RevoCampus: a Distributed Open Source and Low-cost Smart Campus

Olivier Debauche*
Faculty of Engineering - ILIA / Infortech
University of Mons
Mons, Belgium
BioDynE / TERRA
GxABT - ULiège
Gembloux, Belgium
Orcid: 0000-0003-4711-2694

Rachida Ait abdelouahid*
Faculty of Sciences Ben M'Sik
University Hassan II
Casablanca Morocco
Orcid: 0000-0002-9582-7988

Saïd Mahmoudi
Faculty of Engineering - ILIA / Infortech
University of Mons
Mons, Belgium
Orcid: 0000-0001-8272-9425

Yahya Moussaoui
EILCO Student
University of Coastal Opal Coast
Calais, France
Orcid: 0000-0002-4585-0370

Abdelaziz Marzak
Faculty of Sciences Ben M'Sik
University Hassan II
Casablanca, Morocco
Orcid: 0000-0002-6314-1498

Pierre Manneback
Faculty of Engineering - ILIA / Infortech
University of Mons
Mons, Belgium
Orcid: 0000-0003-3990-3621

***Abstract*—Smart Campus can be assimilated to small smart cities in which learning experience and living conditions are improved by smart environment and IoT concepts. In this paper, we present (R)evoCampus our Smart Campus solution based diverted smart Home interoperable protocol platforms, micro controllers ESP32, low-cost sensors. This architecture uses at same time the principles of IoT, smart environment technologies, and smart city concepts to develop an effective use of the resources, and to improve the quality of life inside the whole University. In the proposed solution, Wi-Fi protocol is used for communication in indoor while outdoor communications are ensured by LoRaWAN protocol.**

***Index Terms*—Smart Campus, Smart City, Smart Environment, Internet of Things**

I. Introduction

Smart Campus usually refers to buildings, ground, and places where university is located. Emergence of smart environment technologies and the presence of digital native equipped lively student community have enabled the development of smart campus to improve experience of studying, sharing learning contexts, in time and space [1].

The main technologies supporting the smart campus include Cloud computing and Mobile Edge Computing, Internet of Things, Augmented Reality, Artificial Intelligence and Machine Learning [2] [3].

Abuarquoud et al. [4] have identified important benefits which can be obtained in a smart campus. Among this one, we keep up: (1) The promotion smart energy, water and waste management thought IoT-based services; (2) The monitoring of electrical device to notify automatically maintenance team to improve response time to operation; (3) The detection of non-authorized people in an area or opened windows and doors to prevent intruders; (4) The assistance of students and staff to find a parking place; (5) The automation of students attendance.

Designing a Smart Campus asks to tackle some challenges: social, economical, juridical and technical [5]. In a Smart campus nodes enter and exit dynamically the system. Moreover, nodes can suffer fatal failures, new nodes are continually added and old nodes are deleted or replaced.

Prandi et al. [1] have proposed a categorization of smart campus in three mains approaches: The first group, technology driven, exploits IoT services providers and Cloud Computing to transform a university into an intelligent campus environment. The second group adopts smart city concept considering smart campus as a small and self-contained city where users play a key and an active role int the crowdsourcing and/or crowdsensing. The third based on the development of an organization or business process collect data about environmental aspect to provide services, reduce costs and improve the quality of life inside and outside the campus.

The aim of a (R)evoCampus is to improve the everyday life for the University while decreasing the ecological footprint of buildings sent by reducing the costs of functioning (water, electricity, waste, etc.). This campus is digital, sustainable and intelligent combining innovative teaching equipment, a wide range of sensors, systems of communications, location, storage, and simulation. It applies the three approaches described below in using IoT, smart environment, crowdsourcing and crowdsensing to develop an effective use of the resources of the whole university, and improve quality of life.

Smart Home Automation systems such as Home Assistant or openHAB have been developed to interconnect large wide amount of sensors using different network protocols. Home

Assistant is an open source automation solution hosted on a simple Raspberry Pi or a similar computer. It allows to survey the state and control numerous devices from a simple and user-friendly interface which respect privacy and don't publish any information on the cloud. This platform can be diverted to be implemented in each auditorium, office, laboratory, room of the University. These platforms can publish and subscribe to MQTT topics to exchange between them and transfer data to a central point located in the building.

ESPHome is a system allowing to control and manage ESP32 by simple yet powerful configuration files and control them remotely through Home Automation systems like Home Assistant. ESP32 code can also be updated through ESPHome in over-the-air programming (OTA).

II. Literature review

This section is divided into a background part describing our previous papers and their respective contributions and a second part related to the existing state of the art in this field with major contributions on smart campus.

A. Background

In our previous works, we have progressively developed a cloud architecture [6] through use cases of cattle and farm animals' behavior [7] [8] [9] [10] and digital phenotyping [11] [12]. Afterwards, we have developed of data transmission with use cases such as the health of bee hives [13], connected pivot-center irrigation [14], landslides monitoring [15]. Recently, we have completed our central cloud architecture with a edge AI - IoT architecture [16] and tested this architecture on various uses cases such as smart poultry [17], Patients and Elderly Monitoring [18], Bird Nesting [19]

We have developed practical works protocols for smart home [20], Smart cities [21] and IoT demonstrators [22] applying technologies used to built our Smart campus.

B. Related Works

The Literature contains a tremendous amount of Smart Campus initiatives that address wide range of challenges in the Smart Campus such as energy optimization, Smart Parking, services, mobility, etc. Alonso et al. use motion sensors (accelerometer, gyroscope), position sensors (gps, magnetometer), and environmental sensors (light, ambient temperature, relative humidity) of smartphone stored locally in a SQLite database, and afterwards sent to Ubidot platform [23].

Ward demonstrated that it is possible to use retired smartphone to develop applications such as the availability of lab workspaces, detect and monitor Wi-Fi signal in study areas, follow current location of public transport [24].

neOCampus is a demonstrator of connected, innovative, intelligent and sustainable campus based on a bottom up distributed approach where each device acts as an agent and take decisions. It uses open hardware and software to facilitate implementation of communications. Data is processed locally by machine learning algorithms able to adapt him to the ever-changing data, learn non-linear patterns, and perform with contextual data [5].

Prandi et al have proposed an architecture based on 3 layers : (1) The sensors layer collects and validates data by comparison with different data sources, and make sensors data available thought open-data repositories; (2) The database Layer stores sensed data in MySQL and use ckan an open-source DMS to allows the interaction with open data; (3) The data visualization layer is composed on one hand of a rich web-based application which allow the interaction with the hyperlocal data and on the other hand a log management web interface that enables to perform analysis and visualize data [1].

Fraga-Lamas et al. designed a LoRaWAN Fog Computing Architecture based on Raspberry Pi using Wi-Fi in indoor and LoRaWAn in outdoor for Smart Campus applications [25].

Liu et al. proposed a WiCloud platform built around several MEC servers and using Network Functions Virtualization (NFV), Software-Defined Network (SDN) [3].

III. Our Proposition

Smart Campus generate a large amount of heterogeneous data (numerical or not) of type continuous, discrete, multi-dimensional or not [5] which must be collected and treated. WiFi is widely available inside of university building, it is easy to connect microcontrollers equiped of sensors on the WiFi network. In order to obtain autonomous units taking local decision, we have implemented a home assistant in each room, auditorium and laboratory which collect and process data produced by a set of ESP32 equiped of sensors managed by ESPHome an add-on for Home Assistant. Each micro controller ESP32 is connected on Wi-Fi and support Over-the-air programming (OTA). The Home Assistant installed on an Raspberry Pi 4 process data and afterwards send them to a central point in the building by means of MQTT protocol.

A. Campus interoperability Architecture

The Fig. 1 presents our IoT interoperability architecture apply to this project of smart campus. This project implements the generic IoT interoperability architecture proposed by Ait Abdelouali d et al. in 2018 [26] and composed of 7 layers:

- **Infrastructure Layer** represent different connected things which acquire environmental information or achieve an action on this environment.
- **Information Layer** allows to communicate data between different connected things by means of microcontrollers. It allows to discover and identify connected objects via microcontroller.
- **Communication Layer** allows the transmission of data between connected things by means of various communications protocols and technologies such as MQTT, Wi-Fi, Z-Wave, NFC, Zigbee, 4G, Bluetooth, etc.
- **Connectivity Layer** ensures the connectivity and interoperability of exchanged data between connected objects thanks to IoT platforms which are for example home automation solutions. These platforms contains services which can store, correlate, analyze, and exploit data. In

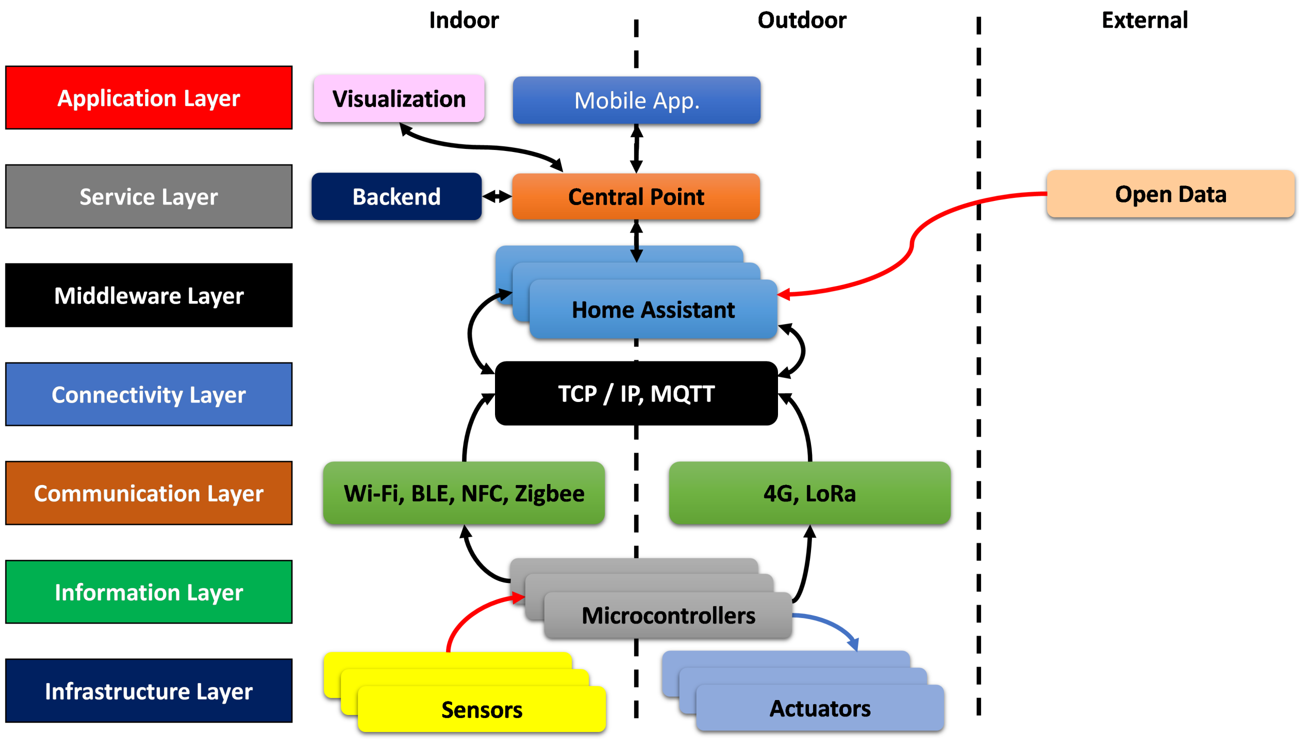


Fig. 1: Smart Campus Interoperability Architecture

our use case of smart campus, we have chosen to use Home Assistant as IoT interoperability platform.

- **Middelware Layer** ensures collect data in the cloud, data processing ETL (Data cleaning, integrity verification, normalization, data sorting and transformation), data interpretation with machine learning algorithms, and finally storage in Big Data.
- **Service Layer** provides critical and reliable services for various applications i.e.: weather station, open data, University backend.
- **Application Layer** proposes different manners of data presentation, visualization, and consultation to end-users.

B. Material

The material is composed of four packages. The first package contains the microcomputer hosting applications which manages the sensor network. The second package integrates microcontrollers used to sense environment and operate actuators. The third collects different sensors used in this application. Finally, the fourth package gathers actuators.

a) Micro computer: is used to host software need to manage and monitor the network of sensors.

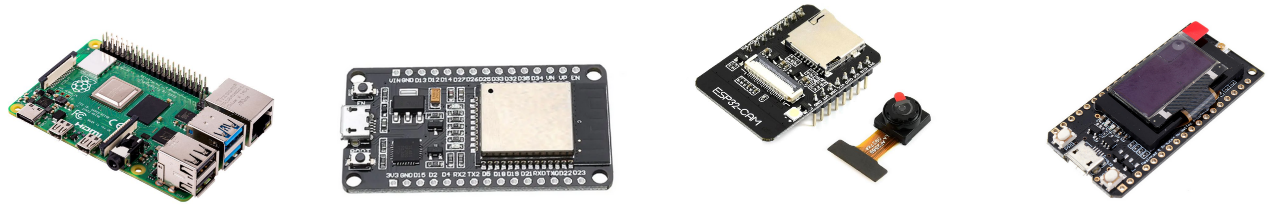
The Raspberry Pi 4 is used to host Home Assistant which can take local decision and tun on/off actuator. It also plays the role of local gateway and transfer data to central point of the building with MQTT protocol. The Raspberry Pi 4 is powered by a Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz with 2GB LPDDR4-3200 SDRAM, a 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE, Gigabit Ethernet, and a Micro-SD card slot for loading operating system and data storage. (See Fig. 2a)

b) Micro controllers: are used to connect sensors and transmit data.

The ESP32-Wroom-32 is equiped with a Wi-Fi and a Bluetooth interfaces that allows it to communicate with the local gateway configurated as Access Point. We use Arduino IDE to program it in the same way as an Arduino UNO. ESP-Wroom-32 contains a Xtensa dual-core 32-bit LX6 microprocessor at 240 MHz, 520 KiB SRAM, 4 MiB Flash Memory. Moreover it provides 12-bit SAR ADC up to 18 channels, 2 DAC of 8-bit, 10 GPIO, 4 Serial Peripherical Interface (SPI), 2 Inter-IC Sound (I^2S), 2 Inter-integrated Circuit (I^2C). It is used to connect sensors inside buildings (Fig. 2b).

The ESP32-CAM AI-Thinker is equiped of an ESP32-S chip, an OV2640 camera, microSD card slot and several GPIOs to connect peripherals (Fig. 2c).

ESP32 Lora is equipped in addition with an Oled 0.96” display and a Semtech chip to communicate in LoRa with a gateway and transmit data on the things network (TTN) (Fig. 2d).



(a) Raspberry Pi 4 (b) ESP32 Wroom 32 (c) EP32-CAM (d) ESP32-Lora

Fig. 2: Computer and Microcontrollers

(a) Air Temperature & Humdity Sensor (b) Light Intensity (c) PIR Motion Sensor

(d) CO_2 Sensor (e) Contact Switch (f) Extremely accurate Real Time Clock

Fig. 3: Environmental Sensor Node

c) Auditorium Sensors: measures environmental parameters.

The AM2315 a I^2C Sensor measures the Air temperature with an accuracy of ±0.1°C between -20°C to 80°C and the humidity with an accuracy of 2% between 0 to 100% (Fig. 3a).

The TSL2591 (Adafruit Industries LLC) is a high dynamic range digital light sensor which can detect light ranges from up to 188 μLux up to 88,000 Lux (lumens per square meter). This sensor provides full spectrum, visible and infrared (IR) measures, and also the light intensity expressed in Lux (Fig. 3b).

The PIR Motion Sensor HC-SR501 is a adjustable sensitivity module that allows for a motion detection range from 3 meters to 7 meters (Fig. 3c).

The MH-Z19 (Zhengzhou Winsen Electronics Technology CO., LTD) is non-dispersive infrared (NDIR) principle to detect the existence of CO_2 in the air. This sensor is connected to the UART at 9600bps and measures carbon dioxide in a range of 0-5000ppm (Fig. 3d).

The contact switch allows to detect the opening or closing of a window (See Fig. 3e).

The DS3231 is a low-cost, and extremely accurate I^2C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO). The RTC plays a crucial role in the automation of control processes of environmental condition. This RTC is also equipped of a 32Kbits EEPROM allowing to store next step in the plant growing process which guarantees the operation of the installation in the event of a network failure. A DC 3V lithium battery ensures the power to the real-time clock for 10 years (Fig. 3f).

d) Weather Station Sensors: .

The BME680 is a sensor which allows to measure at same time the air temperature with an absolute accuracy of 1°C, relative humidity with an absolute accuracy of 3%, and air quality with an accuracy of 15% (See Fig. 4a).

The DPS310 (Infineon Technologies) is a I^2C high precision Barometric Air Pressure Sensors which can acquire the pressure with ±1 hPa absolute accuracy (Fig. 4b).

The GL5516 is a photoresistor with a spectrum peak value at 540n m operating between -30°C to 70°C (See Fig. 4c).

The anemometer (Wind Speed Sensor) for weather station

WH1080 (See Fig. 4d).

The TP40556 is a rechargeable lithium batteries module with protection circuits using the constant-current/constant-voltage(CC/CV) charging method. It can be powered by USB or power supply between 4.5 and 6.0V (See Fig. 4e).

The regulator DC12V-DC5V transform an input power supply of DC 8 to 40V in DC 5V maximum 25V. It is placed between the TP4056 charging module and the DC 12V 5W solar panel (See Fig. 4f).

The DS3231 was described in previously (See Fig. 3f).

e) Actuators: controls external equipment.

To turn on/off actuators, we use also ESP32-Wroom-32 with relay card to activate and deactivate air extractor and electric blinds (See Fig. 5a).

While Z-Wave devices are directly connected and controlled by the Raspberry Pi 4 by means of a USB Z-Wave controller (See Fig. 5b).

C. Implementation

The implementation of our proposition is composed of four parts. A central point placed in each campus collect all data transmitted by all Home Assistant/ESPpHome placed in each meeting room, auditorium, office, laboratory, and office.

a) A2IoT Architecture: centralizes all data from all nodes.

Our A2IoT is completely described in [16]. This architecture uses containerization and kubernetes an container orchestrator. It is composed of a Odroid N2 master which manages a mixed cluster of 3 Nivida Jetson to hosts adapted algorithms of Artificial Intelligence and 3 Odroid N2 to hosts micro-services.

The A2IoT architecture received data from all Home Assistant / ESPHome managing a local network of sensors and actuators. Each Raspberry PI is by the Ethernet Network of the University Network to A2IoT architecture and transmit them sensing data by MQTT Protocol and images taken by AI-Thinker to detect person presence (See Fig. 6). A2IoT architecture centralize all data produce and allows to remotely manages Raspberry PI and operate maintenance on ESPHome using OTA firmware deployment.

b) Home Assistant / ESPHome: acquires environmental data and operates actuators.

The sensors deployed in classrooms enable to collect ambient parameters (air temperature [°C], air humidity [%], CO_2 rate [ppm], light intensity [lux], passive infrared (*PIR*) presence sensor [-], sound level [dB]) connected to a ESP32-Wroom-32. This cheap micro controller transmit data on MQTT server by mean of dedicated Wi-Fi SSID in WAP2-PSK Enterprise deployed in all buildings of the Smart Campus. The same micro controller is used to actuate air extractors, control lights and connected radiator valves on basis of sensed data and auditorium occupancy planning recovered from hyper planning [1].

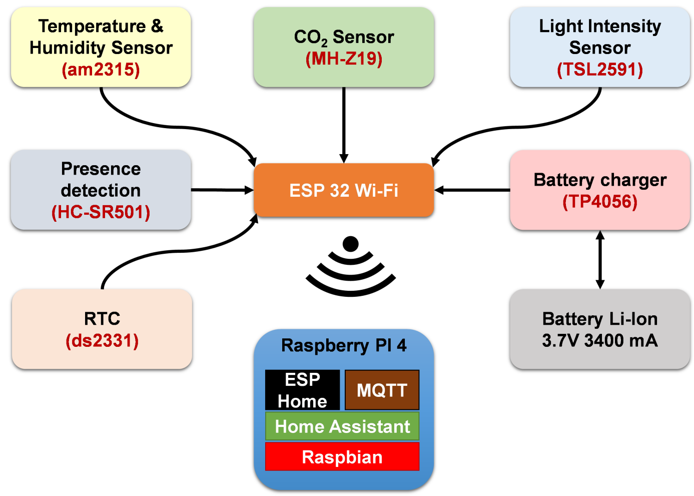


Fig. 7: Block Scheme of sensing device

The Table I shows sensors typically used for the sensing of ambient parameters in an auditorium. Air temperature, Air humidity are acquired by means of a am2315 (Aosong), the light intensity is measured with a TSL2591 (ams AG) in a range comprise 0.000118 to 88,000 lux, the CO_2 rate between 0 to 5000 ppm is determined with a MH-Z19 Winsen, HC-SR501 detect presence of persons in a range of 3 to 7 m.

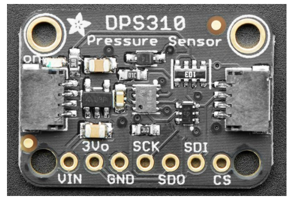
c) Outside of Buildings: data transmission is not ensured with Wi-Fi protocol.

Indeed, sensing is not limited to buildings and must be use adapted protocol to transmit data outside buildings. LPWAN technologies such as NB-IoT[2], SigFox[3], Ingenu[4], Weightless[5], LoRaWAN[6], Telensa[7], and NB-Fi[8] provide a wide area of coverage with a reduced energy consumption and have good capabilities in term of cost, power consumption, and capacity [25].

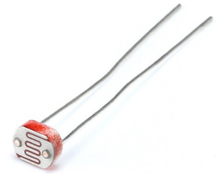
A mesh of antenna cover the campus in which each antenna covers a range up to 2 km. We use LoRaWan protocol to

[1] http://www.index-education.com/fr/presentation-hyperplanning.php
[2] http://www.3gpp.org/news-events/3gpp-news/1785-nb_io_complete
[3] https://www.sigfox.com/
[4] http://www.ingenu.com
[5] http://www.weightless.org/
[6] https://lora-alliance.org/about-lorawan
[7] http://www.telensa.com
[8] http://waviot.com/technology/whats-is-nb-fi

(a) BME680 Sensor



(b) High precision Barometric Sensor



(c) Photo resistor GL5516

(d) Anemometer

(e) Ion Battery Charger Module

(f) Converter DC12V-DC5V

Fig. 4: Weather Station Sensors, converter & controller

TABLE I: Power consumption according to manufacturer's data and interface of connection

Component	**Interface**	**Operation mode**	**Supply Current (Max)**	**Voltage**
Real-time Clock (DS3231)	I^2C	Active / Stand-by / Conversion	300 μA / 170 μA / 650 μA	5V
High Dynamic Range Digital Light Sensor (TSL2591)	I^2C	Sleep / Active	4 μA / 325 μA	5V
Temperature & Relative humidity (AM2315)	Digital	Stand-by / Measuring / Converting	50 μA / 1.5 mA / 10 mA	3 to 5V
Distance sensor (HC-SR501)	Digital	Sleep / Normal	2.5 mA / 20 mA	4.5 to 5.5V
CO2 Sensor (MH-Z19)	UART	Average	$<$ 18 mA	3.6 to 5.5V

(a) 8-Relays Card

(b) Z-Wave Valve

Fig. 5: Actuators

send ambient parameters to the implantation MQTT server and activate actuators in order to act to environment.

d) Weather Station: measures external environmental conditions.

The weather node is built around an ESP32-Lora V2 equipped of a 0.96" OLED Display is powered by a 3.7V Li-Ion battery. A 5W, 12 V solar panel mono crystalline is connected to a DC12V-DC5V converter which powers the TP4056, a Li-Ion battery charger module. A BME680 Digital Sensor Temperature, Humidity, Atmospheric Pressure, Air Quality Sensor Module is connected to the EPS32-Lora microcontroller by means of I^2C bus. A Froggit wind speed sensor acquires the wind speed. A Photoresistor GL5516 allows to measure the light Intensity (See Fig. 8).

IV. Experimentation

We have chosen to equipped a computer laboratory with 6 windows, 1 air extractor implanted in 1 window and 3 radiators. Moreover, it contains also 25 personal computer and a video projector which can be tun on / off manually.

We have installed the last release (109.6) of Home Assistant and the last release of ESPHome (1.14.3) on the local micro computer, a Raspberry PI 4 4gb. Specific libraries for ESPHome have been developed to allows the support of TSL2591 and the DS3231 Real-Time Clock.

We have changed radiator valve with Z-Wave connected radiator valves which can be managed directly by the Raspberry Pi via a USB Z-Wave dongle.

Two environmental Sensor Node built around a ESP32-Wroom-32 on which are connected to I^2C bus a Real Time Clock (DS3231), a light intensity (TSL2591), a Pressure Sensor (DPS310), a CO_2 Sensor (MH-Z19) in UART, and a temperature & humidity sensor and Motion Detection Sensor on digital pin. These nodes are placed on the ceiling and sent

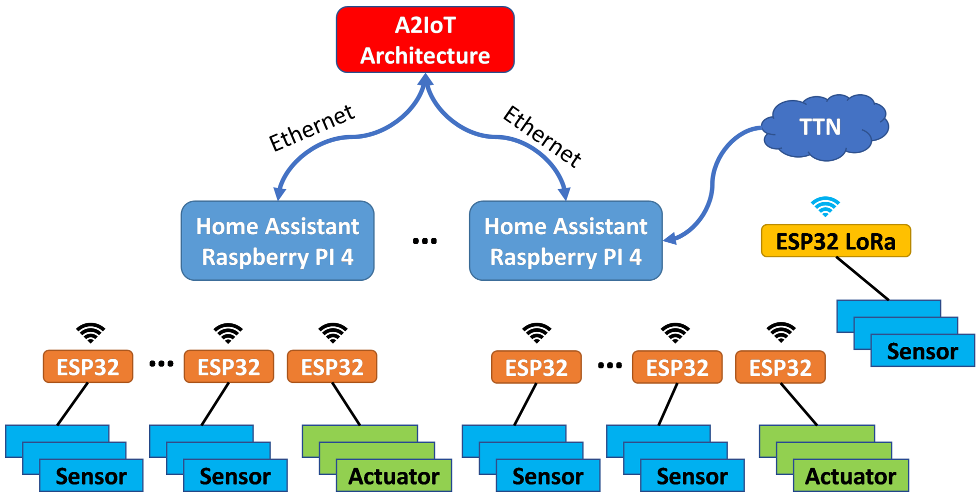


Fig. 6: Global Schema of the proposed architecture

TABLE II: Power consumption according to manufacturer's data and interface of connection for weather station

Component	**Interface**	**Operation mode**	**Supply Current (Max)**	**Voltage**
Real-time Clock (DS3231)	I^2C	Active / Stand-by / Conversion	300 μA / 170 μA / 650 μA	5V
Photoresistor (GL5516)	Analogic	Unkown	Unknown	3 to 5 V
Wind Speed (Foggit)	Analogic	Unkown	Unknown	3 to 5V
Tmp. Hum. Air Quality Sensor (BME680)	Digital	Sleep / Standby / Normal	1 μA / 0.8 μA / 0.09 mA to 12 mA	4.5 to 5.5V
Pressure sensor (DPS310)	I^2C	Average / Standby	1.7 μA / 0.5 μA	1.7 to 3.7V

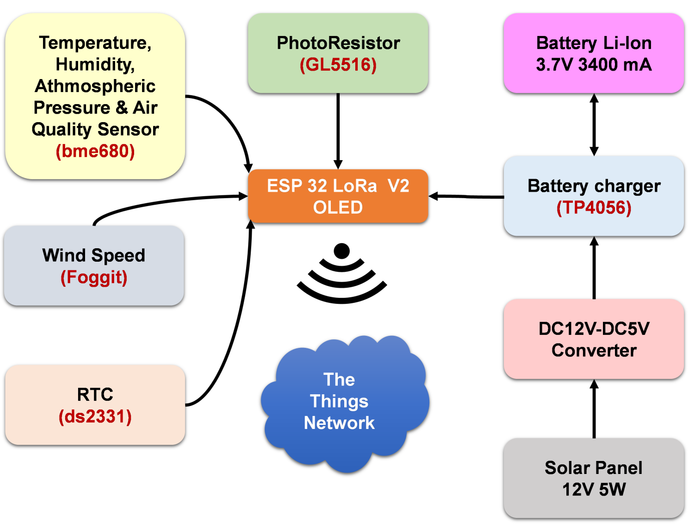


Fig. 8: Block Scheme of weather station

their data at intervals of 60 seconds.

An ESP32 operates the relay card which tun on/off the air extractor, turn on/off the lights, turn off the video projector and go up/down electrical blinds.

An ESP32 is relied with contact switches in order to verify that windows are closed or opened.

Two AI-Thinker nodes are installed front of door to detect people presence. Images are sent to Home Assistant which relay them when it is necessary to the A2IoT Architecture where people detection is operated with Tiny Yolo V3.

A weather station built around an ESP32-Lora V2 equipped of temperature, humidity, air quality sensors (BME680), a Real Time CLock (DS3231), a High precision Atmospheric pressure sensor(DPS310) connected on I^2C bus.

All ESPHome nodes status and wireless signal strength are monitored by Home Assistant in order to detect failed or temporary disconnected nodes.

An automation has been implemented in Home Assistant with following rules (See Table III).

Rule 1 activates the air extractor when the air temperature measures with the am2315 is more than 23°C or air humidity exceeds 70% or CO2 rate exceeds the limit of 1500ppm. Rule 2 stop the air extractor when all the three following conditions are met: temperature ¡ 20°C, air humidity <40%, and CO_2 rate $<$ 1000ppm. Rules 3 and 4 increase or decrease the temperature by operating the Z-Wave radiator valves via the Raspberry Pi 4. Rule 5 tun on light when a people is detected by the HC-SR501 motion sensor or at least one people is detected at A2IoT Architecture level by Tiny Yolo V3 on images sent by AI-Thinker. Rule 6 turns off lights when no body is detected on AI-Thinker images and all windows are closed. This rule prevents the forgetting to close windows. Rule 7 dims lights when the video projector is turned on. Rule

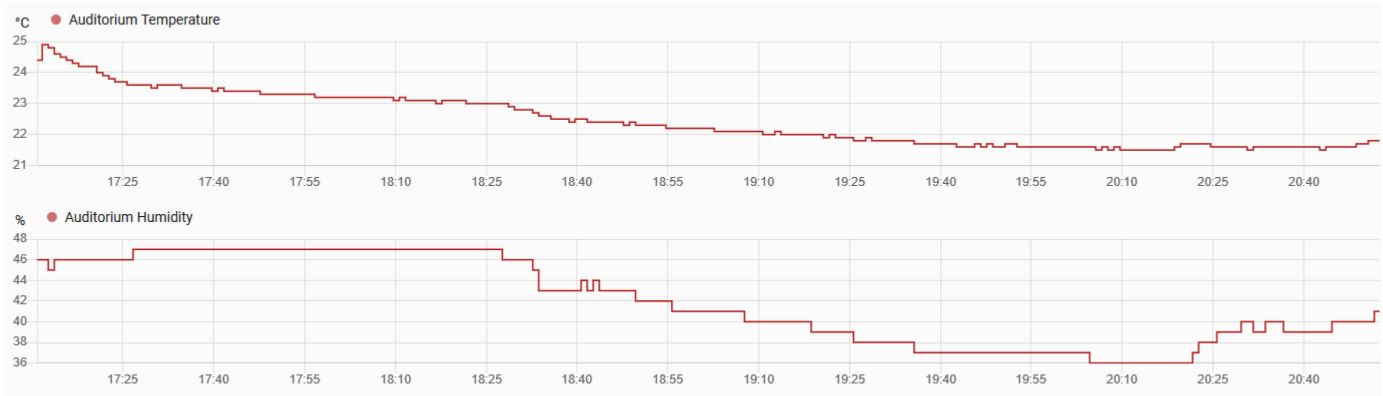


Fig. 9: Example of few parameters monitored an auditorium

TABLE III: Automation expressed in High Level Semantic Rules.

#	Fact	Triggering Rule
1	ExtractorStart	Observation hasTemperature $>$ 23°C $\vee$ Observation hasHumidity $>$ 70% $\vee$ Observation hasCO2 $>$1500ppm $\rightarrow$ ns:ExtractorStart
2	ExtractorStop	Observation hasTemperature $<$ 20°C $\wedge$ Observation hasHumidity $<$ 40% $\wedge$ Observation hasCO2 $<$1000ppm $\rightarrow$ ns:ExtractorStop
3	ValveUP	Observation hasTemperature $<$ 20°C $\rightarrow$ ns:ValveUp
4	ValveDown	Observation hasTemperature $>$ 22°C $\rightarrow$ ns:ValveDown
5	LightOn	Observation hasPeopleDetected true $\vee$ hasPIRDetection true $\rightarrow$ ns:LightOn
6	LightOff	Observation hasPeopleDetected false $\wedge$ hasWindowClosed true $\rightarrow$ ns:LightOff
7	LightDimmed	Observation hasProjectorOn true $\rightarrow$ ns:LightDimmed
8	BlindsUp	Observation hasLightIntensity $\leq$ 1000lux $\rightarrow$ ns:BlindsUp
9	BlindsDown	Observation hasLightIntensity $>$ 1000lux $\wedge$ hasWindowClosed true $\rightarrow$ ns:BlindDown
10	ProjectorOff	Observation hasPeopleDetected false $\wedge$ hasProjectorOn true $\rightarrow$ ns:ProjectorOff

8 goes up electrical blinds when the light intensity measures by the TSL2591 sensor is under or equal to 1000 lux. Rule 9 goes down the electrical blinds when the light intensity measured by TSL2591 is more than 1000 lux and all windows are closed (verified with contact switches placed on windows). Finally, rule 10 closes the projector when it is tuned on and there is nobody in the laboratory.

V. Results and Discussion

The major advantage to use Home Assistant in each room is to have autonomous solution that control actuators such air extractors, radiator valves, dimmers. But this solution is most costly than a central approach because it need a Raspberry Pi to host Home Assistant / ESPHome in each room. On the other hand, the fact of distributing the resources locally makes the solution more resilient to the breakdowns of network or electricity cut which can occur in certain parts of the building. Moreover, Home Assistant and ESPHome are maintained by a important community which add continuously new integration and extend possibilities in term of sensors, actuator supported, and system interfacing.

VI. Conclusion and perspectives

In this paper, we propose a distributed smart campus solution based on diverted Smart Home solution: Home Assistant coupled with ESP Home used to manage a network of ESP32 microcontroller. Each room of the building is equipped of an autonomous Home Assistant which sense environmental parameters and automate actuators in function determinated rules. The solution is easily extensible and the replacement of failed device, the adding of new device are easy. Moreover, devices in the ESPHome network are updated in Over-the-air programming (OTA).

Acknowledgment

The authors would especially like to thank Mr Adriano Guttadauria for his technical support and for setting up all the electronic systems and computing systems necessary for carrying out this research.

References

[1] C. Prandi, L. Monti, C. Ceccarini, and P. Salomoni, "Smart campus: Fostering the community awareness through an intelligent environment," *Mobile Networks and Applications*, pp. 1–8, 2019.

[2] 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, 2020.

[3] Y. Liu, G. Shou, Y. Hu, Z. Guo, H. Li, and H. S. Seah, "Towards a smart campus: Innovative applications with wicloud platform based on mobile edge computing," in *2017 12th International Conference on Computer Science and Education (ICCSE)*. IEEE, 2017, pp. 133–138.

[4] A. Abuarqoub, H. Abusaimeh, M. Hammoudeh, D. Uliyan, M. A. Abu-Hashem, S. Murad, M. Al-Jarrah, and F. Al-Fayez, "A survey on internet of things enabled smart campus applications," in *Proceedings of the International Conference on Future Networks and Distributed Systems*, 2017, pp. 1–7.

[5] M.-P. Gleizes, J. Boes, B. Lartigue, and F. Thiébolt, "neocampus: A demonstrator of connected, innovative, intelligent and sustainable campus," in *Intelligent Interactive Multimedia Systems and Services 2017*, G. De Pietro, L. Gallo, R. J. Howlett, and L. C. Jain, Eds. Cham: Springer International Publishing, 2018, pp. 482–491.

[6] O. Debauche, S. A. Mahmoudi, S. Mahmoudi, and P. Manneback, "Cloud platform using big data and hpc technologies for distributed and parallels treatments," *Procedia Computer Science*, vol. 141, pp. 112–118, 2018.

[7] O. Debauche, S. Mahmoudi, A. L. H. Andriamandroso, P. Manneback, J. Bindelle, and F. Lebeau, "Web-based cattle behavior service for researchers based on the smartphone inertial central," *Procedia Computer Science*, vol. 110, pp. 110 – 116, 2017, 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017) / 12th International Conference on Future Networks and Communications (FNC 2017) / Affiliated Workshops. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050917313066

[8] O. Debauche, S. Mahmoudi, A. Andriamandroso, P. Manneback, J. Bindelle, and f. Lebeau, "Cloud services integration for farm animals' behavior studies based on smartphones as activity sensors," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 12, pp. 4651–4662, 2019.

[9] O. Debauche, S. Mahmoudi, P. Manneback, N. Tadrist, J. Bindelle, and F. Lebeau, "Improvement of battery life of iphones inertial measurement unit by using edge computing application to cattle behavior," in *2017 Symposium International sur les Sciences Informatiques et Applications (ISCSA2017)*, 2017.

[10] O. Debauche, S. Mahmoudi, S. A. Mahmoudi, P. Manneback, J. Bindelle, and F. Lebeau, "Edge computing for cattle behavior analysis," in *2020 Second international conference on Embedded Distributed Systems (EDiS)*, 2020, pp. 1–5.

[11] O. Debauche, S. Mahmoudi, P. Manneback, M. Massinon, N. Tadrist, F. Lebeau, and S. A. Mahmoudi, "Cloud architecture for digital phenotyping and automation," in *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, Oct 2017, pp. 1–9.

[12] O. Debauche, S. A. Mahmoudi, N. De Cock, S. Mahmoudi, P. Manneback, and F. Lebeau, "Cloud architecture for plant phenotyping research," *Concurrency and Computation: Practice and Experience*, vol. n/a, no. n/a, p. e5661, 2020, e5661 cpe.5661. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5661

[13] O. Debauche, M. E. Moulat, S. Mahmoudi, S. Boukraa, P. Manneback, and F. Lebeau, "Web monitoring of bee health for researchers and beekeepers based on the internet of things," *Procedia Computer Science*, vol. 130, pp. 991 – 998, 2018, the 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050918304654

[14] O. Debauche, M. El Moulat, S. Mahmoudi, P. Manneback, and F. Lebeau, "Irrigation pivot-center connected at low cost for the reduction of crop water requirements," in *2018 International Conference on Advanced Communication Technologies and Networking (CommNet)*, April 2018, pp. 1–9.

[15] M. E. Moulat, O. Debauche, S. Mahmoudi, L. A. Brahim, P. Manneback, and F. Lebeau, "Monitoring system using internet of things for potential landslides," *Procedia Computer Science*, vol. 134, pp. 26 – 34, 2018, the 15th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2018) / The 13th International Conference on Future Networks and Communications (FNC-2018) / Affiliated Workshops. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050918311037

[16] O. Debauche, S. Mahmoudi, R. Doukha, S. A. Mahmoudi, and P. Manneback, "A new edge architecture for ai-iot services deployment," *Procedia Computer Science*, 2020, the 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC) / The 15th International Conference on Future Networks and Communications (FNC 2020) / Affiliated Workshops.

[17] O. Debauche, R. A. Abdelouahid, S. Mahmoudi, S. A. Mahmoudi, P. Manneback, and F. Lebeau, "Edge computing and artificial intelligence for real-time poultry monitoring," *Procedia Computer Science*, 2020, the 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC) / The 15th International Conference on Future Networks and Communications (FNC 2020) / Affiliated Workshops.

[18] O. Debauche, S. Mahmoudi, P. Manneback, and A. Assila, "Fog iot for health: A new architecture for patients and elderly monitoring." *Procedia Computer Science*, vol. 160, pp. 289 – 297, 2019, the 10th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2019) / The 9th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2019) / Affiliated Workshops. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050919317880

[19] R. Ait Abdelouahid, O. Debauche, S. Mahmoudi, and P. Manneback, "Smart birds," in *2020 International Conference on Advanced Communication Technologies and Networking (CommNet)*, 2020.

[20] O. Debauche, S. Mahmoudi, M. A. Belarbi, M. El Adoui, and S. A. Mahmoudi, "Internet of things: Learning and practices. application to smart home," in *2018 International Conference on Advanced Communication Technologies and Networking (CommNet)*, April 2018, pp. 1–6.

[21] O. Debauche, S. Mahmoudi, and S. A. Mahmoudi, "Internet of things: learning and practices. application to smart city," in *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*, Nov 2018, pp. 1–7.

[22] O. Debauche, S. Mahmoudi, and Y. Moussaoui, "Internet of things learning: a practical case for smart building automation," in *2020 5th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*, 2020, pp. 1–7.

[23] J. Alonso, C. Bayona, O. Rojas, M. Terán, J. Aranda, H. Carrillo, and C. Parra, "Iot solution for data sensing in a smart campus using smartphone sensors," in *2018 IEEE Colombian Conference on Communications and Computing (COLCOM)*. IEEE, 2018, pp. 1–6.

[24] S. Ward and M. Gittens, "Building useful smart campus applications using a retired cell phone repurposing model," in *2018 Third International Conference on Electrical and Biomedical Engineering, Clean Energy and Green Computing (EBECEGC)*. IEEE, 2018, pp. 43–48.

[25] 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, 2019.

[26] R. Ait Abdelouahid, M. Oquaidi, and A. Marzak, "Towards to a new iot interoperability architecture," in *2018 IEEE International ConfereFnce on Technology Management, Operations and Decisions (ICTMOD)*, Nov 2018, pp. 148–154.