies, environmental sustainability, energy consumption monitoring, Internet of Things, information and communication technologies, smart infrastructure.

I. INTRODUCTION

The initial concept of a smart city was introduced at the beginning of the 1990s to demonstrate how cities were beginning to utilize technologies and innovation to address their urban development challenges [1]. A report published by the United Nations in 2018 on the world's urbanization process revealed that by that time, more than half of the world's population was living in urban areas - 55% - in contrast to 30% in 1950. The projections and estimates made for this report also indicated that the human population growth in the future will be primarily accounted for by the increase in the urban population, with approximately two-thirds of the world's population - 68% - living in cities

The associate editor coordinating the review of this manuscript and approving it for publication was Harikrishnan Ramiah.

by 2050. In addition to the numerous benefits associated with urbanization and city growth, such as economic growth, poverty reduction and human development, these processes also give rise to several challenges, including increased energy consumption and demand for water, land, building materials, food, pollution control measures, and waste management. Therefore, the smart city concept is receiving increasing attention as a means of addressing these challenges and providing enhanced quality services, fostering local economic competitiveness, improving service delivery, improving efficiency and reducing costs, increasing effectiveness and productivity and addressing congestion and environmental issues [2], [3].

In light of the considerations mentioned above and the similar characteristics between university campuses and cities, implementing the smart city concept on a university campus represents a significant opportunity to enhance the smart concept application. This is because university campuses have characteristics that facilitate the application of the smart city concept, namely, a single management unit/organization, which allows for centralized decision-making and the standardization of technologies and equipment brands that are utilized throughout the campus; even though some campuses extend in size similarly to cities, they tend to have smaller sizes, which facilitates communication lines and overall management; the campus community - students, professors, and employees - tends to be more willing to adopt and promote innovations, as well as to get involved as developers and in campus testing [4].

Several definitions of smart campus can be found in the literature [5], [6], [7], [8], [9], [10], [11]. Consequently, there is no consensus on a universal definition. However, delving into all these definitions is beyond the scope of this work. Therefore, in this paper, a definition is introduced that has been derived from a comprehensive literature review. This definition aims to summarize the main aspects of all these definitions into a single cohesive summary:

The term smart campus is utilized to describe a university campus which employs advanced and/or intelligent technologies to enhance its overall operation, acting on the pillars of education and research, energy, environment, management or governance, people, and technologies. This improved operation will have a positive impact on several campus areas, including, the quality of teaching and research developed, water and energy monitoring and management, besides renewable energy generation. It will also improve the quality of life of all members of the campus community, such as employees, professors and students. Additionally, it will enhance the university administration and its assets management; as well as internal and external campus mobility, which may be limited by practical considerations. Finally, it will reduce the environmental impact generated by the campus operation. These improvements collectively define the optimal operation of a smart campus. Nevertheless, a campus may still be considered a smart campus even if it has not yet implemented improvements in all of these pillars of action (e.g. improvements only in the energy and environment areas), due to the gradual and campus-specific nature of the implementation process.

The smart campus concept was initially introduced with the objective of utilizing the campus as a ‘‘smart city studying lab’’. However, the advantages brought by its application led to the concept becoming an extremely relevant independent area of study in its own right. The main benefits that can be achieved by implementing a smart campus in a university, as outlined by [7] and [12], include the formation and retention of top students and faculty, the extension of reach without facility expansion, higher efficiency and productivity, a richer learning, teaching, and research environment, and the resolution of traditional learning barriers via technological enablers. The obstacles to implementing a smart campus can be overcome through the use of technological enablers, which can result in lower capital expenditure and operating expenditure costs. Additionally, a smart campus can provide an interactive and creative environment for students and faculty, as well as a smart energy management system, an effective surveillance system, and real-time incident warnings. Furthermore, an automatic maintenance and business process system and efficient parking and access control management can be implemented, among several others.

Considering all these advantages that implementing a smart campus can bring, the Institute of Science and Technology, Sorocaba (ICTS) - a campus of the São Paulo State University (Unesp) in Sorocaba, Brazil - has committed itself to initiating a transition process. The objective was to become a smart campus by the end of 2021 and the beginning of 2022. This entailed improvements in the campus’ technical and economic aspects, as well as improvements in quality of life and learning for all members - students, professors and employees. The transition process starts with designing an Internet of Things (IoT) architecture, which is employed to accommodate the infrastructure to monitor campus variables, such as energy and water consumption, and later, the technological support for other campus processes gradually.

Despite the growing interest in the smart campus concept, as evidenced by the numerous publications in conferences and journals, works proposing architectures for implementing smart campuses are scarce. As examples of works that propose smart campus architectures, we could cite the works conducted in [13], [14], and [15]. In the first one [13], the authors propose a smart campus model with an architecture composed of layers, aiming to enhance its flexibility by facilitating the incorporation of new solutions into existing infrastructure. In [14], the authors characterize their campus architecture, which include a dashboard to enhance data visualization for decision-making. Additionally, they conduct a study of a predictive model for power factor. In the last of the three references, [15], a five-layered framework is presented to help implement the sustainability leveraging basic ecosystem. The framework also includes suggestions for how to add use cases related to Covid-19 to the given architecture. While the number of works proposing these architectures is limited, when this architecture is presented or mentioned in the context of smart campus applications, it often lacks sufficient detail for replication. Furthermore, it is not always clear if the architecture can be flexibly adapted and how open-source software and hardware can be utilized.

The objective of this work is to propose a flexible, comprehensive and replicable IoT architecture based on low-cost technologies designed for application on smart campuses to facilitate their transition to becoming smart. Furthermore, this architecture is implemented on a Brazilian university campus, the ICTS/Unesp transitioning to a smart campus to demonstrate its functionality and suitability for use on smart campuses. The main differentials of the proposed architecture from other architectures presented in the literature are providing details on each layer function, how each application is implemented, utilizing open-source software and hardware, and a comprehensive architecture, allowing it to be used on campuses of varying sizes. Most importantly, its implementation on the ICTS/Unesp campus also offers access to all monitored data for anyone accessing the campus website, which is seldom done in literature or by companies. This allows anyone to conduct studies on real data, such as consumption prediction or pattern identification.

The remainder of this paper is organized as follows. In Section II, similar works in the literature are presented. The architecture conceptual model and its operational flow are given in Section III. Section IV outlines the interpretation of the model's implementation on the ICTS/Unesp campus and presents the architectural implementation results. Section V presents a discussion comparing the architectural model with similar works found in literature and provides guidance and tips for implementing the architectural model on a campus. Finally, in Section VI, the conclusions of this work are summarized.

II. RELATED WORKS

The first work worth mentioning is the work made by Brand et al. in [13]. This study indicates a gap in the literature regarding smart campus model proposals and the flexibility of these models. In response, the authors propose a smart campus model that utilizes the existing infrastructure of their campus instead of installing new technologies. The model follows the general structure for smart cities and models found in the literature, employing a layered approach, including as layers requirements gathering, perception, network, system, and application. This structure provides flexibility and abstraction from an application point of view, allowing for the adoption of new technologies without impacting the basic implemented infrastructure. The layers function independently, facilitating easy data sharing between them. The model was implemented at the Unisinos Campus located in São Leopoldo and include two applications, temperature measurement and a location app to guide users within the campus. These implementations demonstrated the model's effectiveness in incorporating new hardware components and proved that the installed devices could be utilized in multiple application with distinct configurations and purposes.

In [14], the authors presented a dashboard implemented on the Facens campus to monitor several relevant variables, such as energy consumption, meteorological conditions, PV energy generation, and parking lot occupancy, among others. Throughout the work, authors contextualize why collecting and joining data is important, besides the challenges associated with integrating them. The dashboard construction is presented, showing decisions taken during its development, its final web page version, and diagrams of each dashboard's component system's architecture. The paper also works on the importance on power factor for energy quality, and the authors test the correlation between meteorological variables and power factor, from which they concluded that there isn't any correlation. This way, they predict power factor based on energy consumption and solar panel utilization, achieving low variance and great similarity of lines when plotting real vs predicted power factor. Although the paper presents a well-developed and functional smart campus architecture while also making use of the data collected by it, it is more of a specific use case, focused on the dashboard presenting the collected data and tailored to the Facens and smart cities' needs. This is because the Facens smart campus is highly focused on the development of smart cities. Furthermore, the architecture structure is not adequately explained due to its lack of focus on the reproduction aspect and its implementation is associated with the use of specific technologies, which lack flexibility or generalization. Additionally, the software adopted for processing and storing the collected data is not open-source, made available by a partnership of Facens with its manufacturer.

In [15] a five-layered framework based on IoT for implementing a smart campus is presented. The framework is based on three focus areas considered as the ones that comprise a smart campus. Additionally, a three-process step is suggested for implementing new use cases on the implemented campus. Finally, the authors present a table with possible applications for improving a monitoring system for COVID-19. The paper covers several relevant topics and presents an interesting approach for building an architecture for implementing a smart campus. However, due to its limited length, the layers require further elaboration to enhance their clarity and facilitate their practical application.

The next work worth mentioning is the one done in [16]. In it, the authors present the development of an environmental monitoring system via air quality as the first stage of developing a smart campus in Zacatencos's campus at the National Polytechnic Institute in Mexico City. For this, they primarily focus on the network design, indicating why LoRaWAN was chosen and simulating network coverage over campus, followed by a network real testing using Received Signal Strength Indicator (RSSI) and Signal to Noise plus Interference Ratio (SNIR) as test parameters, proving that the simulated results are very similar to real-world case, except for the software not taking into account the presence of buildings in the way of the transmitted signal. Both simulation and real test results prove that the network will work as expected and within the campus's coverage requirements. After presenting the adopted network design specifications and testing, the authors present the development of their air quality monitoring node. They identify the components utilized in its construction, focusing on low-cost sensors, and present air quality data collected after its implementation on the campus. This is an interesting application for implementation on a smart campus, tackling the environmental aspect of the campus. Nevertheless, more comprehensive contextualization of the potential for this application to facilitate the development of the smart campus at the National Polytechnic Institute in Mexico City is needed, since it represents the initial stage of this smart campus.

In [17], the authors review IoT concepts and show their application in the Flipped Classroom concept on a campus, a model where students watch video lessons at home and go to classes to do their homework. According to the authors, the Flipped Classroom model was already implemented, and they explain how the remote classes are prepared and transmitted, as well as statistics of the number of students watching the classes on YouTube. Finally, they present a comparative table, stating that using the Flipped Classroom concept allowed the students to have better results than students who studied using traditional approaches, while most of the Flipped Classroom students agree that the video lessons are better than the traditional ones. It is an interesting and niche application for the evolution of the education aspect of a smart campus. The authors present arguments for how this application can enhance students' overall results instead of how to apply it on a campus and how to create an architecture infrastructure to support this application properly.

Authors in [18] propose a smart campus monitoring system based on IoT to monitor air quality and analyze students' status based on their physical data to enhance learning. A graph showing the air quality monitoring throughout a 15-day period is presented, alongside students' heartbeat and respiratory rate during a 100-minute class, these latter two being used by the professors to know when the students' learning rate was deteriorating and take actions to improve it again during the rest of the class. Even though the applications proposed for a smart campus are interesting and embrace more than one area of interest in the campus, the implementation of the monitoring systems, as well as the architecture supporting their implementation, is only explained superficially, lacking the necessary details to facilitate replication of the systems and better understanding them.

The work conducted in [19] does a study on smart campuses using IoT to develop a smart campus in Universitas PGRI Yogyakarta, Indonesia, and presents implementations for the mentioned campus in the areas of smart education, which consists of eLearning, Virtual classroom, as well as smart parking a parking system that provides information related to the available parking lot, and also provides information when the parking lot is full, and smart room, a system that provides information related to the vacant room. These applications are explained focusing on what they can improve on the campus, and how to build them is briefly described. Although the proposed applications bring value to a campus, they lack enough detailing for replication and functional results for a use/test case presentation.

Reference [20] presents the design of a monitoring system based on IoT for application on the smart campus project in Universitas Udayana - also in Indonesia - applied to environmental variables monitoring. The hardware utilized for two possible scenarios when collecting data and the adopted server characteristics are presented. The presented results include the graphics generated by a temperature sensor for a few minutes on the interface. The system description lacks explanations of the interconnections between its various elements, as well as guidance on how they should be implemented or alternative applications. This characterizes a niche-specific application.

Work conducted in [21] brings a different case of building an IoT monitoring system, showing the implementation of one to monitor a campus hospital. The objectives are to provide greater medical coverage and patient-doctor remote proximity; monitor patient's data in real-time for diagnosis, treatments and issuing alerts in emergency situations; and monitor healthy people's data to detect early stages of diseases. The authors show the client's system developed, the data conversion system and the data transmission system, and thoroughly cover the system's overall functioning, how it is built and its supporting framework, while also presenting the data tables for storing relevant information and existing differences between the system's web application and Android application. While this work does not propose a general architecture for implementing a smart campus, it indicates how to implement a framework specific for health monitoring, a subject that is often neglected in smart campus development. Given that this framework is intended for a niche application, its characterization is of particular importance. Medical data have specificities related to the format in which they are generated and the protocols for their communication, and the authors provide a detailed account of how to work with this in a smart campus context.

Other works on IoT architectures implemented on power systems that could be part of a smart campus but are done without mentioning the concept are [22], [23], [24].

The first work [22] is the most interesting of the three. On it, the authors propose a low-cost IoT system to monitor mini and micro-PV generation, which monitors direct voltage and current, alternate power and additional meteorological variables pertinent to the PV generation. Throughout the work, the authors thoroughly present the architecture designed, going into detailed explanations of each component, how the datalogger was built and its functioning processes, while also indicating how the data payloads are built and indicating the communication network adopted, alongside its specificities. Furthermore, the authors indicate how the monitored data are stored on the cloud and later retrieved via the web application. Some literature works that resemble their work are presented, where differences are pointed out. The system after its implementation is presented, comparing by MAE, RMSD and WAPE its measured variables values to the ones measured by an industrial datalogger to check the trustworthiness of the system data readings, which results in four out of eight parameters being inside acceptable measurement errors - ambient temperature, DC current strings 1 and 2, and AC Power - and giving insights of why the other parameters do not match the expected value and possible solutions for increasing measurements precision. Finally, they present the cost of building this whole system and compare it to another solution found in literature and to a commercial datalogger, proving that their monitoring system is approximately twice as cheap as the alternatives.

The second one [23] presents the distributed control, heat recovery and load condition monitoring on a combined heat and power (CHP) generation on a microgrid running on a university campus. The objective of the work is to present a methodology for the condition monitoring of critical components of a campus microgrid for the proposes of preventive maintenance and protection. The microgrid diagram and the simulations made to study the load's behavior when submitted to abnormal conditions are presented. It is shown how the thermal power gain is proportional to electrical power gain, and the water flow on the system also impacts both these factors, being 43 gallons per minute the water flow with the best power gains. The proposed study demonstrates that the use of remote automation to control the processes of the CHP system results in higher thermal recovery for space heating. Furthermore, the application of smart monitoring and control of electrical loads provides energy savings and warnings for preventive maintenance.

The last paper [24] proposes a power monitoring system for smart grids assisted by IoT. The system is explained via a block diagram, detailing which components were used and their function. As results, the authors showed a picture of a functional prototype for the system, while also presenting the graphics generated when monitoring it. While the proposed system is an interesting alternative for power monitoring, the authors have not provided sufficient details regarding its overall architecture, which impedes the system's reproduction. Additionally, the focus of the system is on end users; therefore, the core idea was to build something simple and with ease of implementation.

A review of the pertinent literature reveals an absence of works proposing an effective IoT architecture for the development of a smart campus. Among them, the work conducted in [13] is particularly noteworthy. Other related works include [14], [15], but these approaches lack a comprehensive explanation of the architecture layers, limiting their systems' overall replicability. Additionally, the architecture approach presented in [14] is closely aligned with the specific paid software, with no discussion of alternative technologies.

The remaining references [16], [17], [18], [19], [20], [21], [22], [23], [24] primary focus on applications for a smart campus, though they do briefly propose or present architectures for these applications. Most of these proposals do not clearly involve open-source technologies and lack further development in terms of flexibility and scalability. This paper aims to address these gaps by improving the generalization of the proposed architecture by means of the construction of its layers. Most importantly, we develop a real case scenario for implementation, where all collected data are presented and made available to anyone interested in using it for studies.

III. THE ARCHITECTURE

In this section, we will present and thoroughly explain the architecture, starting with characterizing its layers, presenting its functional workflow, and how the scalability aspect is tied up to it.

A. DEFINITION

The proposed architecture was sketched to be a flexible, scalable, and low-cost IoT architecture, to be the main backbone of any campus transitioning to a smart one or any campus that already has some smart initiatives but lacks a system to monitor and/or control its processes. The layered model is the best architecture model that fits all the previously mentioned characteristics, since each layer will compartmentalize components that share a similar functionality. The layer's separation allows for a clearer understanding of the model so as understand where to act when any potential problems exist or innovations should be applied. In addition, the organization in layers allows the architecture to become generalized in a way that even if different campuses implement the architecture and different technologies are applied on the layers for each campus doing it, the architecture operation remains the same. Using open-source technologies when implementing the architecture is suggested to keep it low-cost, but the way it is built allows one to utilize other paid technologies when implementing it on its own campus.

As Figure 1 shows, the architecture is divided into six layers, namely, from bottom to top: Physical Process, Reading, Processing and Transmitting Data, Network, Integration, Data Storage, and Interaction. The arrows on the figure indicate data flow between or within each layer. Next, each layer will be explained in further detail.

Physical Process: the name of the first layer is self-descriptive; this layer contains all physical processes that should be monitored on the campus, alongside the sensors which collect data from these processes. These processes can relate to campus management, its members' quality of life, economical and environmental aspects. Some examples

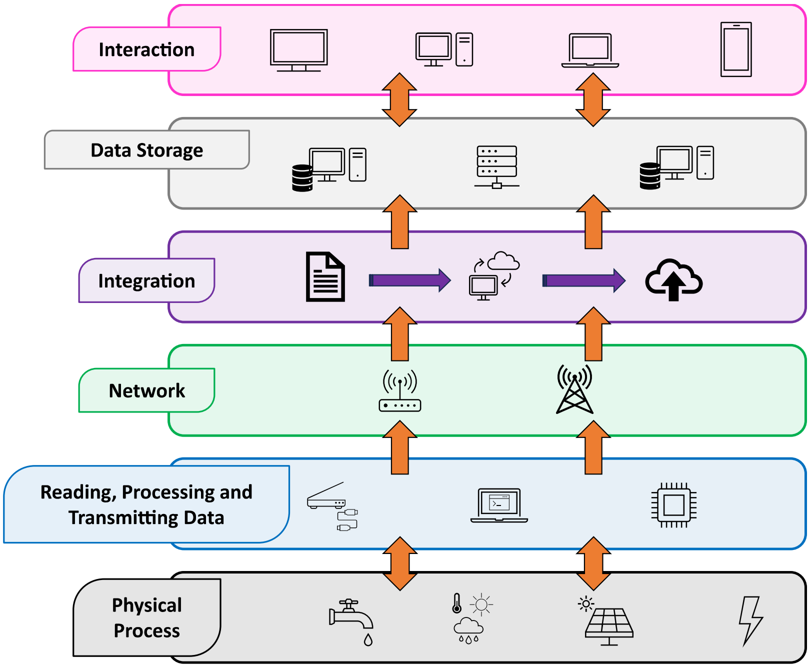


FIGURE 1. IoT layered Architecture defined for implementing the smart campus.

worth mentioning are, respectively: campus access control and parking lot occupancy management; classroom lighting and air conditioning utilization to improve comfort; water and energy consumption; greenhouse gas emissions and air quality; and so many others. As stated, these processes are all related to different campus areas, and any number of these can be included in this layer of the architecture, since the monitored processes depend and vary according to the needs of the campus that is implementing them. As a starting point, the authors suggest this implementation on physical processes that already have a preexisting sensor's infrastructure measuring them, so that the initial implementation would only be worried on how to retrieve data from the sensors. Nevertheless, this is not a limitation, and implementing it on processes without prior sensing devices can also be done easily. If there is any control of the physical processes, such as changing an air conditioning set point, the actuators are also in this layer.

Reading, Processing and Transmitting Data: this layer comprises the devices that retrieve and process the data read from the physical processes by their respective sensors. This layer is important for extracting data from the sensors, since most sensors can send data only via a wired communication, and when doing so, they adopt specific communication protocols which compact data in formats that need to be interpreted. It is also important to send the commands to the physical processes actuators when working with some control of them. Therefore, this layer can be seen as a layer responsible for reading physical processes data, making a pre-processing of these data, formatting them in the way predefined for the rest of the architecture to work on and then sending it to the integration via the network, and to send commands to any physical process which are being controlled. The data reading can be done on any device which has communication capabilities with the sensors, but when thinking about low-cost solutions, the authors encourage utilizing micro-controller boards, which offer good processing capabilities alongside low-cost hardware and open-source software. Each variable or process being monitored must have a predefined reading interval, which is defined according to the frequency each process needs to have its data updated and respecting the communication protocol and hardware limitations. After reading and pre-processing the data, the devices forward them via the network.

Network: this layer consists of the communication network adopted for data exchange between the layer below - device - and the layer above - integration. Since there are many communication technologies available, when defining the network utilized for this layer, one has to consider the pros and cons of the available options, such as network coverage vs campus size, the necessity of using repeaters, data transmission speed/network adaptability to transmit real-time data, portable network devices energy usage, wired vs wireless networks, among others. After defining the network utilized to transmit the data, the communication protocol must also be specified. The definition of the network is also important for the reading, processing and transmitting data layer, since each device must be able to send data to the integration via the chosen network and protocol. It is not impossible to have more than one network working in parallel, but it is advisable to utilize only one, since, if there are two or more networks, it is necessary to have a network adapter and, sometimes, conversion of protocol prior to the integration, since it is probably located on a server which can only communicate with one network.

Integration: the integration layer is responsible for receiving data coming from the devices and processing it in the format expected on the data storage. Depending on the chosen devices, writing data on the configured databases can be done directly by the devices, but in the proposed architecture, it was chosen to separate both processes to clarify better where each thing is happening in the architecture. So, the integration layer consists of a server where an algorithm runs 24/7 waiting for any new data arrival. When they arrive, the server has built-in logic that interprets the data coming and processes them according to how it will be stored, to then store them on the data storage. Because of this, it is interesting to send all data in a predefined pattern - independently from which network is utilized to send it -, since this can simplify the logic required on the integration code to interpret the data correctly. The integration can also work as a middleware in the opposite direction, receiving control commands from the interaction layer and sending them to the devices via the network.

Data Storage: on this architecture layer are the databases for storing all the collected data from the monitored systems. This is an important layer because storing data is what allows them to have intelligence attached to them, in opposition to only monitoring them with sensors. The utilized database must also be chosen based on the desired characteristics, since there are databases based on tables, time series data, or even data lakes, which can store whole files. It is important to have a database that can work with adding data at a later date, since when any data are lost due to network problems, if it was backed up, it can then be written on the database later. Also, the chosen database must have ways to connect with the layer below - integration - and the upper layer of the architecture - namely, the interaction -, so therefore, defining which database will be used to store data is a process tied up with determining how to integrate data and how to display them. It is possible to utilize more than one database in parallel but bear in mind that this always increases the complexity of this layer and the memory necessary for the servers running it.

Interaction: on this layer a human-machine interface (HMI) is implemented, meaning, a graphical interface for the presentation of relevant data collected and stored in the data storage. This interface should contain any information deemed important for the campus implementing the architecture, such as: historical energy consumption graphics, daily water usage, campus mapping and parking lot occupancy, among others. Besides presenting data, it is also located on this layer any data analysis which must be conducted, being it necessary for data presentation - for example, graphics of energy consumption prediction - or for campus processes control - such as turning an air conditioner on or off when a determined temperature is registered.

B. ARCHITECTURE GENERAL WORKFLOW

In this section, the overall operation of the architecture is explained, which can be visualized on the flowcharts of Figure 2. The flowcharts represent the general architecture workflow, which is split into two separate workflows for a better understanding. The orange arrows indicate data flow, and the blue arrows indicate the command's information flow.

On the left one, the monitoring process workflow is shown, independently of which source generates the data. This flow is: data from the physical process are constantly measured by its sensors and periodically requested by the devices, which then process the data and send them via the network to the integration, which then processes the data and writes them on the configured data storage; finally, the configured connection between the interaction and the data storage makes it possible to retrieve data for visualization at any time.

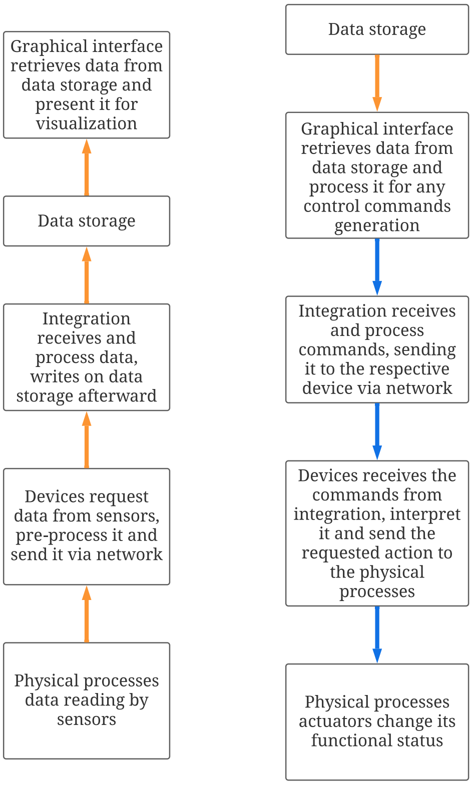


FIGURE 2. Architecture general operation flowcharts.

On the right, the workflow represents the generalized functioning of the control of processes in the architecture, when those are implemented. First of all, the interaction retrieves the necessary data from the data storage, processing them and generating any commands necessary to alter the physical processes operation; next, these commands are sent from the interaction directly to the integration - and therefore, the data storage layer is not utilized for controlling the physical process layer -, which identifies these commands and format them to be on the devices expected format; these commands are then sent to their respective devices via the network; and finally, the devices interpret the commands received and sent them to the physical process actuators, which only then change the application's running status.

Although the workflows presented here are quite simplified, they represent well how the data are collected, stored, visualized, and control commands are communicated, highlighting that the architecture operation is pretty much straightforward and flexible. Additionally, they can be used as a reference when implementing the architecture on a university campus, to understand better each layer responsibility and what applications are a better fit for each of them, depending on what characteristics the designed smart campus must have.

C. ARCHITECTURE SCALABILITY

The layered format adopted for the architecture makes it scalable for application on any university campus. The system's scalability is contingent upon the fact that, even though each layer component may have its own particularities when applied on different-sized campuses, it is possible to maintain the same architectural organization even when scaling it up for larger-sized campuses. This is accomplished by expanding the number of resources utilized on each layer. However, in some situations, this can result in changes in within layer interpretation. An example of that is the network layer. In the case of campuses of a similar scale to small cities, one potential approach to scaling the architecture up is to build local instances of the lower logical levels, including physical process, reading, processing and transmitting data, as well as the network layers. This can be achieved by establishing "islands" of these instances for each university's building or sector, which share a common network for transmitting data, but which are isolated from one another. In addition, for this case, the server running the upper layers would be ideally implemented on the cloud, allowing all the different campus spaces to send their data without the need for repeaters or concern about the adequacy of the network range to reach the database servers. Furthermore, the implementation of cloud-based solutions is an optimal choice for any campus environment, since they offer the flexibility to expand processing and storage capacity as needed.

IV. CASE STUDY: IMPLEMENTING THE ARCHITECTURE ON A SMALL SIZED BRAZILIAN CAMPUS

This section will address how the architecture was interpreted to be implemented on the Unesp campus at Sorocaba, the ICTS. The components utilized in the architecture for its implementation will be identified, and an explanation of how everything was gathered under the architecture and integrated as a sole system will be presented while also presenting the motivations behind the decisions made throughout its implementation. Before going into details of our implementation of the architecture, it is important to point out some of the main characteristics of the campus to help understand the decisions that were made related to the interpretation of the architecture for our campus.

The ICTS/Unesp campus currently has two graduation programs and five post-graduation programs, totaling seven ongoing courses divided into two areas of study, and with approximately 700 students distributed on these programs. The campus's total area is 16,172 m^2, from which 7,600 m^2 are constructed area. For a university campus, it is a relatively small size and a small number of programs. Taking Facens, a university center also located in Sorocaba, as a comparison: Facens has nineteen undergraduate programs and twenty two graduate programs, with more than 4,000 students and a total area of 100,000 m^2 [25]. Another example of comparison is the Unicamp campus, which has a smart campus under development: more than 3.5 million m^2 of area with 559,000 m^2 of built area, as well as approximately 53,107 students between its three campuses - Campinas, Piracicaba and Limeira - of which around 41,000 attend the Campinas campus [26]. This comparative data are important to understand that since ICTS/Unesp is a smaller campus, the infrastructure needed to monitor it is also delimited according to its size. Figures 3 and 4 show, respectively, the ICTS/Unesp campus's top perspective and a more detailed mapping of its buildings.

FIGURE 3. Campus top view retrieved from satellite images.

A. LAYERS IMPLEMENTATION

Figure 5 shows the layer's adaptation to the ICTS/Unesp campus. The following paragraphs provide further information on each layer's implementation.

Entering on the architecture implementation, the first layer designed was the physical process layer. The processes defined as the ones of interest in this initial smart campus transition were processes that could bring economical and environmental benefits to the campus and had previously installed sensors for its monitoring. The processes that fit these characteristics, and therefore are the ones currently

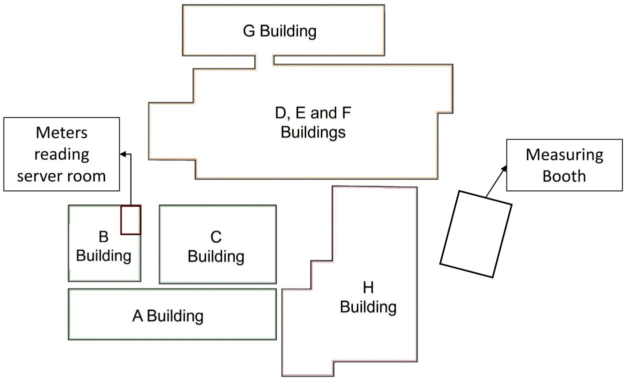


FIGURE 4. Campus buildings detailed view.

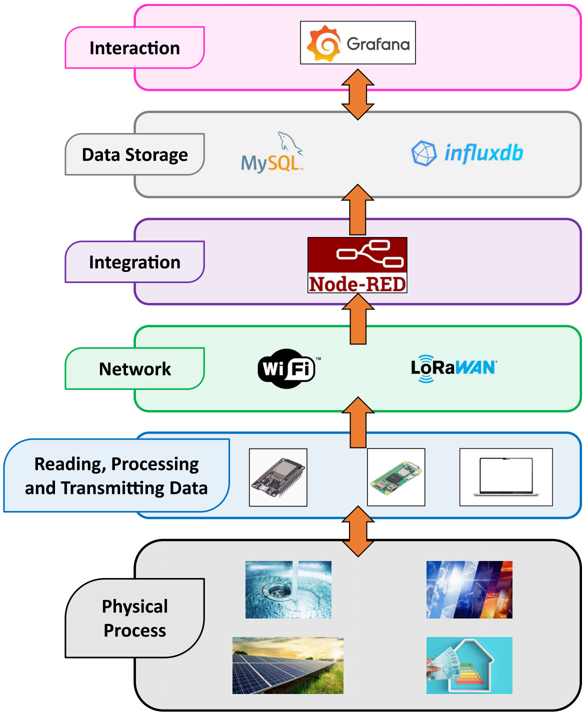


FIGURE 5. Architecture layers adapted to the ICTS/Unesp case study.

monitored via the architecture implementation are water consumption, meteorological variables, photovoltaic (PV) energy generation and energy consumption. There is no actuator characterization on the physical process layer because, during this first architecture implementation on the ICTS/Unesp, no physical process control is addressed, only data monitoring.

Water meters, flow, and pressure sensors were used to read data on water consumption. The models utilized for this are, respectively, an Itron Unimag Cyble, a YF-B6, and an Ebowan USP-G41 1,2 MPa. The water meter is characterized by several key attributes, including robustness, adaptability, rapid readings, leakage, reverse flux and fraud detection and magnetic fraud detection. Communication with the meter can be achieved through radio walk-by systems, fixed data collection via radio, M-Bus systems, or any other system based on pulse output - for instance an optical sensor, the solution adopted in the presented architecture. The main characteristics of the pressure sensor include a voltages range of 5 to 15 Vdc, a flow rate range of 1 to 30 L/min, an operating temperature of 0 to 80 ºC, and a maximum temperature of 120 ºC for the liquid flowing through it. Additionally, the sensor has a maximum current of 15mA. The communication of the pressure sensor is facilitated by a 0 to 5 Vdc data wire, which is compatible with several microcontrollers. The flow sensor can be easily connected to Arduino, Raspberry Pi or ESP32 and similar boards. This enables the flow sensor to communicate its readings with them via an analog output that varies in the range of 0 to 5 Vdc, which is similar to the pressure sensor. The device is constructed from a carbon steel alloy and is capable of functioning within a pressure range of 0 up to 1.2MPa, with a temperature working range of 0 to 85 ºC.

One meteorological station was installed with the purpose of measuring the ambient variables. It is the Smart Weather Sensor manufactured by G. Lufft Mess- und Regeltechnik GmbH, model WS502-UMB. The device is capable of measuring a range of meteorological variables, including temperature, relative humidity, air pressure, wind direction, wind speed and radiation. For this purpose, it utilizes a range of measurement technologies, including ultrasonic for wind, NTC for temperature, capacitive for relative humidity, MEMS capacitive for air pressure, and Lufft Pyranometer for radiation. The communication interface is a 2-wire RS485 half-duplex, with supported protocols including UMB-Binary, UMB-ASCII, Modbus-RTU, Modbus-ASCII, XDR and SDI-12. The sensor has been constructed incompliant with IEC 61724- 1:2017 Class C. Other specifications that are worthy of mention include wind detection with birdproof construction, a compact all-in-one weather sensor, low power consumption, a heater, an aspirated radiation shield, maintenance-free operation, and an open communication protocol. A CR800 datalogger from Campbell Scientific is employed to initially extract the data from the meteorological station. The datalogger is designed to read data from sensors in its inputs and then transmit them via a communication peripheral.

The photovoltaic (PV) system is equipped with a Huawei solar inverter, model SUN2000- 20KTL-M0. This inverter has a rated power of 20 kW and employs communication protocols including RS485 via the Modbus RTU protocol and via WLAN.

Lastly, energy consumption is measured by a total of 11 Sirax BM1200 meters from Camille Bauer Metrawatt AG. These meters are distributed throughout the campus in order to measure the consumption of different parts and buildings. They are capable of measuring several electrical parameters for single-phase or three-phase systems. The communication infrastructure is based on an RS485 interface, through which the data are sent via the Modbus RTU protocol.

Different microcontroller boards were adopted to request data from the sensors on the reading, processing, and transmitting data layer. In order to read the data from the water consumption and the meteorological variables, the devices defined were ESP32 microcontrollers. The device's main specifications include a single or dual-core 32-bit LX6 microprocessor with a clock frequency of up to 240 MHz; 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM. One ESP32 was installed on each water measuring point, communicating with the physical process sensors via its I/O pins. Similarly, one ESP32 was installed in direct communication with the datalogger, which primarily gathers information from the meteorological station and sends it via serial communication.

To read the data of the inverter connected to the PV system, the selected hardware was a Raspberry Pi W 2, which has the following main characteristics: 1GHz quad-core 64-bit Arm Cortex-A53 CPU; 512MB SDRAM; and 2.4GHz 802.11 b/g/n wireless LAN. The Raspberry Pi utilized has a Modbus to Serial RS-485 converter module to communicate and receive the data requested from the inverter.

The energy consumption monitoring system is the only one currently not running on a microcontroller. It is installed on a notebook with a Python algorithm running on it as a Modbus master to read the data via a Modbus to Serial RS-485 converter. Since all meters installed on campus are located in the measuring booth shown in Figure 4, directly connected to the outlet of the busbars that run throughout the campus, a single long serial communication cable data was previously installed for the data transmission. This cable communicates with all the meters via a single point to the far-away server room, also identified in Figure 4. Hence, this explains the absence of separate measurement devices for this process.

The decision to use a notebook instead of a microcontroller board was made due to the necessity of this specific monitoring system to possess a more trustworthy and long-term memory, as it performs a local backup of each measurement taken. However, in the future, if this issue is solved effectively, it is encouraged to replace the notebook with another microcontroller, such as those utilized in monitoring the other systems, to reduce the overall system building cost. This heterogeneity on devices installed for requesting data from the physical process sensors is due to different people who worked on their implementation and, therefore, different lines of thought. This proves the architecture flexibility, since it does not matter which microcontrollers are defined to work as devices on the reading, processing and transmitting data layer; all of them will have the same functionality and work on the same architecture layer.

The main network adopted was the campus's preexisting WiFi network. Since the ICTS/Unesp is a relatively small campus and all the microcontroller boards are connected to external power sources, the range and high battery usage issues associated with WiFi are not a problem for its utilization. Additionally, considering that the upper layers servers are implemented on computers, which customarily only have access to wired or wireless internet, when using WiFi, it is not necessary to have any gateway to convert data from one network type to another, and there is no need to install any additional network support/technologies, since the campus already has WiFi for all its integrants to utilize the internet. The water consumption device is the only system that sends its data to our own local LoRaWAN gateway. This is done to create a benchmark for comparing the LoRaWAN network performance vs the WiFi network. However, LoRaWAN is not recommended for the other applications because of their stricter time intervals. Both networks utilize the MQTT communication protocol to transmit their data due to its asynchronous nature.

The adopted solution for integrating the data was the Node-RED, a programming tool utilized to wire together hardware devices, Application Programming Interfaces (APIs), and online services, being a web-based flow editor. Node-RED was chosen because it has an easy-to-understand programming environment and a good variety of tools on its constructing blocks palettes. Therefore, it can handle the incoming data processing and write them on the chosen databases efficiently. Additionally, for both databases utilized, because of how the data are formatted prior to arriving at the integration and the way the databases work, it is possible to have a single line in the flow for each database to receive the data coming from any of the processes being monitored and then writing them on the database, contributing even more to the architecture's generalization. Access to both databases is achieved by downloading the respective palettes relative to each of them and configuring these blocks with the specific information relative to the databases, such as the IP address of its server, access port, user, and password.

Two databases are currently being utilized for data storage: InfluxDB and MySQL. The InfluxDB is an excellent choice for monitoring systems because it is a time series database, therefore ideal for working with data with a timestamp, and its bucket infrastructure allows for better database generalization. The MySQL database is utilized because the InfluxDB time series characteristics prevent data from being backed up at later times. Therefore, both databases are utilized in parallel to enhance the architecture reliability and to supplement disadvantages of the other database.

Finally, for the interaction, the Grafana was adopted. It is an open-source software with great flexibility for building visualizations and alerting configurations, and it is also compatible for joint work with many different databases. Some of these are paid and some are not included in the Grafana original installation bundle, needing additional download to its server. However, the ones adopted on our solution are already included, requiring only configuration of the interaction's connections to them, similar to the configuration of database access on the integration layer. Most importantly, they are free to use, another reason for choosing both databases utilized. The built HMI is used as the ICTS/Unesp Smart Campus website, where anyone can gather information on the existing smart campus and consult data on several variables monitored throughout the campus for visualization. Grafana also allows for the external download of the visualized data, and therefore, opting for utilizing it is part of the campus data-sharing policy adopted when implementing the proposed architecture on the ICTS/Unesp. It is important to state that all the components adopted and presented in the architecture must be functional 24/7, since monitoring of the processes cannot be interrupted.

B. DATA READING

This section will address which parameters/data are read in each monitored application and the time intervals defined for reading each data in the different systems included on the presented architecture. Table 1 contains a complete list of all the measured and collected parameters for each system implemented on the architecture, the sensors utilized to read them, and the hardware connected to these sensors. It is important to state that the physical process monitoring systems read more data than what are currently collected by the devices. However, these parameters were defined, for now, as the ones useful for storage. That said, this does not impose any restriction on adding an additional reading for some of these parameters currently not read or to drop the reading of any of these parameters being monitored right now. Another important thing is that not all of these parameters are available on the smart campus website because some are private and have only internal use.

First, we will discuss the water consumption monitoring. Its parameters currently being measured are water flow, pressure, and volume, at the campus entrance and in two additional buildings, D and H, and these data are collected in a 10-minute time interval. The sensors communicate these parameters to ESP32 boards, each measuring spot having its own board.

The next system in question is the meteorological station, located on top of the H building, which is the highest on campus. The main variables being read are air temperature, radiation, air pressure, air relative humidity, wind speed and direction, accumulated rain, and global radiation. Data from it are read every 10 seconds. The meteorological station first transmits its data to an industrial datalogger, which is designed to receive sensor data and transmit them via its peripheral communication modules. The output of this datalogger is connected and sending the data read to an ESP32 board.

The PV generation system is located on top of the E/F buildings, and some of the variables read are, but not limited to, generated energy, grid voltage and current, active and reactive power, internal temperature, and efficiency. Its data are collected on a time interval defined based on the proprietary software of the inverter that runs in parallel with the installed system. This proprietary software realizes its measurements in a 5-minute time interval. Our system, similarly, also does its readings with this interval, utilizing the communication line idle moments to request its own readings. The inverter data are read using a Raspberry Pi W 2 connected directly to the inverter.

Finally, some of the main parameters collected from the energy meters are voltage, current, active, reactive, and apparent power, power factor, energy imported or exported, neutral current, voltage and current total harmonic distortion, among others, being most of them measured on each of the three phases available on the grid. The energy consumption data reading is conducted every minute. At first, it would be made in parallel to the proprietary software which used to run on a server on campus, similar to what is currently done on the PV generation system, but for this application, using two parallel systems would overload the communication line and thus, the proprietary software was deactivated. The data are read via a single communication line, since all the meters are located on a single measuring booth, and the server room with the reading device is located far away from the meters - as seen in Figure 4. Therefore, the data from the meters are read sequentially, and the time taken to read all data from all meters is approximately 31 to 32 seconds, so the reading interval was defined as a minute to leave a margin for adding new meters on the system without having to alter the time characteristics of the collected data, and also because for the intended future applications, which do not involve monitoring electrical events, and therefore a 1-minute time interval is quite satisfactory. All the data are read via Modbus RTU communication by a notebook, which runs as the Modbus master.

Each data point read has several additional fields appended to it by their respective device. These fields characterize each data point read and are a vital part of the data processing, since they allow the data to be correctly stored in the databases. After adding these fields, the data are also converted to JSON format, since this is the expected data format on the integration. The fields that should be included on each sent data are, alongside their data type:

- ID: ID of the device that generated these data - string;
- Value: the value of the measurement itself - double;
- Application: number or name of the application of this measurement - string;
- Location: where on campus is the device that generated these data - string;
- Type: type of the variable or device source - string;
- Variable: which variable is being sent - string;
- Unity: measurement unity used to quantify the variable - string;
- Network: which network is used to send these data - string;
- Professor: name of the professor responsible for the application - string.

These data format was defined on project conception because the first database adopted was the InfluxDB. Since it is a time series database, one can store and integrate all data in a single database bucket - the analog of a SQL table in Influx -, being these fields useful afterwards to filter

TABLE 1. Data read on all monitored systems

Monitored process	Parameters read	Physical process layer sensor devices	Devices reading sensor data
Water consumption	Water flow; water pressure; volume	Itron Unimag Cyble, YF-B6, and Ebowan USP-G41 1,2 MPa	ESP32, communicating via its I/O pins to the sensors
Meteorological station	Air temperature; global radiation; radiation sensor; irradiation sensor; panel internal temperature; atmospheric pressure; relative humidity; contact temperature gauge 1; contact temperature gauge 2; rain gauge; wind direction; wind speed sensor 1; wind speed sensor 2; battery level	Smart Weather Sensor model WS502-UMB	ESP32, communicating via its I/O pins to a CR800 datalogger which primarily reads data from the sensors
PV energy generation	Rated power; maximum active power; maximum apparent power; maximum output reactive power; maximum input reactive power; input power; active power; reactive power; day active power peak; grid voltage; line voltage AB; line voltage BC; line voltage CA; phase A, B, C voltage; grid current; phase A, B, C current; grid frequency; efficiency; internal temperature; accumulated yield energy	Huawei solar inverter, model SUN2000-20KTL-M0	Raspberry PI Zero W2, communicating to the inverter's Modbus port via a Modbus/Serial RS-485 converter
Energy consumption	Volts 1, 2, 3; current 1, 2, 3; active power 1, 2, 3; apparent power 1, 2, 3; reactive power 1, 2, 3; power factor 1, 2, 3; phase angle 1, 2, 3; volts average, sum; current average, sum; active power average, sum; apparent power average, sum; reactive power average, sum; power factor average, sum; phase angle average, sum; frequency; watt-hour import; watt-hour export; Vl 1-2, 2-3, 3-1; voltage total harmonic distortion % 1, 2, 3; current total harmonic distortion % 1, 2, 3; system Voltage THD(%); system Current THD(%); neutral current	Sirax BM1200 Meters	Notebook, communicating to the meter's Modbus port via a Modbus/Serial RS-485 converter

each data required from all other data stored on the bucket. Furthermore, with this data format and the way the integration was designed, the flow for processing and writing data in the InfluxDB was built in a generic fashion, meaning it has a logic that can process any data coming into it without adding any new code into the flux - as long as data are received on this predefined format -, independently from the source generating these data, while still keeping the database well organized.

V. RESULTS AND DISCUSSIONS

A. INTEGRATION RESULTS

The main result of the presented IoT architecture implementation is its functional status with data monitoring and collection, which proves its flexibility and strengths. These results are somewhat subtle to perception, since their visualization is not clear for anyone outside of the development of the architecture or one who does not know about the layers' implementation. The best form of visualizing it is via the interaction developed, i.e., it shows the data being monitored in near real-time and all the existing data history for anyone with access to it. This is an important statement to make to clarify why the interaction details are presented in the results while not being an actual result: the interaction is part of the architecture layers and development, but also its ability to present the data in a human-friendly manner makes it the best option to show how the system was successfully implemented and is functional.

The interaction was implemented as a website dedicated to the ICTS/Unesp smart campus and is available at http://smartcampus.sorocaba.unesp.br/d/WusCo658k9/main-screen?orgId=1&kiosk. The home page contains a picture of the campus entrance, general information on the smart campus functioning and architecture, links to the specific systems' monitored data pages and links to published works relative to the architecture. Besides being important for a clearer data visualization, the website is also relevant to help promote the smart campus in development on the

ICTS/Unesp. This helps not only in the attraction of new students and possible industrial partnerships but also in spreading knowledge on smart campus implementation and its strengths, since this concept is defined by many as the future of university campuses but is little addressed in the Brazilian context, less so in this degree of detailing. Another important contribution derived from the architecture implementation is making most of the collected campus data available to anyone with access to the website.

Each monitored system has its own specific pages, being two for each: a main data page and a historical data page. Their main pages explain the system's general characteristics, its insertion on the architecture, a picture of the physical system itself and some instantaneous values of the main variables currently being read on it, and a link for accessing the historical data. On this historical data webpage, these main variables are presented again, but this time with time series graphics, showing the monitored data behavior throughout specific times and also allowing for a time span filtering so that one visiting the website can change how much data he/she wants to see or change for a specific date. Additionally, Grafana allows for the download of these data in CSV format, allowing one to have these data locally for conducting further studies and deeper analysis.

The water consumption system is important to understand the water usage on campus, identifying, for instance, months where the consumption is bigger or smaller, if the water is coming with enough pressure from the street pipes, which buildings on campus have the most consumption and even highlighting possible leakages when the consumption have an abnormal offset on its levels. While the first one is identifiable by the main water meter, the two latter possibilities can only be fully achieved with meters installed on all campus buildings. This all adds up to financial economies for campus administration and also helps the planet, helping to avoid unnecessary waste and overuse.

The meteorological station data collection brings a lot of informational value rather than economical. This happens because with accumulation of the historical climate data, it is possible to understand climatic behavior on the campus region, which can then be used to better plan the PV energy generation, as well as the use of batteries alongside this system when necessary. When talking about instantaneous data, they provide information about weather characteristics on campus to its students, and it is possible to combine these data with future Heating, Ventilating and Air Conditioning (HVAC) automatic systems if they were to be implemented on the campus.

In addition to that, it is very difficult to find databases with detailed historical climatic data from the region of Sorocaba - the city where the ICTS/Unesp is located - and with these data made available for people outside the campus by the IoT architecture, this opens a vast amount of opportunities for historical climate studies, or how climatic change can be seen on the region, or help on climate prediction studies, among so many other studies, which are outside the scope of a smart campus - and therefore does not necessarily adds value to our database itself -, but brings value to the world/community, as a smart campus is supposed to do.

PV generation data monitoring is important to understand the overall generation patterns. The solar generation pattern is a consensus in the literature, based on the incidence of solar radiation on the PV panels. However, studying a real case that complements the theories is interesting. These data show moments of the day, or week, or even of the year when generation is at its peak - and therefore contributing the most to reduce how much energy is bought from the utility company - or when it is not there at all. Combining these data with the meteorological data registered, one can understand why the generation had a lower value than expected in certain moments of the day and generate more accurate predictions of energy generated in the future based on predictions of how the weather will behave. Also, these data can be used in conjunction with the energy consumption data to study how much energy the campus would need to generate to be self-sustainable in terms of energy, how many new PV generation systems it would need to achieve this status and how batteries would have to be installed on campus in conjunction to the PV generation to make this viable - right now the generation contributes to something in between 5 to 11% of the campus utilized energy per month, approximately.

Lastly, regarding the energy consumption monitoring system, as defined previously, there are 11 meters throughout the campus sectors, and the main ones are located measuring the A, B and C buildings consumption; the D, E, F and G buildings consumption; and the H building consumption. Besides these main meters utilized on the mentioned buildings, other meters are installed in more specific sectors, giving a more granularized view/understanding of this consumption by sector or application. These data can be considered the most interesting for practical applications. The meters installed referring to different campus sectors give a general overview of their energy consumption. Energy consumption monitoring is important to understand how energy is utilized according to where on campus this is being analyzed and how laboratories, or the cantina, or the library, among other university sectors, affect the overall consumption.

These gains are informational, but based on them, it is also possible to point out economical gains, i.e., the identification of where on the network the biggest energy consumption bottlenecks are and planning strategies which can lead to energy savings - such as automatic HVAC control systems, for instance. Also, the consumption history registered can be used to understand how much energy is consumed on average, which can lead to a renegotiation of the current demand contracted with the utility company.

Observing other network energy quality parameters also brings many advantages to the ICTS/Unesp. One is the RMS voltage monitoring, which can show if and when bus voltage extrapolates the acceptable values, being inside a precarious or critical range according to the Brazilian regulation, as defined in [27]. Each moment when this happens must be considered, because taking measurements of the RMS voltage on a 10-minute interval over a week period, when energy is delivered inside the precarious and critical ranges for over 3% and 0.5% of the samples taken, respectively, the utility company must financially compensate its consumers. Another parameters which are very important to analyze when talking about energy quality are the voltage and current total harmonic distortion, which are also being monitored.

The monitoring of energy consumption has been conducted since October 2022. However, the campus administrative sector provided the monthly electricity bills dating back to 2012. The data were then digitalized by transcribing them into a database file. This information allows the integration of the methodology applied for implementing the architecture on the ICTS/Unesp with the historical data from when the architecture was not yet employed on the campus. The monthly energy consumption can be uploaded to our databases to further expand its analysis and it is worth noting that there are several potential alternatives for uploading these data in order to facilitate its utilization. One such approach would be to upload the monthly consumption as a single value for each month to the databases. However, it is essential to consider that in this case these data must be retrieved differently than when retrieving the data gathered after the system was implemented. Another option would be to use statistical tools on the monthly consumption to disaggregate it into equal values for the same minute wise period utilized on the monitoring system's measurement. Additionally, advanced tools can be utilized to extract the current consumption patterns and the monthly consumption from the bills can be used to replicate it daily on the past dates. This allows the total monthly consumption to be integrated and returned.

Although the energy consumption bills are the only ones which have been digitalized and are ready for integration with the newer data, this process of joining historical data to the newly collected data by the architecture implementation can also be expanded to include the water consumption monitoring system, using past monthly water consumption bills. Similarly, historical climate data can be retrieved from climate specialized websites for use in the meteorological station monitoring database.

There are several further possibilities for studying the energy consumption data more deeply. For example, it is possible to apply statistical methods and evaluations, machine learning algorithms, or artificial intelligence on the data for pattern recognition and energy consumption prediction. This latter option is important for generation and energy usage planning. Even without applying such tools, when visually analyzing the generated graphs integrating data - for example, the consumption graphs registered over two weeks, from Monday, October 2^{nd} to Sunday, October 15^{th}, for meters # 4 and 5, the H building's air conditioning systems metering (in kilowatt-hour, kWh) vs the temperature (in degrees Celsius, ºC) in Figure 6 -, one can identify consumption patterns on the data, and how they tend to repeat weekly, or how the system behaves when is supposedly idle - meaning on days where the campus is closed. Besides these patterns, the temperature presentation on the same graphic shows clearly that the air conditioning consumption is directly proportional to the daily temperature, as expected.

Also, both Wednesdays have the biggest weekly consumption - even though on the first one, October 4^{th}, the maximum temperature is smaller than on the two previous days -, pointing out that probably on these days of the week the campus has a bigger concentration of people working and a greater number of classes for this specific semester. Since the number of students, classes being offered, and how classes are scheduled change between semesters, this pattern can change when analyzed after six months. The consumption of the first Thursday and first Friday, October 5^{th} and 6^{th}, respectively, are similar to Wednesday, indicating that Mondays and Tuesdays are "slower" days, with the campus being less crowded. This timespan presented on the graph, in particular, is interesting because October 12^{th} is a national holiday, and because it was on a Thursday, the campus was also closed on Friday. Even on the six days when the campus was closed - the first weekend and after from Thursday to Sunday - there is a pattern in air conditioning usage. This happens because the environmental engineering labs are located in the H building, which, depending on the activities being carried out in the labs, need certain temperature conditions even when there are no people in it. Additionally, each H building's floor has a server room with several computers, which means their air conditioning system must work all the time.

Another example of conjunct application of collected data can be seen in Figure 7, where the PV energy generation (in kilowatt-hour, kWh) is plotted throughout the same two-week span alongside the solar radiation (in watt per square meter, W/m^2) registered on these dates. Both curves have a very similar shape, demonstrating, in a practical way, how these data are directly proportional and how the solar generation panels are fully functional and maintained, since any difference in these curves' shapes could indicate that the PV panel generation potential is deteriorating, the panel may be not properly cleaned or, it was a cloudy day, or one of the systems monitoring these data is facing some problem to measure it properly.

The availability of energy consumption data allow for an investigation of energy savings and cost reduction. Although this analysis is beyond the scope of the paper, it has the potential to improve the outcomes of the transition of the ICTS/Unesp campus to a smart one. The authors intend to further examine this topic in future works. These are only some examples of the applications that can be done with the data being collected with the implementation of the smart campus infrastructure on the ICTS/Unesp, and which can benefit all its members and bring intelligence to the campus. However, a lot more can be done, for example:

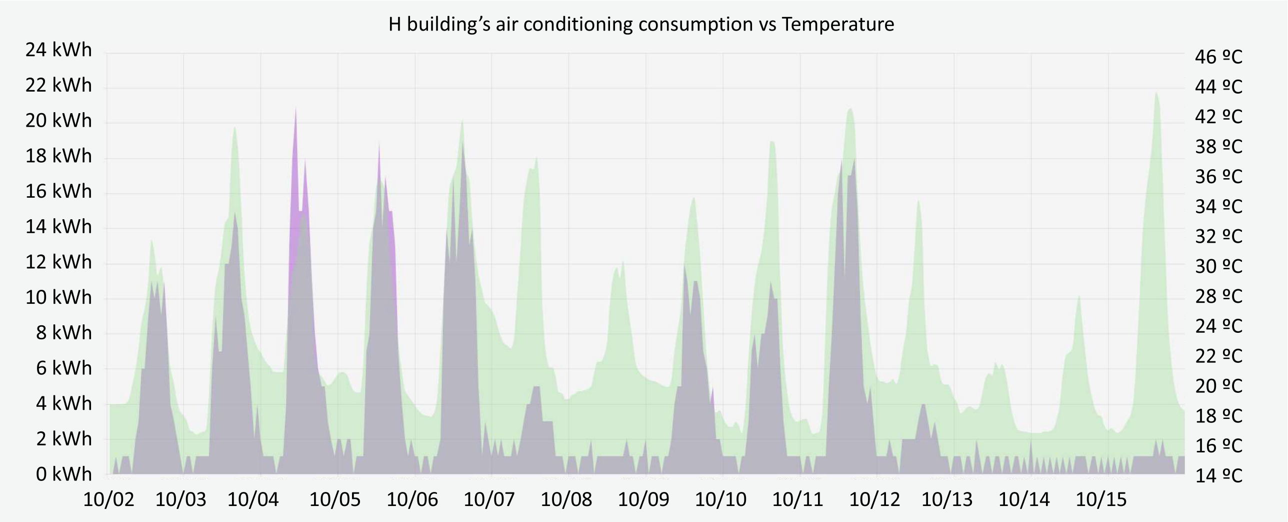


FIGURE 6. H building's air conditioning consumption vs temperature from October 2nd to October 15th.

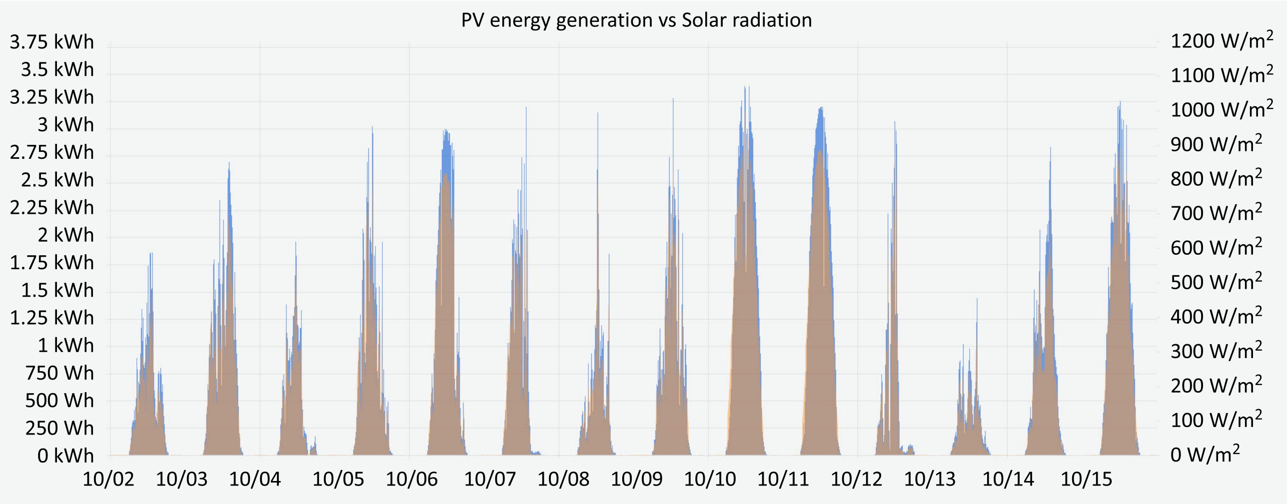


FIGURE 7. PV generation vs solar radiation from October 2nd to October 15th.

- Correlate the campus water consumption with the daily temperature to see if the expected water consumption behavior - days with bigger temperatures have bigger water consumption since people need to drink more water to keep hydrated - is what actually happens;
- Correlate water consumption with energy consumption to see if these data have any relationship between them;
- Study the campus energy efficiency to understand if there are any wastes or unnecessary overuse, which would lead to improvement in how much energy is utilized;
- Installing sensors for monitoring air quality to identify if there is any impact on the campus emissions, if the industry in the vicinity may be affecting the air quality on the campus, or even if for any reason the campus air quality is compromised, and any action is needed to preserve their member's health;
- Installing classroom environment monitoring modules to control the air-conditioning system, avoiding energy waste.

As previously stated, one of the main advantages of our architecture is its incorporation of open-source technologies, which serve as the basis for our implementation of the architecture. Although the current implementation use of open-source solutions is only reflected in the utilization of open-source hardware and software, a GitHub page will be created in the near future to provide open access to each of the processes folders and documentation. This will result in the entirety of the applications being open-source and available as an online example for anyone to access. The GitHub page will be linked to the website built in the interaction layer for an easy and straightforward access.

Regarding the scalability of the implemented architecture, given the limited size of the campus, the monitoring of new processes would only require the addition of new WiFi devices to the network. This is also true in the case of campus facilities being expanded, since WiFi is a scalable network technology, and the addition of new routers or repeaters can properly address this demand. Additionally, a communication network for IoT devices was implemented, which is ready for use and only requires the addition of new devices. Considering the addition of new processes where the real-time functionality is not necessary, LoRaWAN is also scalable, and it has its own local gateway which supports thousands of devices connected and covers all the campus' area. Moreover, with regard to the upper-level layers (i.e., integration, storage and interaction), data center and cloud computing solutions can accommodate any size of campus. Their flexibility allows for adaptation to varying demands and the potential for expansion in operational capacity for storage, visualization and processing. This represents a viable approach for scaling the IT infrastructure implemented in this paper.

B. COMPARISON WITH SIMILAR WORKS ON LITERATURE

In this subsection, we make a link to this paper's second section, namely, the similar works found in the literature. Following the same order presented in the mentioned section, we compare each similar work to what we have presented in ours. The comparison could be made based on several metrics, such as system building cost, complexity, and memory usage. However, since the architecture is still in the implementation phase on the ICTS/Unesp, and due to its specific characteristics, it is challenging to conduct meaningful comparisons. Concerning the issue of costs, not all references provide information on this aspect of their implementation. Furthermore, since the majority of the works [15], [16], [17], [18], [19], [20], [21], [23], [24] have not been implemented in Brazil, the currency aspect also presents a challenge in making comparisons. When related to the complexity, due to the relatively small size of our campus, the complexity for implementation is reduced. For universities that wish to implement this architecture on their campus, larger sizes mean more complex implementations, since adaptations must be made to accommodate the several processes to be monitored and the requisite network coverage. However, this is only reflected in the design phase. Finally, concerning memory usage, no information is provided in any of the works discussed in Section II, therefore, it is impossible to make a comparison. This paper presents an analysis of the methodologies and outcomes described in the references, highlighting how they differ from our approach and how our work aims to contribute more significantly to the state-of-the-art of smart campus architectures.

Before starting the comparisons, it is important to highlight the potential constraints of our proposed architecture. Firstly, it should be noted that the proposed architecture flexibility allows for using paid and more expensive technologies. However, throughout the paper, we encouraged the use of low-cost and open-source options because these technologies facilitate implementation at a reduced cost and with greater ease of replication on any campus, without the need for external companies' support. Nevertheless, it is important to note that while low-cost technologies have significantly evolved and are now widely used, their accuracy still falls short compared to high-budget investments. Additionally, these low-cost solutions typically offer fewer mechanisms to ensure the cybersecurity of the collected and transmitted data. This creates a trade-off between data reading reliability and system-building costs. Furthermore, the reliability of transmitted data is contingent upon the choice and availability of the adopted network. As mentioned earlier, the implementation complexity of the architecture is highly dependent on and proportional to the target campus size. The larger the campus which aims to become smart, the more detailed and extensive the implementation design phase must be to align with the campus's unique characteristics.

The approach presented in [13] is the one that most closely resembles the one conducted here. In it, the authors also proposed an IoT architecture based on a layered infrastructure. They split their proposed architecture into five layers: the Requirements Gathering Layer, the Perception Layer, the Network Layer, the System Layer and the Application Layer. Despite the difference in the number of layers from our approach, how the layers are organized is also different. For instance, in this architecture proposed in [13], the System Layer, encompasses data analysis, data storage, middleware, and control, which are distributed across three different layers in our proposal. We believe that consolidating several systems with different logical functions into a single layer introduces complexity and increases the difficulty of interpreting, adopting and maintaining the system. In contrast, the distinct layers of the proposed architecture facilitate a more straightforward understanding, adoption, and maintenance process for the architecture. Their architecture's testing was implemented utilizing open-source technologies, and its experimental testing was conducted with two systems - temperature monitoring and a mobile application to support user orientation. However, the collected data are not made available for external use. An additional difference which can be pointed out is that the flexibility which the authors mention is related to the capability of utilizing a system within the architecture for more than one function/application, in opposition to what we characterize as flexibility, which is our architecture capability of utilizing any technologies within the layers, as long as they are doing their projected functions. Finally, it is important to highlight that, in [13], the authors do not mention anything about the scalability of their architecture.

Related to work [14], we can say it is also similar to ours. However, in the reference authors work with two objectives - monitor campus variables and show power factor importance for energy quality -, and because of this, their work has a slightly split focus, which culminates in the dashboard construction lacking details for further reproduction. The monitored variables are almost the same ones monitored in our proposed architecture implementation - except for some additional ones e. g., parking lot occupancy. However, despite the authors' efforts to illustrate the integration of the monitored applications within their proposed architecture, they have not provided sufficient detail regarding the construction of these applications. This is likely due to their primary focus on showing the importance of the final dashboard and the potential application of the collected data. The hardware and software utilized differ from those employed in the implementation presented here. Most importantly, the software utilized for processing and storing the collected data in [14] is not open-source. This is a significant distinction, because the use of paid technologies without guidance on lower-cost alternatives makes the replication of the approach challenging, while also increasing its complexity and making it impossible to reproduce without the necessary budget. The authors state that for them, the major integration has to do with visualizing all data in the same dashboard, in opposition to what we propose here, which is that the integration can be seen in dashboards visualization, but it is mainly done in the invisible layers, such as sending data in a predefined pattern, treating and storing them in a centralized database.

Further differences that can be pointed out are their application with data - predicting power factor -, which is not done in the implementation of the architecture presented here but only presented in the form of some suggestions of what can be done and how we aim to highlight the impact the architecture can bring to our campus in the future, since here it is the first step towards becoming a smart campus, thing which the Facens' campus is already one for a while now. Lastly, the mentioned work does not say anything about sharing data for outside studies, which we do with our architecture implementation. The final advantage that can be pointed out is the scalability present in our architecture, since our architecture can be expanded to utilization in campuses of any size, which is something not addressed in the mentioned reference.

The last contribution to the field of smart campus architecture addressed here is presented in [15]. The authors base their architecture on a similar layer structure to that employed in this work. However, despite its interesting approach, the paper's broad scope, which also addresses areas that make up a smart campus and suggestions for applications for monitoring Covid-19, results in a shallow development of the actual architecture. The authors present only a figure and a brief description of the layers without providing guidance on how to implement or replicate the architecture on other campuses. Additionally, it fails to mention anything about flexibility, cost, or scalability of their proposal.

In [16] the authors present an air quality monitoring application as the initial stage in the development of a smart campus, similar to the implementation of the architecture proposed here (i.e., at the ICTS/Unesp). The first difference is the distinct focus of the applications developed. The variables monitored in the reference are related to air quality, a system that is yet to be implemented at the ICTS/Unesp. Additionally, the authors conducted network testing to assess the suitability of the selected communication technology for their campus. This step was not conducted in the present work, due to the discrepancy in campus size. The Zacatenco campus has a perimeter of approximately 3.4 km, which presents a significant challenge in terms of network coverage. In contrast, ICTS/Unesp campus is relatively smaller in size and does not face the same difficulties in this regard. In addition to these differences, the most significant difference can be found in the scope and context of the implementation. While the project in [16] aims to initiate the development of a smart campus, it places greater emphasis on the implementation of a specific application, without adequately contextualizing its contribution to the advancement of the smart campus. Furthermore, the underlying architectural framework that supports its implementation is not adequately addressed.

In a similar vein, the authors in [17], [18], [19], and [20] present different applications for IoT in the context of a smart campus. With the exception of [20], in which the authors develop the monitoring of similar variables to the architecture implemented on the ICTS/Unesp (such as temperature, humidity, and rain). The applications differ from those implemented at the ICTS/Unesp, with a primary focus on the educational aspects of the campus. Nevertheless, as previously commented about the [16], the papers are primarily focused on developing smart campus applications, rather than on presenting an architecture which serves as the basis for these applications and for the smart campus implementation, which is the main objective of the present work. Additionally, [17], [18], [19], [20] either show no significant outcomes or provides minimal to no in-depth analysis of the presented subjects, methodology, and hardware or software utilized. This lack of detail is a significant issue, particularly concerning the replicability of the methods described in the previously mentioned papers.

The main idea and development on [21] are similar to what we propose here, mainly the implementation of our architecture: utilizing IoT to monitor data in a smart campus context, explaining in detail how the data are sent, the network utilized, server and data storing functioning, demonstrating the system's working status. In [21] the authors do not propose a general architecture for implementing a smart campus. Instead, they indicate how to implement a framework specific to health monitoring applications. Although the overall framework can probably be extended to characterize a smart campus architecture for other applications, there is not enough information or detailing on how to expand it to include other campus systems, given the specialized nature of the medical application, which uses a specific data format and communication protocols, exhibiting notable differences.

The main implementation differences can be found firstly in how the data are sent and the communication protocol adopted. Another difference lies on the papers' focus, since in [21], the primary objective is to improve campus health within the context of a smart campus, rather than to propose an infrastructure for a smart campus. This health-centric approach is quite different from what is typically found in the literature and from the implementation of our architecture proposed in this paper.

The main relevance of [22] lies in its very well-structured IoT infrastructure, which has an architecture that resembles the one we present in our paper and giving a detailed step-by-step of its construction, layers and functional requirements. In [22], the focus is solely on PV generation and meteorological data monitoring. Similarly to the previous approaches, the architecture is designed for specific applications and does not address smart campus applications. Adaptations would be required If the architecture were to be applied to a smart campus. Another structural difference worth mentioning is that the authors of the reference build the sensors for reading data from scratch, in opposition to using industrial ones, as it is more common in literature. Finally, the architecture is also flexible, but only related to adding different rated PV generation systems, instead of having availability to add different systems, such as energy and water consumption. In the paper context, it does make sense, but for smart campus applications, it is a crucial limitation.

In [23] and [24] the authors focus on the implementation of different power consumption monitoring systems. However, their approaches diverge from the one presented in this paper, primarily due to the lack of emphasis on smart campus applications. In [23] the authors focus on microgrids installed in universities, which represent a natural evolution of the energy aspect of a smart campus. The paper concentrates more on how the monitoring process can enhance the thermal recovery for CHPs, provide energy savings and offer warnings for preventive maintenance. The emphasis on test cases and results in [23] precludes a comprehensive presentation of the construction of the monitoring and control infrastructure. Consequently, the approach is more application-oriented than the architecture developed here. A parallel can be made between the power monitoring system for smart grids assisted by IoT presented in [24] and the energy consumption monitoring system implemented at the ICTS/Unesp (i.e., the approach presented in this paper). However, in [24], a substantially smaller number of variables are read. Moreover, the monitoring system implemented is more focused on an application aspect than on an architecture to support further applications. Additionally, there is no focus on smart campus applications.

The review of the literature reveals that the majority of similar works do not directly propose or address smart campus architectures. When they do, they often fail to sufficiently delineate the proposed architecture details, thereby rendering interpretation and reproduction more challenging. Additionally, the applications developed, whether within the architectures or not, also tend to lack sufficient details on their implementation or integration into smart campus architectures. Additionally, these works do not explicitly indicate whether their proposals are compatible with open-source technologies. The only exception that encompasses all these characteristics is the work in [13], but the layer definitions are more complex than those proposed in the architecture presented in this paper. Encompassing these characteristics alone is already a differential from our proposed architecture, but alongside having all these characteristics, this work's proposed IoT architecture also has a huge differential because our implementation of the architecture also offers full access to most of the data being monitored. As far as this work's authors understanding, there is a scarce number of campuses or even companies willing to give access to their monitored data, and when doing so, it involves several bureaucratic actions and giving access to only part of the data.

C. ARCHITECTURE REPLICABILITY

The previous sections of this paper fulfill their goal of presenting the architecture and a case study of its implementation to aid in replicating it in other university campuses' contexts. Nevertheless, for full replicability, some additional comments and suggestions are in order, as follows:

- The first important step is understanding how each physical process generates and communicates its data To achieve this, it is necessary to thoroughly study each application specific manual in order to figure out how data are generated and stored, which data are generated by the system in question, how the actuators work, besides which communication protocols are utilized by the physical system monitoring equipment, and the memory addresses mapping to correctly acquire the data.
- With this previous understanding in mind, the device selection involves analyzing the optimal choice between the vast microcontroller options. For example, if the code requiring data needs some extra complexity and/or additional libraries, the best option would be a microcontroller with bigger processing power, and the best programming language option is Python. Therefore, it is not viable to utilize boards such as ESP32, which has a smaller processing capacity and can only run Arduino codes, a simpler and more limited language. Also, it has to be taken into consideration if the selected board has ways of communicating with the physical process it will monitor, directly or utilizing some protocol converter in between them. Lastly, relating to the device, it is important to understand if it has ways of connecting directly - wirelessly or via cable - or if it needs additional shields to connect to the selected network.
- Related to network selection, it is important to evaluate the campus size. The ICTS/Unesp is relatively small, so it is possible to utilize WiFi as the network without any problems, but it is not viable on bigger campuses.

A suggestion for utilizing WiFi on a larger campus is building small "WiFi islands", i.e., having WiFi networks for each campus sector or building, in a way that applications would only share the network with other nearby applications. This would only work if the applications had any way of being powered by outlets or other power sources, since WiFi utilizes more energy and battery-powered boards could have their battery lifetime decreased faster by utilizing WiFi. According to the campus size, it is important to analyze which communication protocol fits it better. Other viable options for wireless communication are LoRaWAN - as also utilized in ICTS/Unesp -, Bluetooth, Zigbee, and Xbee, among others, each presenting its own advantages and disadvantages.

- In addition, when choosing the components for adapting the architecture to its own campus, it is essential to consider whether the data that will be generated has the potential to be characterized as Big Data. When this is a possibility, it is advisable to build the architecture from scratch with the support for Big Data generation, since the overall infrastructure necessary for this differs significantly from that for Small Data, mainly in the data storage layer. Furthermore, a transition from Small Data to Big Data technologies can be challenging, since it is not a straightforward process. In our implementation of the architecture, no support for Big Data was implemented, as it was not considered that the campus would generate it. However, this may change in the future, and further studies on this topic are planned for future works.
- Another important issue not pointed out throughout our work is that if the university campus has a Hospital on its facilities, such as Unicamp, it is vital to thoroughly study energy supply reliability and backup generators, since health applications are critical and cannot stay without power for any period of time. Therefore, it is important that when replicating our architecture, one designs its own ways of ensuring this reliability of power supply and extra generation control, since this would be the most critical system monitored/controlled.
- It is necessary to study how the utilized systems support themselves, meaning, the selected integration must have support to write on the selected database, and the interaction also must have support to read data from the selected data storage technology and to send any necessary commands to the chosen integration, and the devices must have support to communicate with the physical processes. The devices, integration and interaction must also have support to communicate via the determined network.
- Lastly, the architecture is scalable, and although it was initially implemented on a small campus, it can be readily replicated on a larger campus without encountering any significant challenges. For this, if the campus size is sufficiently large to impact the network range, it is advisable to either adopt a communication network that better aligns with the data communication and network range requirements or to configure the network by buildings or campus sectors to enhance its maintenance and facilitate a more comprehensive understanding. With regard to the database and interaction layers, it is recommended that cloud-based services be used in such case. However, the overall system architecture remains the same. Therefore, except for the communication network utilized, the architecture only needs to be expanded with additional monitoring devices and computing power for the servers running it to be implemented following the same guidance provided here.

VI. CONCLUSION

In this work, an IoT architecture was proposed and implemented on the developing ICTS/Unesp smart campus. It was presented as a unified platform for monitoring and integrating several data relevant to the smart campus fields of action and controlling any number of these processes.

The main contribution of this work is the detailed proposal of a layered, scalable, flexible, and low-cost architecture for a smart campus implementation based on IoT technologies. The proposed architecture can be replicated on other campuses on the path to becoming smart, fully based on open-source technologies, and its implementation on a Brazilian university campus also offers open access to most of the data currently being monitored. Even though some of these contributions were already made by different authors in the literature, joining all of them is, in the authors of this paper understanding, something that was not done to this date, mainly the monitored data availability.

The implementation of the architecture on the ICTS/Unesp campus was made based on academic works, and all the works utilized in its development can be found on the smart campus website, which is available in Portuguese and English. However, the mentioned works are mostly available in Portuguese. After several months of the architecture running, it has been demonstrated that the proposed architecture achieved its goal of delivering a reliable smart campus backbone system. This system provides continuous and almost real-time data monitoring, presents the data straightforwardly and seamlessly integrates data from different sources, processes, and hardware. The final built architecture brings informational and economic value to the campus management and all its members.

The next steps for this work's continuation are to implement the architecture on campuses of different sizes and utilize different technologies when doing so, thereby strengthening the evidence of this architecture's generalization and scalability. A further expansion of the studies presented in this paper is the continuation of the smart campus implementation on the ICTS/Unesp, focusing on the potential benefits of the proposed architecture for other structural pillars, in addition to the current focus on energy, water and environmental pillars. Although the architecture does not require the implementation of improvements in all campus areas to be considered a smart campus, establishing a comprehensive smart campus infrastructure on a single campus can yield numerous benefits for the institution and its constituents. When fully achieved, this infrastructure can serve as a model for other campuses seeking to become more intelligent and efficient. Another possible further development of this work is to expand the analysis of the benefits brought to the campus by the architecture implementation, such as analyzing the savings and cost reductions provided by the energy consumption monitoring system, a future work which is in the authors' plans.

REFERENCES

[1] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, "Smart cities and the future internet: Towards cooperation frameworks for open innovation," in *The Future Internet*, Berlin, Germany: Springer, 2011, pp. 431–446.

[2] *World Urbanization Prospects: The 2018 Revision*, United Nations, New York, NY, USA, 2018.

[3] *Smart Cities and Infrastructure*, United Nations, Geneva, Switzerland, 2016.

[4] S. Fortes, J. A. Santoyo-Ramón, D. Palacios, E. Baena, R. Mora-García, M. Medina, P. Mora, and R. Barco, "The campus as a smart city: University of Málaga environmental, learning, and research approaches," *Sensors*, vol. 19, no. 6, p. 1349, Mar. 2019.

[5] P. Fraga-Lamas, M. Celaya-Echarri, P. Lopez-Iturri, L. Castedo, L. Azpilicueta, E. Aguirre, M. Suárez-Albela, F. Falcone, and T. M. Fernández-Caramés, "Design and experimental validation of a LoRaWAN fog computing based architecture for IoT enabled smart campus applications," *Sensors*, vol. 19, no. 15, p. 3287, Jul. 2019.

[6] Z. Y. Dong, Y. Zhang, C. Yip, S. Swift, and K. Beswick, "Smart campus: Definition, framework, technologies, and services," *IET Smart Cities*, vol. 2, no. 1, pp. 43–54, Mar. 2020.

[7] A. Alghamdi and S. Shetty, "Survey toward a smart campus using the Internet of Things," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2016, pp. 235–239.

[8] W. Hidayat, M. Hendayun, A. S. Sastrosubroto, R. Hidayat, and S. Haris, "Developing smart campus readiness instrument based on Pagliaro's smart campus model and smart city council's readiness framework," *J. Physics: Conf. Ser.*, vol. 1783, no. 1, Feb. 2021, Art. no. 012051.

[9] E. S. Abuelyaman, "Making a smart campus in Saudi Arabia," *Educause Quart.*, vol. 31, no. 2, pp. 10–12, 2008.

[10] K. Chakal, M. Korkiakoski, H. Mehmood, T. Anagnostopoulos, P. Alavesa, and P. Kostakos, "Augmented reality integration for real-time security and maintenance in IoT-enabled smart campuses," in *Proc. IEEE 31st Int. Conf. Netw. Protocols (ICNP)*, Oct. 2023, pp. 1–6.

[11] C. H. Li, S. L. Mak, C. C. Lee, T. T. Lee, N. H. Y. Yuen, and W. F. Tang, "A review of 5G building management technologies and applications in smart campus," in *Proc. IEEE 21st Int. Conf. Ind. Informat. (INDIN)*, Jul. 2023, pp. 1–5.

[12] J. W. P. Ng, N. Azarmi, M. Leida, F. Saffre, A. Afzal, and P. D. Yoo, "The intelligent campus (iCampus): End-to-end learning lifecycle of a knowledge ecosystem," in *Proc. 6th Int. Conf. Intell. Environments*, Jul. 2010, pp. 332–337.

[13] B. S. Brand, S. J. Rigo, R. M. Figueiredo, and J. L. V. Barbosa, "Sapientia: A smart campus model to promote device and application flexibility," *Adv. Comput. Intell.*, vol. 2, no. 1, p. 18, Feb. 2022.

[14] R. S. Netto, L. A. Faria, J. V. Lochter, E. R. Filho, and G. A. A. Rodrigues, "Facens smart campus integrated dashboard: A use case applied for energy efficiency," in *IoT and WSN based Smart Cities: A Machine Learning Perspective*. Cham, Switzerland: Springer, 2022, pp. 67-88.

[15] P. Agarwal, R. Kumar G. V. V., and P. Agarwal, "IoT based framework for smart campus: COVID-19 readiness," in *Proc. 4th World Conf. Smart Trends Syst., Secur. Sustainability (WorldS4)*, Jul. 2020, pp. 539–542.

[16] M. S. Meraz, M. G. J. Gutiérrez, and I. M. Juárez, "Air quality monitoring in a smart campus," in *Proc. IEEE Int. Smart Cities Conf. (ISC)*, Sep. 2020, pp. 1–7.

[17] A. Zhamanov, Z. Sakhiyeva, R. Suliyev, and Z. Kaldykulova, "IoT smart campus review and implementation of IoT applications into education process of university," in *Proc. 13th Int. Conf. Electron., Comput. Comput. (ICECCO)*, Nov. 2017, pp. 1–4.

[18] Z. Liu and C. Han, "An IoT-based smart campus monitoring system," in *Proc. Int. Conf. Adv. Learn. Technol. (ICALT)*, Jul. 2022, pp. 395–396.

[19] M. W. Sari, P. W. Ciptadi, and R. H. Hardyanto, "Study of smart campus development using Internet of Things technology," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 190, Apr. 2017, Art. no. 012032.

[20] N. P. Sastra and D. M. Wiharta, "Environmental monitoring as an IoT application in building smart campus of universitas udayana," in *Proc. Int. Conf. Smart Green Technol. Electr. Inf. Syst. (ICSGTEIS)*, Oct. 2016, pp. 85–88.

[21] Y. Liang and Z. Chen, "Intelligent and real-time data acquisition for medical monitoring in smart campus," *IEEE Access*, vol. 6, pp. 74836–74846, 2018.

[22] G. C. G. D. Melo, I. C. Torres, Ì. B. Q. D. Araújo, D. B. Brito, and E. D. A. Barboza, "A low-cost IoT system for real-time monitoring of climatic variables and photovoltaic generation for smart grid application," *Sensors*, vol. 21, no. 9, p. 3293, May 2021.

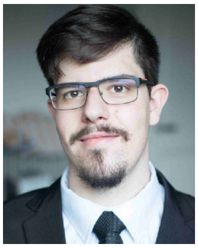
[23] K. R. Khan, M. S. Siddiqui, Y. A. Saawy, N. Islam, and A. Rahman, "Condition monitoring of a campus microgrid elements using smart sensors," *Proc. Comput. Sci.*, vol. 163, pp. 109–116, Jan. 2019.

[24] F. Khan, M. A. B. Siddiqui, A. U. Rehman, J. Khan, M. T. S. A. Asad, and A. Asad, "IoT based power monitoring system for smart grid applications," in *Proc. Int. Conf. Eng. Emerg. Technol. (ICEET)*, Feb. 2020, pp. 1–5.

[25] Facens. (Nov. 2018). *Institucional—Facens*. Accessed: Dec. 6, 2023. [Online]. Available: https://facens.br/a-facens/institucional/

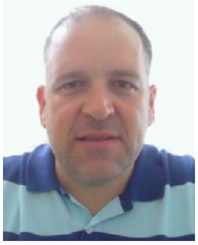
[26] (2022). *Unicamp, Conheça O Campus*. Accessed: Dec. 6, 2023. [Online]. Available: https://www.comvest.unicamp.br/ingresso-2022/vestibular-2022/manual-de-ingresso-2022/conheca-o-campus/

[27] Agência Nacional de Energia Elétrica (ANEEL), *Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional (PRODIST), Módulo 8—Qualidade da Energia Elétrica*, (in Portuguese), Brasília, Brazil, 2020.



BRUNO OLIVEIRA ZARPELLON received the B.S. degree in automation engineering from FACENS, Sorocaba, in 2020, and the M.Sc. degree in electrical engineering from São Paulo State University (UNESP), Sorocaba, São Paulo, in 2024. He did exchange studies with the University of Wisconsin–Madison, in 2015 and 2016, and did an internship exchange with the Fraunhofer Institute for Production Technology, Aachen, in 2018 and 2019. His research interests include smart campuses, the Internet of Things, PCB design, and smart metering devices.

LUIS DE ORO ARENAS received the B.S. degree in electronic engineering from the National University of Colombia, Bogotá, in 2012, and the M.Sc. and Ph.D. degrees in electrical engineering from São Paulo State University (UNESP), Sorocaba, in 2014 and 2019, respectively. He conducted postdoctoral studies with the Power Electronics Laboratory, UNESP/FEIS, in 2019, and the Institute of Science and Technology, UNESP, from 2020 to 2022. He is currently an Assistant Professor with ICTS, UNESP, affiliated with the Department of Environmental Engineering. His research interests include electronic instrumentation and data science applied to environmental systems, power electronics, power quality, and digital and analog signal processing.



EDUARDO PACIÊNCIA GODOY received the B.Eng. degree in control and automation engineering from the Federal University of Itajubá, Minas Gerais, Brazil, in 2003, and the M.Sc. and Ph.D. degrees in mechanical engineering from the University of São Paulo, São Paulo, Brazil, in 2007 and 2011, respectively. Currently, he is an Associate Professor with São Paulo State University (UNESP), São Paulo. His research interests include automation and networked systems, wireless networks and telemetry, the Internet of Things, and Industry 4.0.

FERNANDO PINHABEL MARAFÃO (Member, IEEE) received the B.S. degree in electrical engineering from São Paulo State University (UNESP), Brazil, in 1998, and the M.Sc. and Ph.D. degrees from the University of Campinas (UNICAMP), Brazil, in 2000 and 2004, respectively. In 2002, he joined the Power Electronics Group, University of Padua, Italy, as a Visiting Student. In 2013, he joined Colorado School of Mines, Golden, CO, USA, as a Visiting Scholar on autonomous and intelligent distributed energy systems. Since 2005, he has been an Associate Professor with the Group of Automation and Integrating Systems, UNESP. His current research interests include smart grid technologies, renewable energies, energy management, and power theories. He is a member of Brazilian Power Electronics Society (SOBRAEP) and Brazilian Automation Society (SBA).

HELMO K. MORALES PAREDES (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from San Agustin National University Arequipa, Arequipa, Peru, in 2002, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Campinas, Campinas, Brazil, in 2006 and 2011, respectively. In 2009, he joined the Power Electronics Group, University of Padova, Italy, as a Visiting Student. In 2014, he joined the PEMC Group, University of Nottingham, U.K., as a Visiting Scholar. In 2018, he joined the ACEPS Group, Colorado School of Mines, Golden, CO, USA, as a Visiting Scholar. Since December 2011, he has been with São Paulo State University (UNESP), Sorocaba, Brazil, where he is currently an Associate Professor and the Leader of the Group of Automation and Integrating Systems (GASI). His current research interests include power quality, power theories, harmonics propagation, power electronics applied to renewable energy systems, and microgrid control. He is a member of Brazilian Power Electronics Society (SOBRAEP) and Brazilian Automation Society (SBA). He received the Prize Paper Award from IEEE TRANSACTIONS ON POWER ELECTRONICS, in 2012, and two consecutive Best Paper Awards from Brazilian Power Quality Society (SBQEE), in 2021 and 2023.

• • •