INTERREG MED Programme

2014-2020

ESMARTCITY

Enabling Smarter City in the MED Area through Networking

(3MED17_1.1_M2_022)

Priority Axis 1. Promoting Mediterranean innovation capacities to develop smart and sustainable growth

Specific Objective 1.1 To increase transnational activity of innovative clusters and networks of key sectors of the MED area

WP3 – Testing

Activity 3.3 – Pilot Testing

Deliverable 3.3.1 – Pilot deployment – Partner PP1

Contractual Delivery Date: **30.06.2019**

Authors: **Christos Alexakos, George Kalogeras, Alkis Savvopoulos, Athanasios Kalogeras (ISI)**

Project Coordinator: **Iris Flacco (LP – ABREG)**

Dissemination Level		
PU	Public	X
PP	Restricted to Programme Partners and MED Programme	
RE	Restricted to a Group defined by the Partnership and MED Programme	
CO	Confidential, only for members of the partnership and MED Programme	

Content

Project co-financed by the European Regional Development Fund

1 Introduction

WP3 aims at testing the concept of Smart Cities through deployment of pilot implementations in partner project areas and testing new concepts. One of the testing challenges to be adhered to is an increase in the level of innovation in MED cities by enriching city infrastructure via smart devices, embedded systems and sensors/actuators, seamlessly integrated and interoperable, and deploying on top of them novel applications / services increasing the level of well-being of the citizens.

Industrial Systems Institute undertakes this challenge by deploying a pilot implementation at its own premises. The pilot consists of installing a number of smart sensors and metering devices to collect data on building operation and user comfort, as well as to control conditions and energy usage remotely and automatically.

This deliverable describes the pilot implementation in detail and lays out the system design and architecture, the installation of all devices as well as the data collection and analysis procedure. Finally, pilot deployment results and possible future works are discussed in the final chapters.

2 Summary of the Pilot Deployment

The pilot testing undertaken by the Industrial Systems Institute (PP1 – ISI) has been deployed at its own premises at the Patras Science Park (PSP) building in Platani, Patras, Greece. It addresses two concepts: the concept of **energy efficiency** and the concept of the building as a **living lab**. Energy efficiency is achieved by monitoring the power consumption of the building through installed sensors and smart metering devices. Afterwards scenarios are executed based on the analysis of the collected data. These scenarios highlight the areas where interventions should be made so as to lower the building's power consumption.

The living lab concept involves scenarios in real life use cases in areas where people actually work that will allow experimentation with innovative ideas and technologies.

Project co-financed by the European Regional Development Fund

Figure 1: View of Patras Science Park Building, Platani Patras, Greece

The pilot testing utilizes different types of sensing / actuating equipment in order to perform some testing / experimentation with different scenarios: sensors that measure temperature, relative humidity, lighting levels and room occupancy, as well as smart metering devices that measure power consumption and quality are deployed throughout the premises of ISI offices.

Furthermore, actuating devices along with analytics and decision making software are used for controlling power consumption. Thus, the pilot infrastructure comprises:

- Smart devices, IoT, smart embedded systems (Energy metering equipment, Gateways, Different sensors related to the external climatic conditions and being capable of measuring temperature, humidity, irradiance, light, Different sensors for determining space occupancy, Remotely controlled switches)
- Wireless connectivity
- Edge computing, Cloud computing
- Big data analytics

Project co-financed by the European Regional Development Fund

The pilot has been deployed using affordable IoT technologies and diverse smart devices together with open source solutions to ensure vendor neutrality and transferability of the results.

2.1 Motivation

The rapid growth of global energy consumption in recent decades has raised concerns over generated problems, ranging from difficulties in energy supply to environmental problems. It is therefore entirely logical that energy efficiency in buildings has been brought to the spotlight of recent developments. In fact, growth in population, increasing demand for building services and comfort levels, together with the rise in time spent inside buildings, assure the upward trend in energy demand will continue in the future [1] (Pérez-Lombard et al., 2008).

Buildings are the basic blocks of a city and account for a major part of its energy consumption. It makes sense that to name a city “smart” it needs to be made up of smart, energy efficient buildings. Smart buildings are able not only to monitor, but also to regulate their energy consumption and CO2 emissions using an array of sensors and actuators that gather data on internal conditions and building user behavior.

Apart from the increased energy efficiency, a smart building also aids its occupants by providing better and more customized working and living conditions. Especially in office buildings, maintaining ideal working conditions, which include air quality, lighting, internal temperature, humidity and security, is essential for raising productivity of the building occupants.

Another benefit of transforming buildings into smart ones is the capability for predictive maintenance. A smart building can monitor the entire building performance and trigger alerts for maintenance before anything malfunctions. Therefore, it is easier and less costly to maintain the building in operational conditions.

Finally, smart buildings offer their administrators large volumes of real data which, when analyzed correctly, provide useful insights that should be taken into account when planning with an eye on making the use of resources more efficient.

2.2 Pilot Definition and Goals

The pilot deployment utilizes a number of IoT devices in the form of energy meters, sensors and actuators that gather information on the building operation and act upon results from the analysis of the gathered data. The analysis is based on big data analytics performed using open source tools and IoT data acquisition platforms.

The results from this analysis are interpreted to provide insights and to create rules for when actuators or building administrators need to take action. This action refers to making the

[1] Pérez-Lombard, L., Ortiz, J., & Pout, C. (2008). A review on buildings energy consumption information. Energy and buildings, 40(3), 394-398

necessary adjustments in an operating system or in preventively maintaining equipment before it reaches a point where it would need replacement.

For any data analysis to be useful, the quality of data must be of a very high level. That is why the pilot takes advantage of high performance, yet affordable IOT technologies that monitor building performance in real time. By utilizing edge and cloud computing technologies and wireless communication, the pilot can also analyze data in real time, making the system response to any event almost instant. This means that the system can detect anomalies in the building operation and act to remove them, either for increasing the energy performance of the building, or for improving the level of comfort for building occupants.

The system has been designed to be vendor neutral. That is why it uses a set of diverse devices for heterogeneity. Moreover, open source solutions are always preferable, and they have been used in this specific pilot as well, together with standard protocols used in similar applications. Finally, transferability is a major concern for every pilot deployment. That is why the implemented solution has been designed to be completely scalable and replicable, through the use of open source tools.

2.3 User and Functional Requirements

The deployed pilot aims to support the Smart City paradigm through two different concepts both referring to a smart building. The first concept is the building energy efficiency, while the second one is referred to as the “living lab” concept. To support these two concepts, the pilot deployment monitors and analyzes in real time a wide variety of parameters such as temperature, humidity, human presence, light, vibrations and sounds, door/window opening. The gathering of such data allows the system to create energy profiles of spaces as well as the entire premises of the Institute and associate these profiles with external conditions and internal factors like occupancy.

The creation of these profiles allows the system to perform control and actuation over several building factors and systems related to the optimization of the energy consumption, the assurance of user comfort and the testing of different scenarios that will help provide insights on the characteristics of the building and the behavior of the occupants that affects energy consumption.

Through this procedure the pilot deployment covers the following user requirements:

- Exploit cutting-edge intelligence to test different control and actuation scenarios, aiming at building energy consumption optimization while maintaining the user’s required comfort level.
- Derive occupancy behavioral characteristics and associate them to the building energy consumption

- Deploy a highly heterogeneous net of sensors and actuators and manage their interconnection and real-time communication through different network protocols.
- Create a Graphical User Interface for devices management, data visualization and actuators control. Data visualization includes graphic charts and colorful alert messages that can be managed by the user.
- Provide a solid data management plan to perform statistical and machine learning analysis in order to detect anomalies or patterns that will provide knowledge for optimizing energy consumption.

Based on the pilot deployment user requirements presented above, the pilot functionality is presented in the following list:

- Real Time Monitoring. The monitoring devices provides real-time data, meaning that for power consumption at least 0.1Hz sampling rate will be needed (1 sample per 10 seconds). In the case that a measured quantity is characterized by smooth changes, such as temperature, the sampling rate can be decreased to 0.016Hz (1 sample per 1 minute)
- Real Time Transferring. The monitored values are sent from the device to the IoT data acquisition Platform in a period of no more than 8 sec, 2sec less than the sampling rate. Even if a gateway is acting in the middle the latency cannot be more than 2 seconds, in order not to have overlapping with the device sampling rate.
- Real Time Analysis. The data analysis and decision making that takes place in the platform tier cannot be over 10 seconds. Also, any response action such as message sending or enable an actuator to do something must not be over 10 seconds.
- Real Time Control. The deployed network system can control and actuation on the energy management systems.
- Device diversity. The system supports a variety of sensing and actuating devices, like smart plugs/sockets, smart switches, power meters, light sensors, temperature sensors, etc. Also, the system supports vendor neutrality.
- Machine-to-machine (M2M) communication. Data protocols such as CoAP, MQTT and HTTP/HTTPS are supported by the infrastructure.
- Network Protocols. The devices network is based on TCP/IP network protocol, or other protocols specialised for M2M communication such Bluetooth, zigBee and Ocean3.
- Wireless M2M communication. Device communication can be done wirelessly since the longest distance in the offices is about 10 meters. Wireless technologies that can support this limitation are WiFi, Bluetooth, ZigBee.
- Data Collection. All data is collected in the Platform Tier of the application, where it is processed. Results are then pushed to the cloud tier.

Project co-financed by the European Regional Development Fund

- Big Data Analytics. Perform of statistical and machine learning analysis in order to detect anomalies or patterns that will provide knowledge for optimizing energy consumption.
- Infrastructure & Data Visualization. IoT platform provides a Graphical User Interface for devices management, data visualization and actuators control, including easily noticeable alerts when action is required.

3 Pilot Implementation

3.1 Design and Planning

3.1.1 Installation Plan

The installation plan shows where the physical devices are installed. For the Living Lab the following offices of PSP[2] where used:

- Office A7 (1st Floor) - Administration
- Office A6 (1st Floor) – Director Office
- Office A5 (1st Floor) - Researchers & Collaborating Personnel
- Office B13 (2nd Floor) - Collaborating Personnel
- Office B8 (2nd Floor) - Researchers & Technical Personnel

Some offices are separated to smaller rooms. The figures depict the installation places of each device in the areas of each office area.

[2] https://www.psp.org.gr/

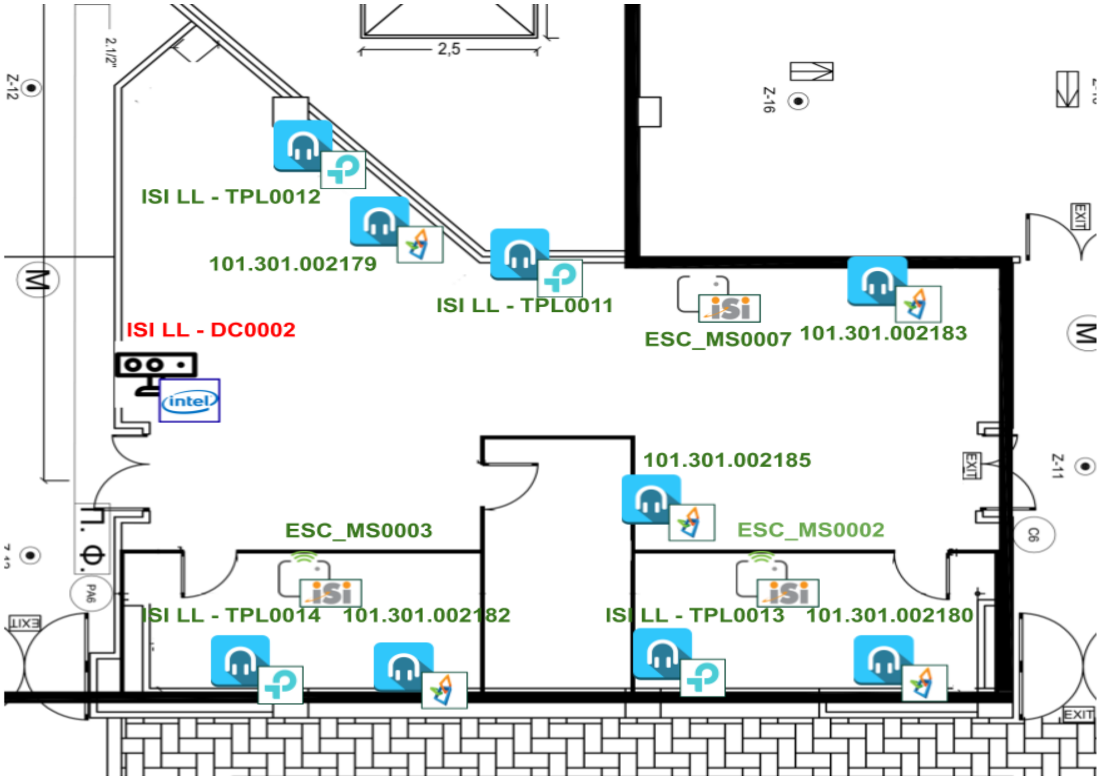

Figure 2: Installation plan for office A7

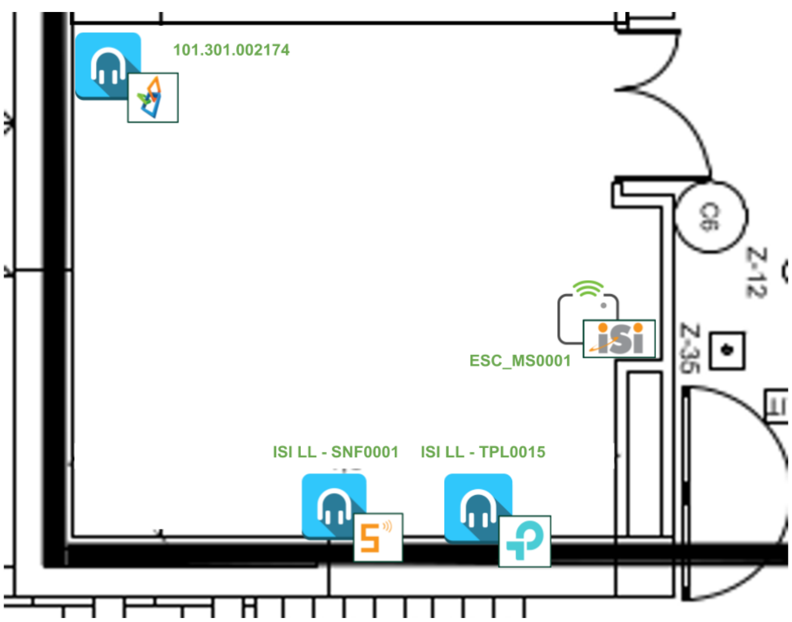

Figure 3: Installation plan for office A6

Project co-financed by the European Regional Development Fund

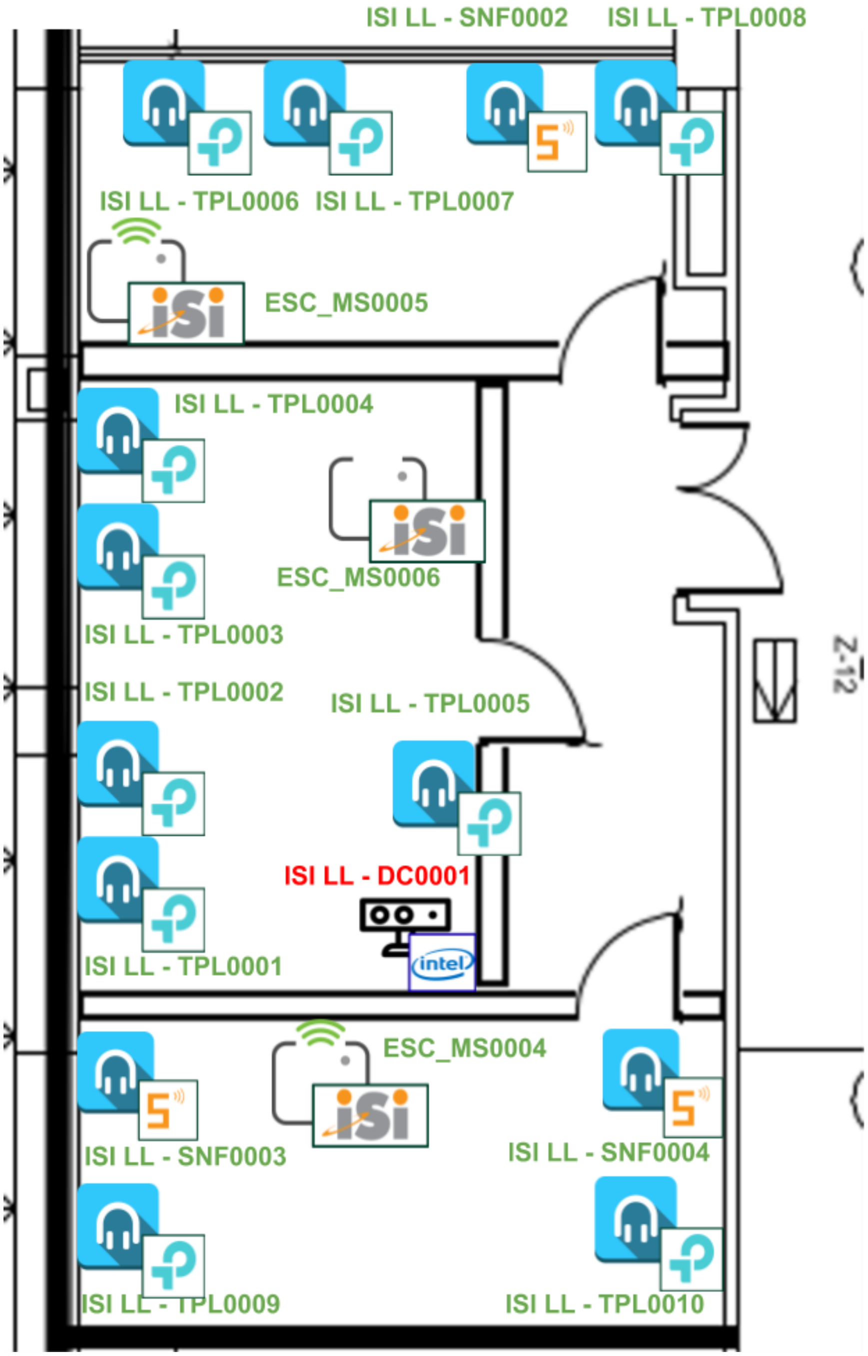

Figure 4: Installation plan for office A5

Project co-financed by the European Regional Development Fund

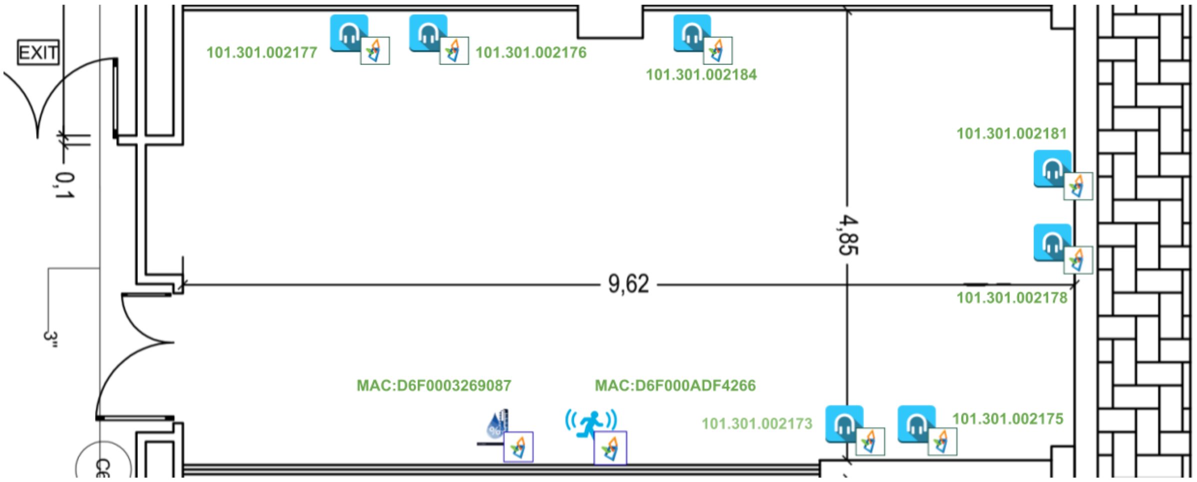

Figure 5: Installation plan for office B13

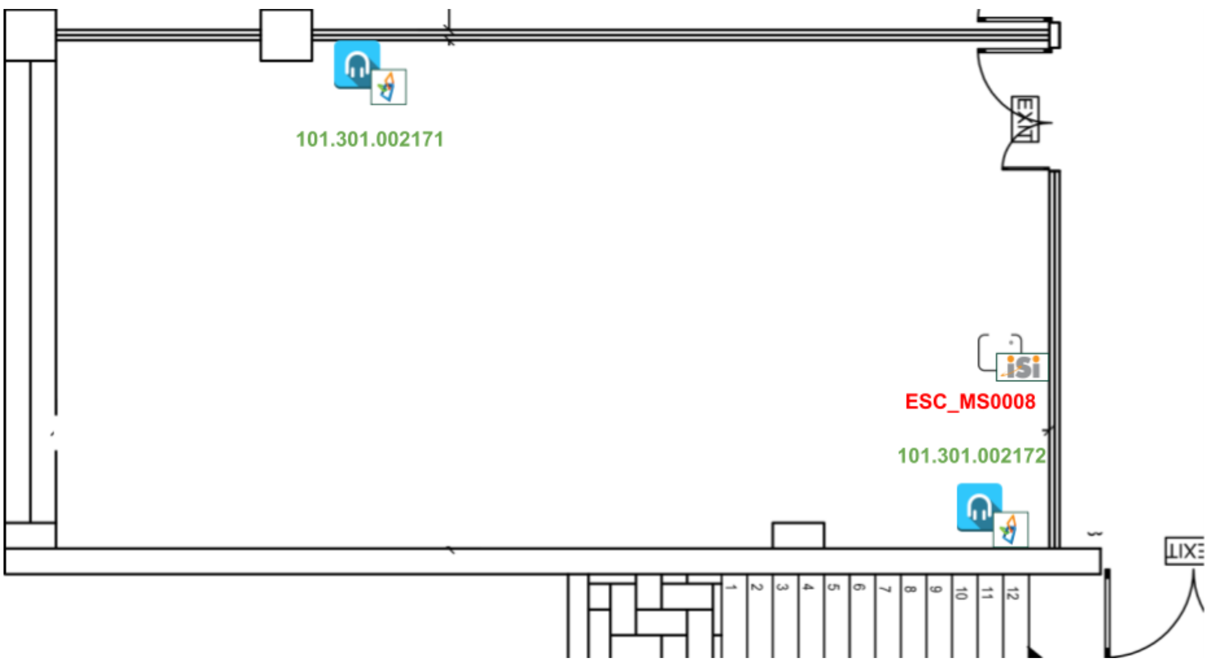

Figure 6: Installation plan for office B8

Project co-financed by the European
Regional Development Fund

3.1.2 Bill of Materials

3.1.2.1 Smart Plugs

The measurement of power consumption and the control of the power supply is made mainly by the use of smart plugs. In the installation there are three different models of smart plugs from different vendors.

- MEAZON BizyPlug[3]
- TP-Link Wi-Fi Smart Plug[4] with Energy Monitoring (HS110)
- Itead Sono[5]ff S20 EU WiFi Wireless Smart Socket Sonoff S20 EU

MEAZON BizyPlug

Meazon BizyPlug is used for controlling the power feed (on/off/scheduling) and measuring energy & power consumed on electrical appliances. It connects wirelessly via ZigBee[6] an appliance to the Meazon Janus Gateway which forwards the measurements to the IoT platform. It's an ideal device to plug in appliances such as pc monitors, printers, lighting, coffee makers, water heaters, chargers, air conditions or any other electrical appliance that could be controlled by switching it on and off. In addition, Meazon Bizy plug is suitable for larger appliances such as refrigerators and window air conditioners.

Specifications:

- Voltage 85 - 265 VAC 45-65 Hz
- Maximum Load 16A
- Internal relay 16A on/off state
- Power consumption Less than 0.5 W
- Accuracy Higher than 99%
- Communications IEEE 802.15.4 / ZigBee
- ZigBee Standard Home Automation 1.2
- Frequency 2.4 GHz
- Coverage Dimensions Up to 50m indoor / mesh topology, 43 x 87 x 87 (WxHxD) in mm
- Operating Environment Indoor rated : -20°C to 50°C

[3] https://meazon.com/
[4] https://www.tp-link.com/gr/home-networking/smart-plug/hs110/
[5] https://www.itead.cc/smart-home.html
[6] https://zigbee.org/

Figure 7: Picture of Meazon BizyPlug

TP Link HS110

TP Link HS110 is a smart plug ideal for Energy Monitoring. It supports the following features:

- Remote Access – Control devices connected to the Smart Plug wherever you have Internet using the free Kasa app on your smartphone.
- Scheduling – Schedule the Smart Plug to automatically power electronics on and off as needed, like setting lights to come on at dusk or turn off at sunrise.
- Energy Monitoring – Analyze a device’s real-time and historical power consumption.
- Away Mode – Turns your devices on and off at different times to give the appearance that someone is home.
- Amazon Echo Voice Control – Amazon Echo (sold separately) lets you control devices connected to the Smart Plugs just using your voice.

The following table presents TPLink HS110 specifications.

Project co-financed by the European Regional Development Fund

NETWORK	
Protocol	IEEE 802.11b/g/n
Wireless Type	2.4GHz, 1T1R
System Requirements	Android 4.1 or higher, iOS 9.0 or higher
GENERAL	
Certification	RoHS, EAC, CE
Package Contents	Smart Plug HS110, Quick Start Guide
Environment	• Operating Temperature: 0 ºC ~ 40 ºC (32°F ~ 104°F) • Operating Humidity: 5%~90%RH, Non-condensing
Dimensions(H X W X D)	3.9 x 2.6 x 3 in. (100.3 x 66.3 x 77 mm)
Materia	PC
Buttons	Power button, Settings button
Weight	131.8g
Packaging Dimensions	3.5 x 3.5 x 5.7 in. (90 x 88 x 144 mm)
WORKING STATUS	
Input voltage	100 - 240VAC
Output voltage	100-240VAC
Maximum Load	16A
Maximum Power	3.68KW

Project co-financed by the European
Regional Development Fund

Figure 8: Picture of TP Link HS110

Itead Sonoff S20 EU

S20 Smart Socket (EU/US/UK/CN plug) is a wifi wireless smart socket that can connect any home appliances and electric devices via wifi, allowing you to remote control on iOS/Android APP eWeLink.

Its main features are:

- Remote ON/OFF–Turn electrical devices on/off from anywhere
- App Support –Free iOS and Android mobile App eWeLink
- LAN Control –Turn on/off the device even when WiFi has no Internet access
- Sync Status–Real-time device status provided to App
- Timing–Set scheduled/countdown/loop timers to turn on/off at specified time
- Share Control– Control your smart home together with your family

Project co-financed by the European
Regional Development Fund

- Scene–Turn on/off a gang of devices with one tap
- Smart Scene–Triggered on/off by temp, hum or other environmental conditions from sensor
- Compatibility –Works perfectly with Amazon Alexa, Google Assistant, IFTTT, Google Nest

Specifications:

- Working Conditions
- Support 90~ 250V AC input
- Max. input current:10A
- Max. power: 2000W
- Support fast configure SSID and password connection through APP
- Support auto-connect to server, register and update status info
- Support tracking device status and timely remote control through APP
- Support setting countdown, single and repeat timing schedules
- WIFI Characteristics
- 802.11 b/g/n
- Built-in Tensilica L106 ultra-low power consumption 32-bit micro-MCU, dominant frequency support 80 MHz and 160 MHz, support RTOS
- Built-in TCP/IP protocol stack
- Built-in TR switch, balun, LNA, power amplifier and matching network
- Built-in PLL, voltage regulator and power supply management components, 802.11b mode +20 dBm output power
- A-MPDU&A-MSDU aggregation and 0.4µs guard interval
- WiFi @ 2.4 GHz, supports WPA / WPA2 safe mode
- Support cloud OTA upgrade
- Support STA/AP/STA + AP mode
- Standby power consumption is less than 1.0 mW (DTIM3)

Project co-financed by the European Regional Development Fund

Figure 9: Picture of Itead Sonoff S20 EU

3.1.2.2 Smart Switches

For light control, the Itead Sonoff Touch EU and Itead Sonoff T1 2 Gang EU was selected.

Itead Sonoff Touch EU and Itead Sonoff T1 2 Gang EU

The Sonoff Touch smart wall switch is a touch control luxury crystal glass panel wall switch. The wireless wall switch can be added to iOS/Android App eWeLink via WiFi, allowing users to remotely turn on/off connected LED and lights from anywhere at any time. With this touch light switch, users can check real-time light status on their smartphones. The remote wall switch's LED backlight lets users easily find and turn on the lights at night. The WiFi wall switch can be used to control lights or home appliances. Upgrade your home with this modern, fashionable alternative to traditional light switches.

Features:

- Touch ON/OFF
- Remote ON/OFF–Turn lights on/off from anywhere
- LAN Control –Turn on/off the device even when WiFi has no Internet access
- App Support –Free iOS and Android mobile App eWeLink
- Sync Status–Real-time device status provided to App
- Timing–Set scheduled/countdown timers to turn on/off at specified time
- Share Control– Control your smart home together with your family

- Scene–Turn on/off a gang of devices with one tap
- Smart Scene–Triggered on/off by temp, hum or other environmental conditions from sensor
- Compatibility –Works perfectly with Amazon Alexa, Google Assistant, IFTTT, Google Nest

Figure 10: Picture of Itead Sonoff T1 2 Gang EU

3.1.2.3 Multisensors

For the monitoring of the indoor environmental conditions and the human presence in the rooms, three types of devices where used:

A custom multisensor that was developed by ISI that integrates temperature, humidity, light and motion detection sensors.

1. A motion detector by MEAZON.
2. A temperature and humidity detector by MEAZON.

Project co-financed by the European Regional Development Fund

ESC Multisensor

The ESC MultiSensor was developed by ISI for the project ESMARTCITY. It consists of an Arduino UNO microcontroller with integrated WiFi. Furthermore, three types of sensors are connected to the I/O pins of the Arduino:

- DHT21 AM2301 Digital Temperature Humidity Sensor module with SHT11 SHT15 for Arduino.
- Photo Light Sensitive Resistor 5mm GL5539
- HC-SR501 PIR Motion Sensor Module for Arduino

The data collected is transmitted via WiFi to the IoT platform.

The exact Bill of Material for each ESC MultiSensor is depicted in the following table.

Part Type	Part	Number
Microprocessor	Arduino Uno Wifi	1
Temperature/Humidity Sensor	DHT21 AM2301 Digital Temperature Humidity Sensor module with SHT11 SHT15 for Arduino	1
Motion Detector	HC-SR501 PIR Motion Sensor Module for Arduino	1
Light Detector	Photo Light Sensitive Resistor 5mm GL5539	1
Breadboard	SYB-170 Mini Prototype Breadboard for Arduino	1
Case	Arduino Box for Arduino Uno and Ethernet Shield	1
Power Supply	ARDUINO POWER SUPPLY - 12V 2A - HIGH VOLTAGE (UNIVERSAL)	1

The following figure depicts a diagram of the wiring of the sensors to the microcontroller for the ESC MultiSensor.

Project co-financed by the European Regional Development Fund

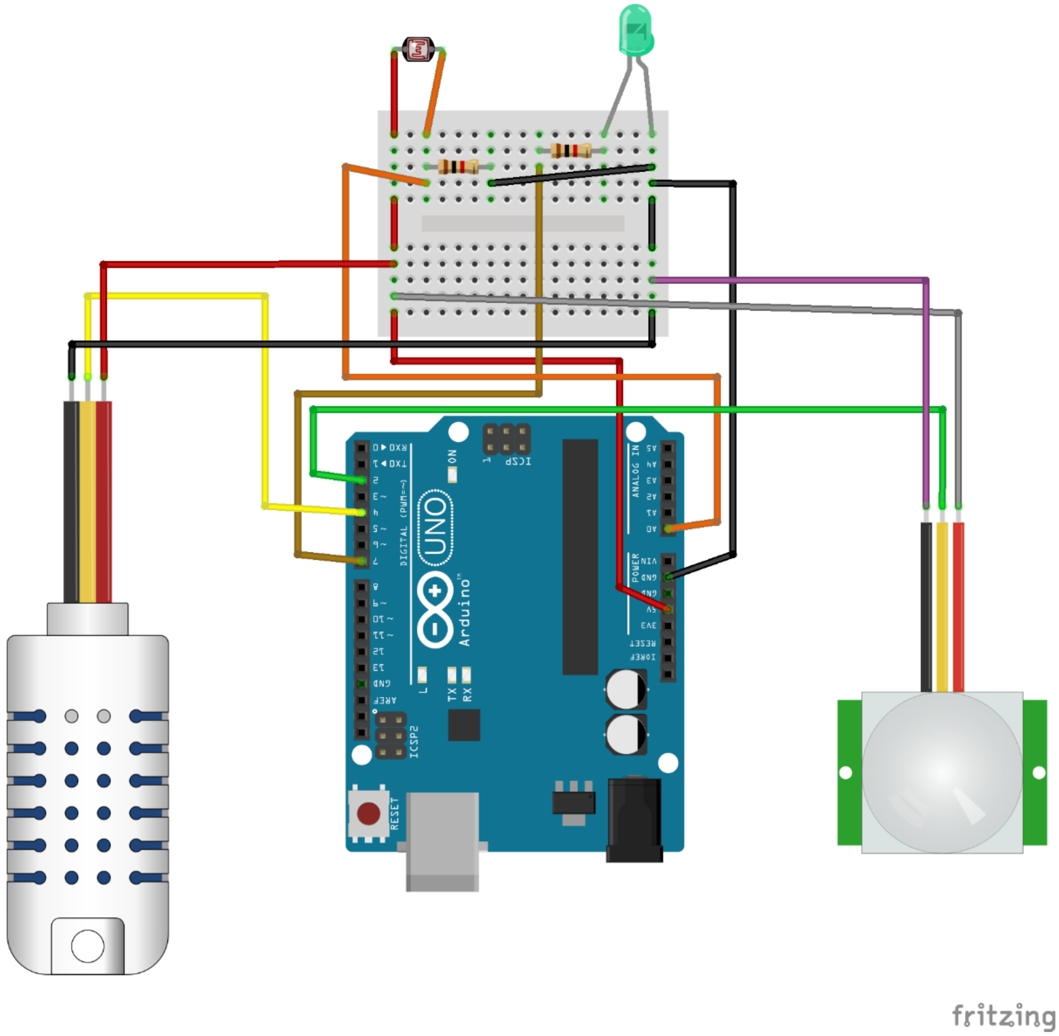

Figure 11: Wiring Diagram of ESC MultiSensor

Figure 2.4: Wiring Diagram of ESC MultiSensor

Project co-financed by the European
Regional Development Fund

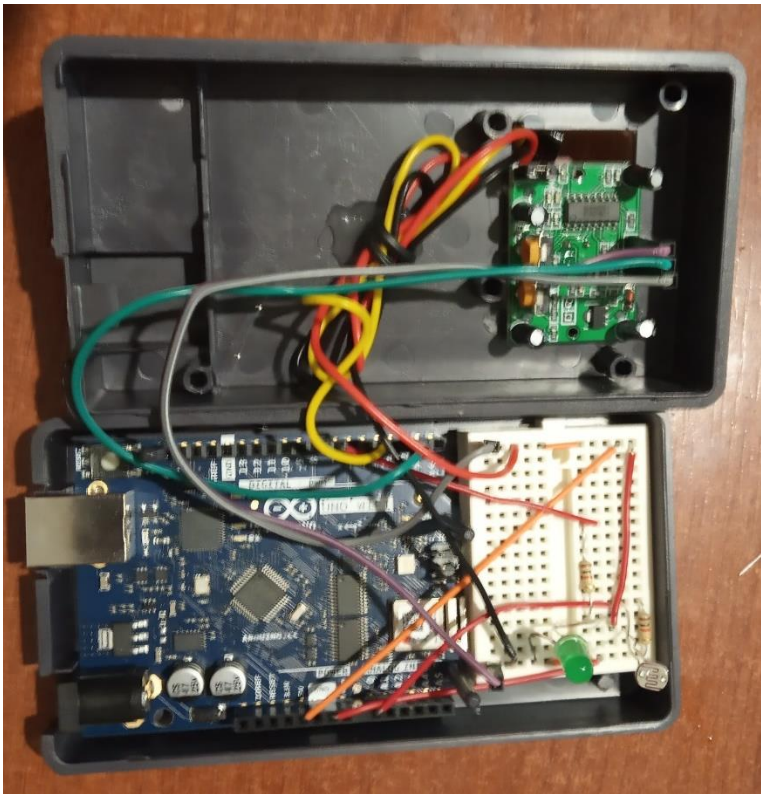

Figure 12: Box interior of ESC MultiSensor

Figure 13: ESC MultiSensor device

MEAZON Motion Detector

The MEAZON motion detector is a device that is able to detect any motion in each range. The device is transferring the data to the Meazon Janus Gateway via ZigBee wireless network. Then, the data sends them to the IoT Platform.

Figure 14: MEAZON Motion Detector

MEAZON Temperature/Humidity

The MEAZON Temperature/Humidity sensor is a device that is able to measure the level of temperature and humidity in a room. The device is transferring the data to the Meazon Janus Gateway via ZigBee wireless network. Then, the data sends them to the IoT Platform.

Figure 15: Temperature/Humidity sensor

Project co-financed by the European
Regional Development Fund

Depth Cameras

In order to preserve the privacy of the persons that are residing in a room and in parallel to acquire useful data for their movements in the space, a depth camera is used. The depth camera is using the Stereo triangulation technique which is an application of stereophotogrammetry where the depth data of the pixels are determined from data acquired using a stereo or multiple-camera setup system. This way it is possible to determine the depth to points in the scene, for example, from the center point of the line between their focal points. The main advantage is that the depth camera can provide a range image where the shape of the objects is clear (including humans) but no detailed characteristics are provided. In this context with the use of Computer Vision algorithms, we can detect humans but it is impossible to identify them.

For the living lab concept, the Intel® RealSense™ D435 Depth Camera[7] was used. The Intel® RealSense™ depth camera D435 is a stereo tracking solution, offering quality depth for a variety of applications. Its wide field of view is perfect for applications such as robotics or augmented and virtual reality, where seeing as much of the scene as possible is vitally important. With a range up to 10m, this small form factor camera can be integrated into any solution with ease, and comes complete with our Intel RealSense SDK 2.0 and cross-platform support. The camera is connected with a mini PC which executes the appropriate algorithms for detecting human presence in a room. The results are send to the IoT Platform.

The algorithm that was created for the motion detection functionality which utilizes the Intel® RealSense™ D435 Depth Camera, was written in the coding language 'Python'. The Python script utilizes the depth stream of the camera through the Python framework "pyrealsense2". The aforementioned framework presents the stream feed frame by frame, while its hyperparameters regarding the resolution as well as the color scheme of the frame can be configured.

The core of the algorithm consists of two main parts: the motion detection, and the blob detection, for both of which the framework "OpenCV" is utilized; a framework that can handle and process images in Python. Firstly, for the initial algorithmic part of detecting motion, the utilized submodule was the contouring detection tool of OpenCV. Each frame of the stream, excluding the initial, was applied Gaussian Blur and transformed to grayscale.

[7] https://www.intelrealsense.com/depth-camera-d435/

Figure 16: Intel® RealSense™ D435 Depth Camera

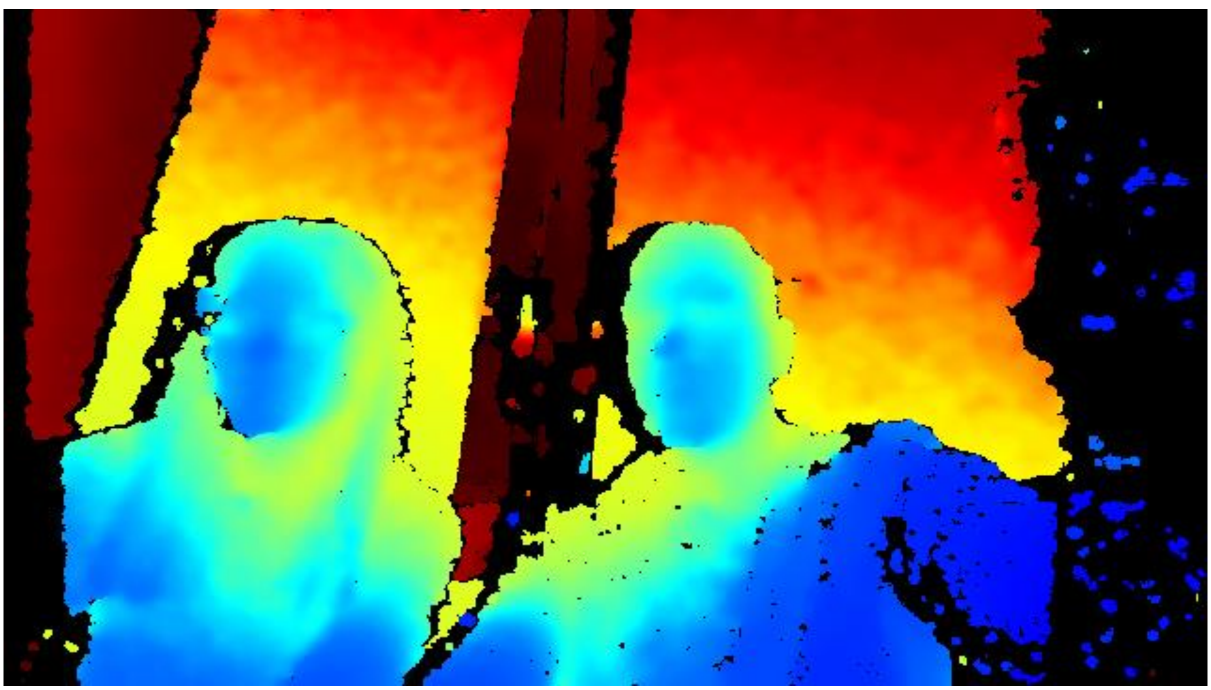

Figure 17: Intel® RealSense™ D435 Depth Camera Screenshot

Project co-financed by the European
Regional Development Fund

After this small preprocessing step, it was then compared with the previous and already preprocessed and saved frame. Their difference, subtracting the current frame from the previous, resulted in a sparse matrix with few pixels of information. The "findContour" tool was then utilized on this matrix product, with a hyperparameter here being the threshold over which an area is considered a contour by the pixel and when is it not.

Therefore, the total of hyperparameters for this module are:

- Contouring Threshold
- Contour Area

This OpenCV submodule produces as a result the bounding boxes of the contours, as well as their center. Intuitively, the contours represent detected movement between frames, thus by detecting contours, movement is detected.

The second part of the algorithm consists of the blob detection. The main idea in this part, lies in the fact that not everything that moved in comparison to its previous frame was an important event, but could be small movement of objects, simple intensity flickers of the camera quality or an unlikely movement of the same person. Thus, detecting the blobs which will represent actual humans is important.

Normally, a preprocessing step is needed before attempting to detect blobs such as the Difference of Gaussian or the Determinant of Hessian methods. However, with the aid of the Intel® RealSense™ D435 Depth Camera, the depth stream performs all of this workload. Therefore, with a simple usage of OpenCV's submodule "SimpleBlobDetector", all blobs can be pinpointed which are already represented in the depth stream's color scheme.

The hyperparameters of this functionality are:

- Blob size
- Minimum distance between blobs
- Minimum Blob Area & Minimum Inertia Ratio
- Maximum Blob Area & Maximum Inertia Ratio
- Convexity & Circularity Filters

After detecting all blobs, only a few of them are kept and the rest are discarded. This functionality is necessary because the depth camera may detect scenes inside the frame which resemble blobs, however are neither humans, nor do they move. Thus, the two parts of the algorithm are combined. Only blobs whose center lie inside a contour bounding box are kept. This means that each specific blob that is kept, has pixels that participated inside recent movement in the frames. Once per minute, the script sends a message to the designated MQTT topic containing the information regarding whether there has been motion inside the monitored room or not.

Project co-financed by the European Regional Development Fund

3.1.2.4 Servers

The collection and storage of all the data of the installed IoT devices as well as the asset management requires the installation of resourceful hardware. The hardware is installed at the ISI premises and is able to support the connection of various devices as well as the storage of Gigabytes of telemetry data.

Two servers were installed, two DELL Server PowerEdge T330 with the following characteristics:

- Case: Tower
- CPU Vendor: INTEL
- CPU: Intel Xeon E3-1220 v6(3.00GHz)
- Memory Speed: 2400MHz
- Memory: 32GB
- Hard Drives: 4x 1TB / 7200rpm
- Hot swap: Yes
- RAID: Raid 1
- DVD-RW
- 2 x Ethernet 10 / 100 / 1000 Adapters
- PSU: 2
- Hardware RAID Controller: Yes
- Guarantee: 60 months
- Guarantee Type: On Site - Repair Next Business Day

For the support of virtualisation, the bare-metal hypervisor VMware ESXi was installed in both servers and Linux VMs were used for the installation of the IoT Platform.

Project co-financed by the European Regional Development Fund

Figure 18: The two DELL PowerEdge T330 servers

3.1.2.5 IoT Platform

With reference to IoT Platform, the open-source Thingsboard Community Server was used.

Thingsboard

ThingsBoard[8] is an open-source IoT platform for data collection, processing, visualization, and device management. It enables device connectivity via industry standard IoT protocols - MQTT, CoAP and HTTP and supports both cloud and on-premises deployments. ThingsBoard combines scalability, fault-tolerance and performance so you will never lose your data.

Basic Features:

- Telemetry Data Collection: Collect and store telemetry data in reliable way, surviving network and hardware failures. Access collected data using customizable web dashboards or server-side APIs.

[8] https://thingsboard.io/

Project co-financed by the European Regional Development Fund

- Multi-tenancy: Support multi-tenant installations out-of-the-box. Single tenant may have multiple tenant administrators and millions of devices and customers.
- Data Visualization: Provides 30+ configurable widgets out-of-the-box and ability to create your own widgets using built-in editor. Built-in line-charts, digital and analog gauges, maps and much more.
- Horizontal scalability: Amount of supported server-side requests and devices increase linearly as new ThingsBoard servers are added in clustering mode. No downtime, server restarts or application errors.
- IoT Rule Engine: Process incoming device data with flexible rule chains based on entity attributes or message content. Forward data to external systems or trigger alarms using custom logic. Configure complex notification chains on alarms. Enrich server-side functionality or manipulate your devices with highly customizable rules. Define your application logic with drag-n-drop rule chain designer.
- Fault-tolerance: All ThingsBoard servers are identical. No master-workers or hot standby. Node failure is automatically detected. Failed nodes can be replaced without downtime. Persisted data is replicated using reliable NoSQL database.
- Device Management: Provides ability to register and manage devices. Allows to monitor client-side and provision server-side device attributes. Provides API for server-side applications to send RPC commands to devices and vice-versa.
- Security: Supports transport encryption for both MQTT and HTTP(s) protocols. Supports device authentication and device credentials management.
- Asset Management: Provides ability to register and manage assets. Allows to provision server-side asset attributes and monitor related alarms. Ability to build hierarchy of entities using relations.
- Customization and Integration: Extend default platform functionality using customizable rule chains, widgets and transport implementations. In addition to MQTT, CoAP and HTTP support, ThingsBoard users can use their own transport implementations or customize behaviour of existing protocols.
- Alarms Management: Provides ability to create and manage alarms related to your entities: devices, assets, customers, etc. Allows real-time alarms monitoring and alarms propagation to related entities hierarchy. Raise alarms on device disconnect or inactivity events.
- 100% Open-source: ThingsBoard is licensed under Apache License 2.0, so you can use any it in your commercial products for free. You can even host it as a SaaS or PaaS solution.

Project co-financed by the European
Regional Development Fund

- Microservices or Monolithic: Supports monolithic deployment for getting started or small environments. Provides ability to upgrade to microservices for high availability and horizontal scalability.
- SQL, NoSQL and Hybrid database: Supports various database options and ability to choose where to store main entities and where to store telemetry data

Figure 19: Example of Thingsboard Dashboard

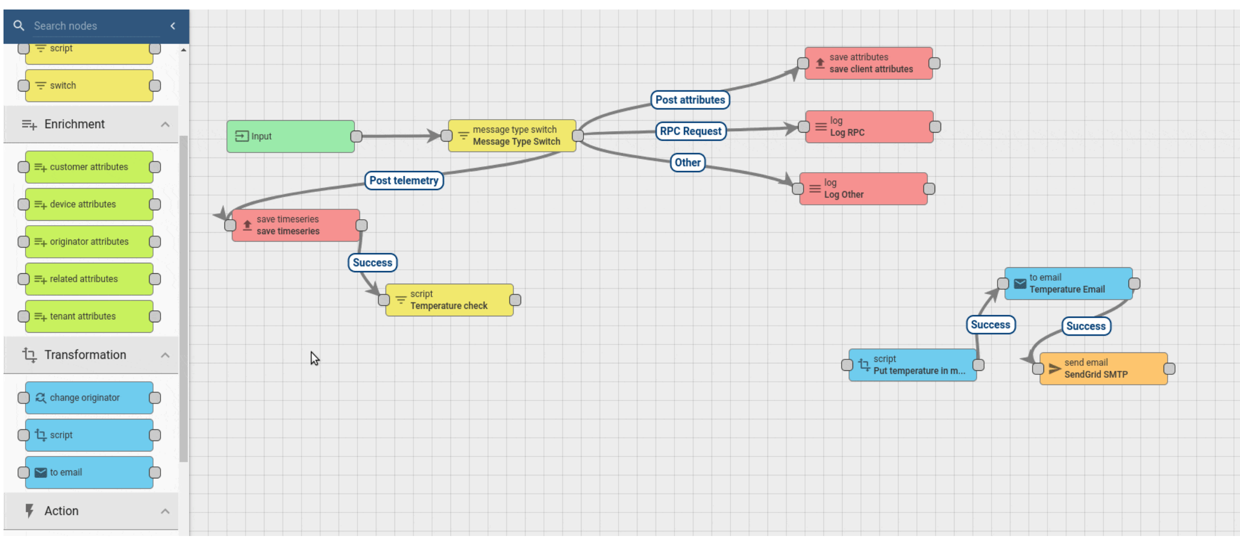

Figure 20: Example of Thingsboard IoT Rule Engine

3.1.3 System Integrated Architecture

One of the most important branches of IoT which aided vastly in its evolution is the Industrial Internet of Things. IIoT maintains a great impact on today's manufacturing technologies and its contributions shaped the Fourth Industrial Revolution. Ever since, there have been initiatives and attempts in order to create standards and models for the variety of available IIoT architectures such as "Industry 4.0" and the "Industrial Internet Consortium (IIC)"[9] which helped develop the "Industrial Internet Reference Architecture (IIRA)"[10]. Nowadays, IIoT system implementations have specific architectural patterns such as the three-tier architecture pattern which is also implemented in this work. This pattern consists of three main components which manage data and control mechanisms among the flow in the pipeline. As also shown in Figure 1, the three tiers are:

- The *edge tier,* where data are collected from the edge nodes through the proximity network. Important factors of the tier are the breadth of distribution, location, governance scope and the nature of proximity network
- The *platform tier,* which manages the control commands and distributes them accordingly amongst the tiers. This tier is also responsible for the analysis and the redirection of data flows with the corresponding services
- The *enterprise tier*, which is responsible for the implementation of applications which are specific to a domain and end-user or operation specialist interfaces such as decision support systems

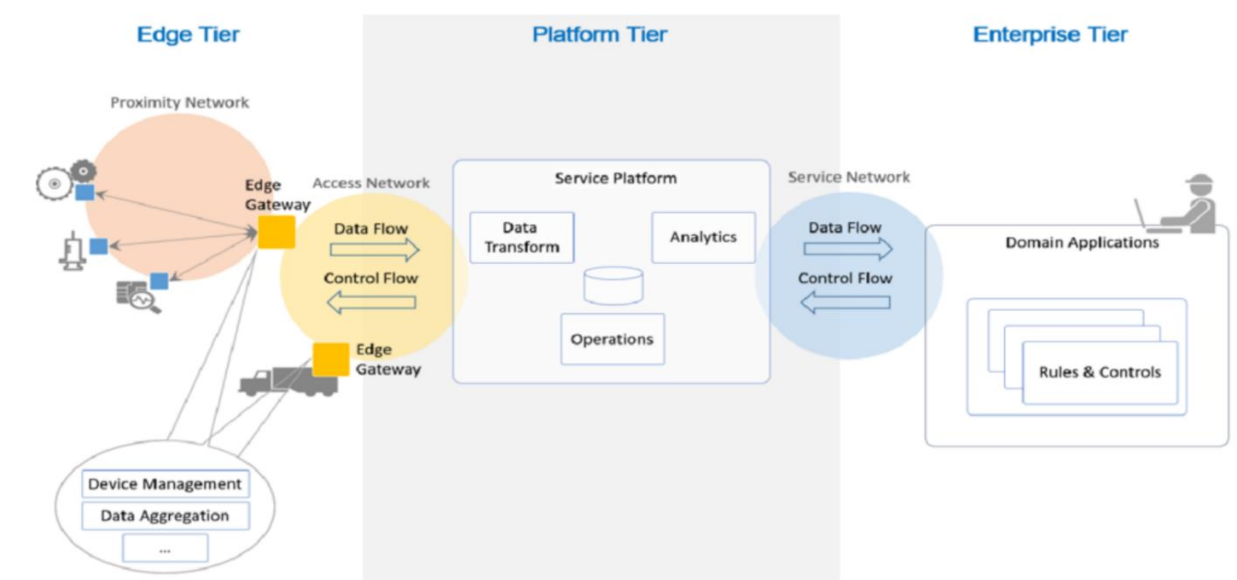

Figure 21: Three-Tier IIoT System Architecture for IIRA

[9] M. Hankel and B. Rexroth, "The reference architectural model industrie 4.0 (rami 4.0)," ZVEI, April, 2015
[10] S.-W. Lin, B. Miller, J. Durand, R. Joshi, P. Didier, A. Chigani, R. Torenbeek, D. Duggal, R. Martin, G. Bleakley et al., "Industrial internet reference architecture," Industrial Internet Consortium (IIC), Tech. Rep, 2015

Project co-financed by the European Regional Development Fund

The way that tiers communicate with each other, depends on these following networks that connect them:

- The *proximity network*, which connects all sensors and devices, meaning the components of the edge nodes represented as clusters
- The *access network*, which allows the connection of data and control flows between the edge tier and the platform tier
- The *service network,* which allows the connection between all services that take place in the platform and the enterprise tiers as well as the services inside each tier. The internet itself is also used as a service network, or in most occasions, an overlay private network which guarantees an enterprise grade of security

The energy efficient living lab concept that is implemented in I.S.I.'s IIoT infrastructure was built around an IIRA driven architecture. After meticulous consideration of the available IoT platforms, Thingsboard was chosen as the selected IoT platform for the implementation. The facilities of the research institute consist of five different and independent offices, located in Patras Science Park. Due to related restrictions it was not possible to place smart energy meters in the central energy infrastructure. However, many other sensors were installed across the offices.

Firstly, the smart plugs were a product of the Greek SME company MEAZON which provided open telemetry protocols, as well as Sonoff, a Chinese company with open source firmware. Other sensors included custom Arduino devices which were used for measuring temperature, sensitivity, light, as well as room population through motion detection. The intercommunication of all the aforementioned devices was enabled through the existing Wi-Fi and Ethernet networks. In order to connect the MEAZON smart plugs with the selected IoT platform, a ZigBee gateway was installed in every office, in proximity with all the plugs that the room contains. For the motion detection, an Intel RealSense depth camera was installed in order to identify human presence and capture the number of people that coexist in the room in the monitored area. The cameras and a corresponding embedded network are connected in order to process each image and extract the information needed in order to process the actual number of monitored humans and send said information to the central platform as well. The MQTT (MQ Telemetry Transport)[11] protocol is used in order to handle the latency that is caused by the heterogeneity of the ready off the shelf devices and custom built hardware ensemble.

The proposed architecture is heavily influenced by the IIRA three-tier model where the platform and the business layer are represented by the Thingsboard platform. In Figure 22, it is shown how the I.S.I. Living Lab was implemented and connected. At the edge tier of I.S.I.'s Living Lab, the Wi-Fi, the Ethernet and the ZigBee gateways allow the devices to communicate with each other as well as with the Thingsboard platform designated server, through the aforementioned MQTT

[11] http://mqtt.org/

Project co-financed by the European Regional Development Fund

protocol. In the platform tier, there exist two MQTT brokers. The first is utilized by the Thingsboard platform as endpoint for the input and the output of the data. The second, is an Eclipse Mosquito[12] broker that runs on a Node Red[13] service and it is used for the communication of the devices that do not support the message content as it is defined by selected the IoT protocols.

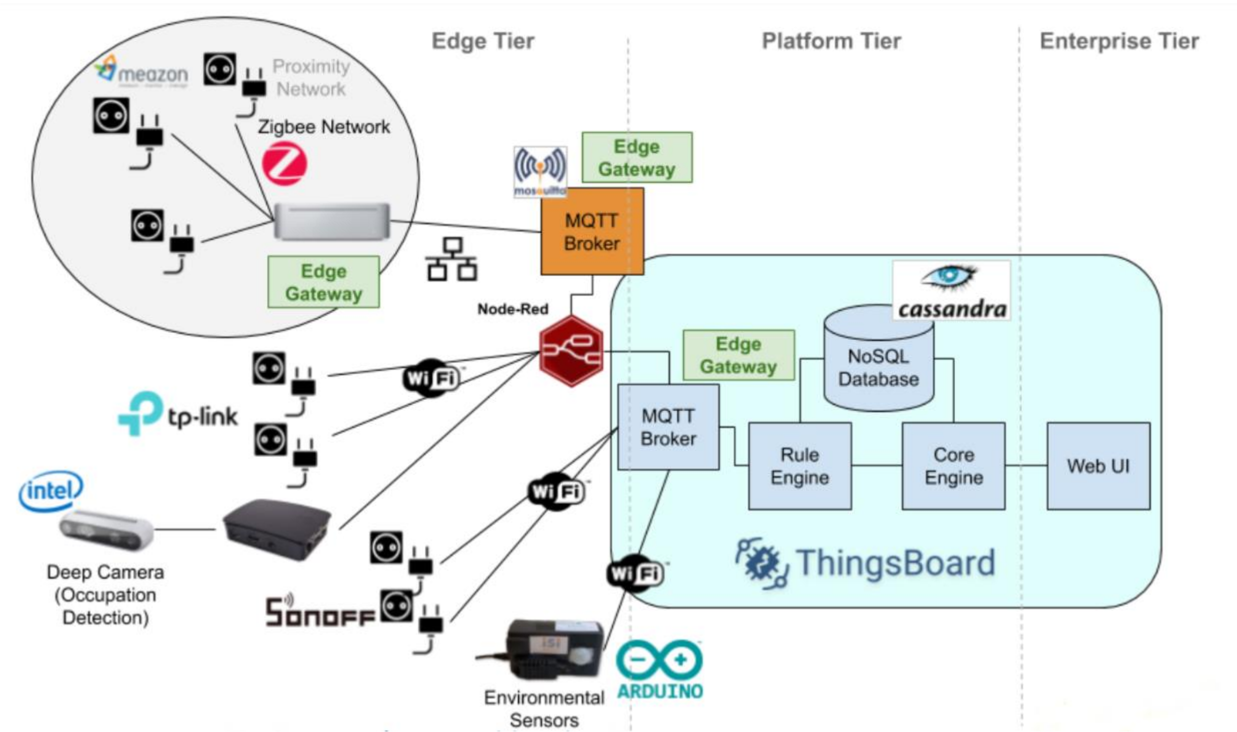

Figure 22: IIRA-based Three-Tier IIoT System Architecture at the ISI living lab

If a device from the installed infrastructure allows for the programming of its firmware the native MQTT is used for the exchanging of the data which contain telemetry and platform commands. For devices that maintain their own message exchange protocol, the messages are converted to Thingsboard complying protocols through a NodeRed broker service. For the Data Transformation and Analytics module, where data streams are filtered and transformed, the in-built Rule Engine is used. After the transformation of the incoming data, they are transferred to the Thingsboard module that is responsible for the storage to local databases, with the aid of Apache Kafka[14]'s stream management services. The telemetry data as well as the device

[12] https://mosquitto.org/
[13] https://nodered.org/
[14] https://kafka.apache.org/

Project co-financed by the European Regional Development Fund

configurations are saved in a Cassandra No-SQL database[15]. Also, as the IIRA indicts, Thingsboard provides modules for asset management and user authorization. Lastly, at the enterprise tier, dashboards and web interfaces allow the visualization of the data as well as the interaction of the end-user with the provided metrics, as shown in Figure 23.

Figure 23: Screenshot of Thingboards top level Dashboard

For the analysis of the accumulated big data, Apache Kafka is integrated in the pipeline as a means to provide datastreams to outgoing applications. Also, batch processing is allowed through a NoSQL database so data can be recalled in bulk as per request of an external application. Apache Spark[16] is also utilized in order to perform a variety of functions for data analytics on the incoming stream of data, with Spark's library "MLlib". With these functions of "MLlib", classifiers can be trained which identify patterns and events on the monitored IIoT systems. The data processed by the Thingsboard platform can either be done with batch processing using the Thingsboard storage, with batch processing using Spark storage, or with stream processing directly from Apache Kafka.

[15] http://cassandra.apache.org/
[16] https://spark.apache.org/

3.1.4 Communications System Architecture

For the purpose of interconnection of the System Architecture's components, the main computer network of ISI was used. In order to understand the connectivity of the devices, the following figure depicts the network topology of ISI's network.

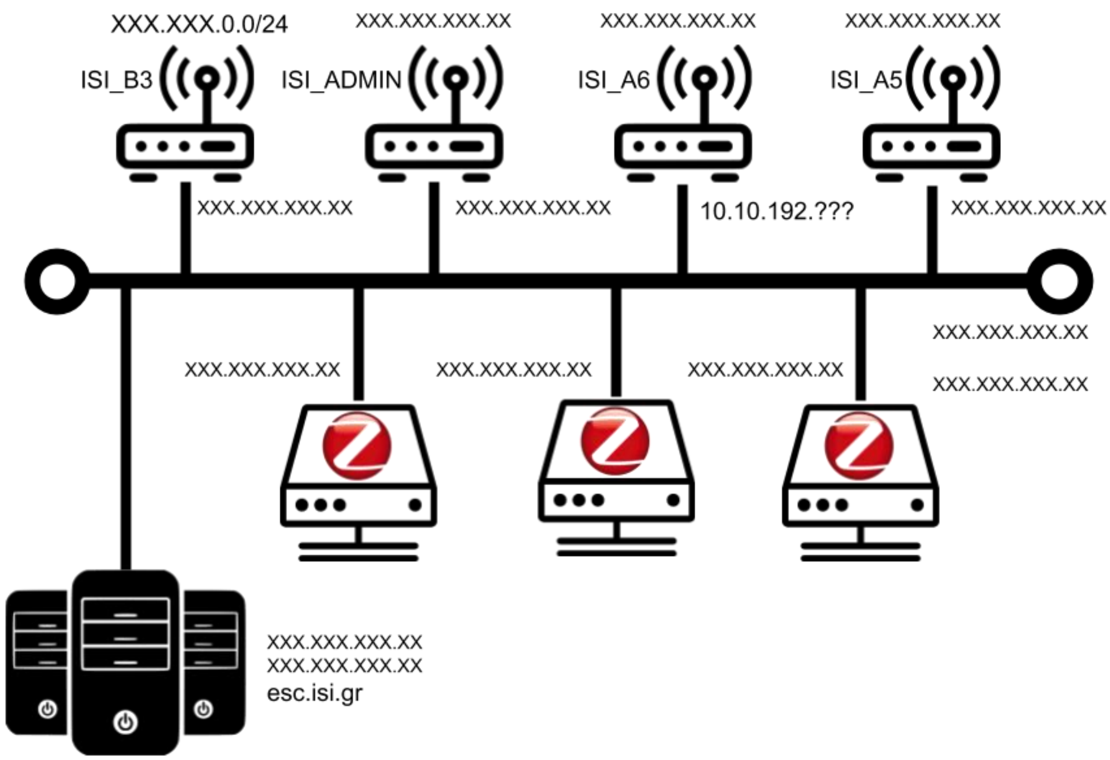

Figure 24: ISI Main Network Topology

As depicted in the figure above, the ISI network consists of one Ethernet network connecting all the offices. The network supports two subnets, an internal one and an external which is accessed by Internet. On this Ethernet network there are four Wi-Fi Routers/Access Points that provide Wi-Fi network to the offices. For the purpose of Living Lab, three gateways (MEAZON Janus Gateway) that are connected to the Ethernet and provide ZigBee wireless network. Finally, the main IoT platform is installed on the servers that are connected directly to ISI's main Ethernet network. The VM has both internal and external network IPs in order to connect to all the devices while the platform can be accessible from the Internet.

All the MEAZON devices are connected via ZigBee to the MEAZON Janus Gateway and send data to the IoT platform. All the rest of the devices are connected via Wi-Fi to one of the Wi-Fi Routers/APs.

Project co-financed by the European Regional Development Fund

3.1.5 Meeting User Requirements

After thorough analysis of the user requirements, they can be divided into two bigger categories that pertain to the project. The first vast category is the "Common User Requirements", a category which can be applied to most of the pilot deployments. Examples of such user requirements that comprise the category and are included in it are: the replicability that is needed in order to validate any potential spread of a pilot deployment's proposed solutions, the promotion of innovative and cutting edge technologies that accompany the requirement of replicability, the utilization of smart devices such as sensors, actuators and embedded devices that are used for data aggregation and accumulation, the experimentation that takes place at the edge level, open source platforms and cloud based services that the project may utilize in its data pipeline, devices that maintain fast-paced installation procedures with a focus on simplicity, the durability of the aforementioned devices which should preferably be easy to maintain and inspect, the respect of legal framework issues and general privacy complying activities, low intrusion in terms of interrupting the aesthetics of the building where a device is installed, the quality of the information that is saved and transferred through the integrated network, and the overall costs of the devices and systems of such a network.

The second category of user requirements includes issues of building energy efficiency. Components of this requirement category are control strategies for different heating and cooling structures, the storage of the monitored data in databases and servers for further big data analysis, a real-time interface which can inform about historical data and relevant other features, the identification of the number of people inside a room, the overwatch of environmental and structural characteristics of the building, the measurement of the effects of thermal networks on the general consumption of the building, the deployment of heterogeneous devices within a network and the management of their interconnections, the aggregation of electrical consumption across a number of various buildings, the meticulous study and installment of renewable energy systems, as well as a tool that can provide a means of intervening the planning of the incorporated systems.

The pilot framework was developed with respect to the idea of creating smarter labs whose workers can finetune their hyperparameters and optimize energy consumption as well as the comfort of people working inside such labs. Hence, the pilot project meets a certain number of the aforementioned user requirements:

- A network of multiple types of sensors is deployed inside the lab, in order to monitor
    - energy consumption
    - humidity
    - light
    - human presence

Project co-financed by the European Regional Development Fund

- vibrations and sounds
- Door and Window opening

and provide information and analytics accordingly. The system in conjunction should be able to provide solutions for control and actuation in order to efficiently optimize energy, maintain user's comfort, test use cases of control and actuation, and utilize the provided edge intelligence.

- After the installment of the proper equipment into the according spaces, each office will be able to maintain an energy profile, said profile can be associated with external climatic conditions, data analytics can be performed on the collected data aggregations that can produce behavioral characteristics of each profile, and aim for energy efficient behaviour of the occupants in the long run of the living lab.

3.2 Deployment

3.2.1 Installation of Equipment

All the equipment mentioned in Bill of Materials was installed according to the Deployment Plan.

Totally 52 devices has been installed for monitoring 152 physical variables. The following pictures shows examples of the equipment installation.

Project co-financed by the European Regional Development Fund

Figure 25: Deployment of TPLink and Sonoff smart plugs

Figure 26: Deployment of TPLink and MEAZON smart plugs

Figure 27: Deployment of Sonoff smart switch

Figure 28: Deployment of ISI ESC Multisensor

Figure 29: Deployment of Intel ReaslSense depth camera

Project co-financed by the European Regional Development Fund

3.2.2 IoT Platform Installation and Configuration

The next step was the installation and configuration of ThingsBoard Platform. This step included the creation of an Ubuntu 18.04 LTS Server Virtual Machine and the installation of ThingsBoard Community Server and the Cassandra Database. Also, the Eclipse Mosquito MQTT Server was installed along with Node-Red service.

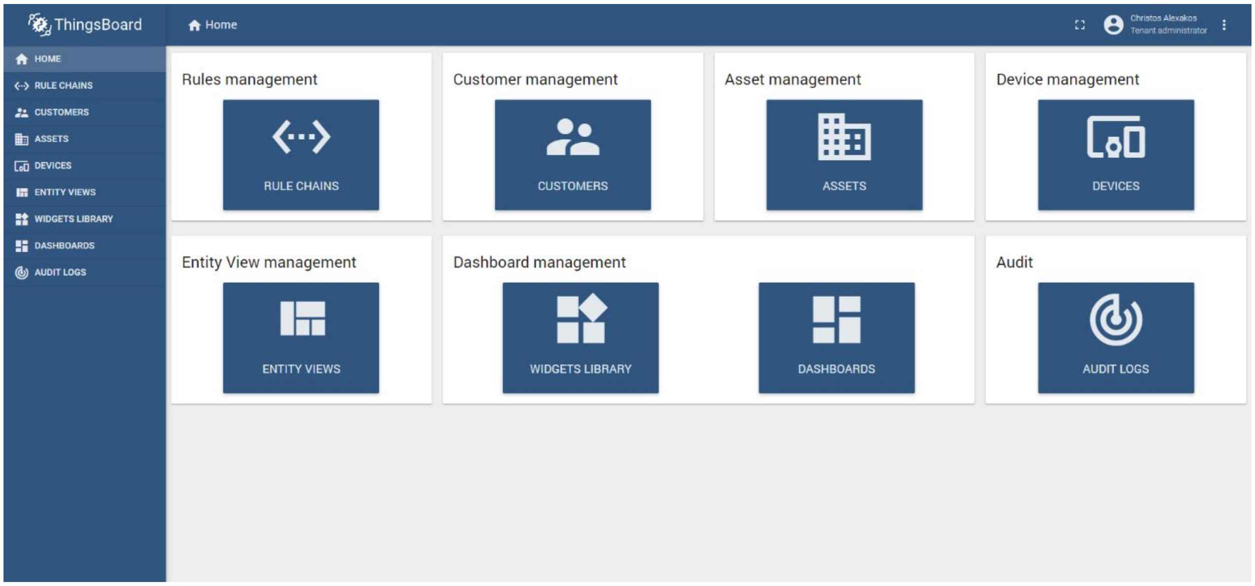

Figure 30: Screenshot of the Home page of the installed ThingsBoard

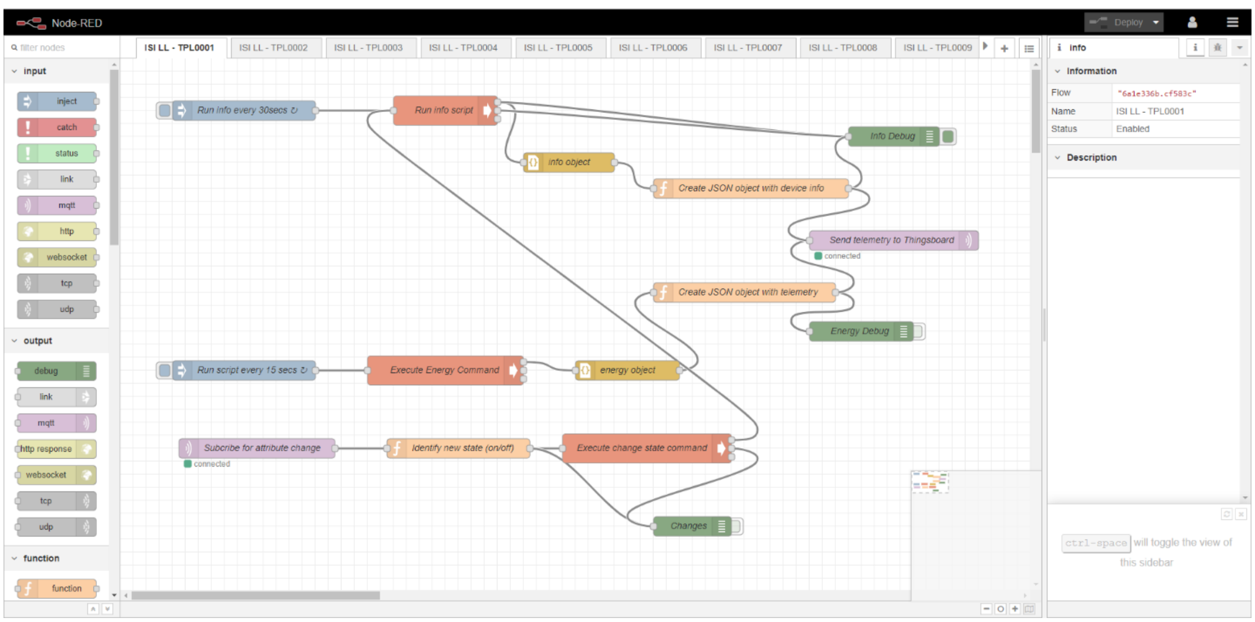

Figure 31: Screenshot of the installed Node-Red interface

Project co-financed by the European Regional Development Fund

Afterwards, an asset scheme depicting the rooms and spaces of the real deployment environment has been configured.

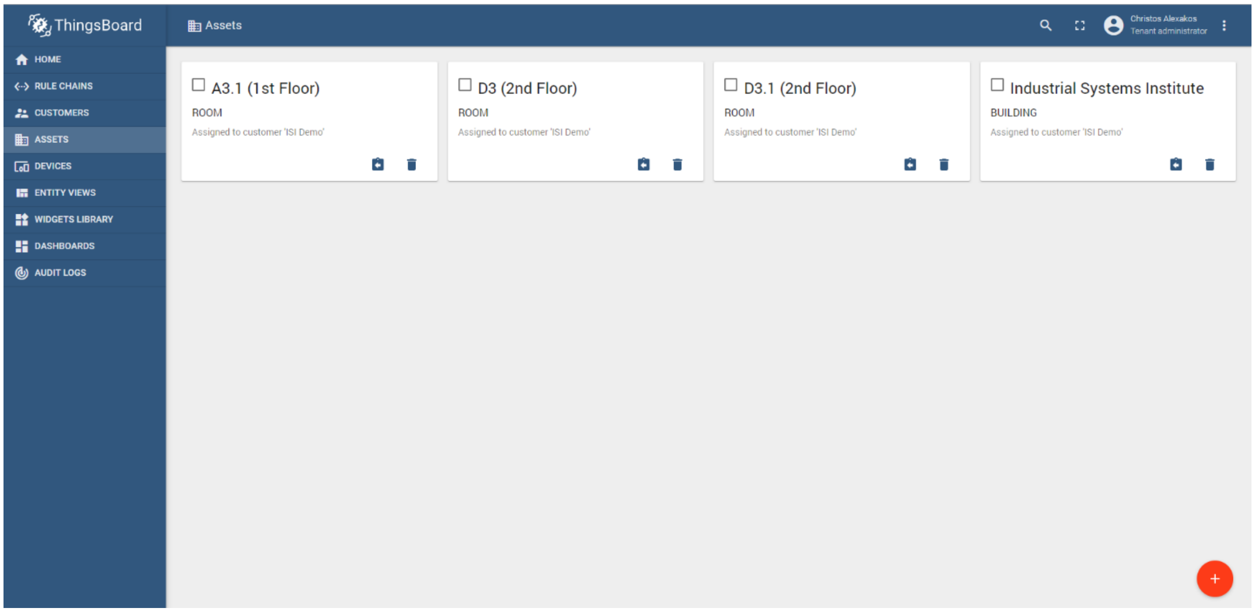

Figure 32: Screenshot of the Assets page of the installed ThingsBoard

3.2.3 Device Integration to IoT Platform

After finishing the parameterization of the IoT Platform, the next step included the integration of the devices in the ThingsBoard platform in order to send telemetry data and receive commands from/to the device.

The configuration was made by adding each device in the ThingsBoard platform and assign it to an asset.

Project co-financed by the European Regional Development Fund

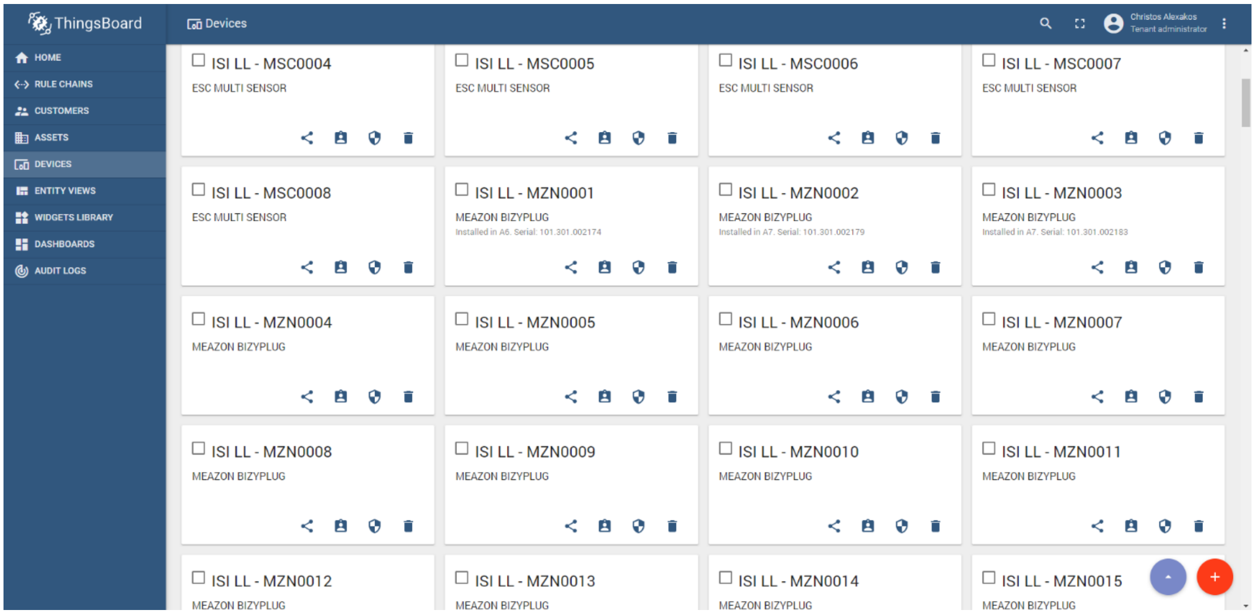

Figure 33: Screenshot of the Devices page of the installed ThingsBoard

3.2.4 Data Presentation

After the integration of the devices to the ThingsBoard platform, each device started sending telemetry and status data.

ThingsBoard Dashboards

In order to present this data to the user, dashboards using tables and graphics were created in the ThingsBoard.

Project co-financed by the European Regional Development Fund

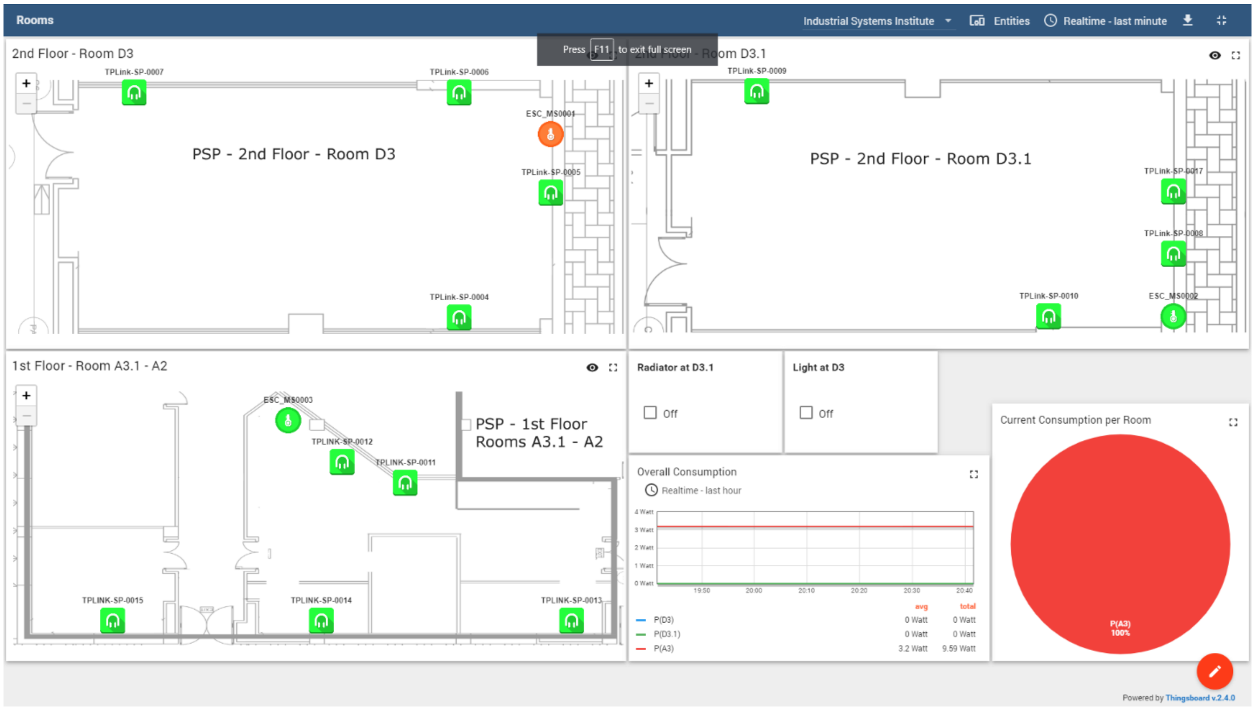

Figure 34: Screenshot of the Main Dashboard of the installed ThingsBoard

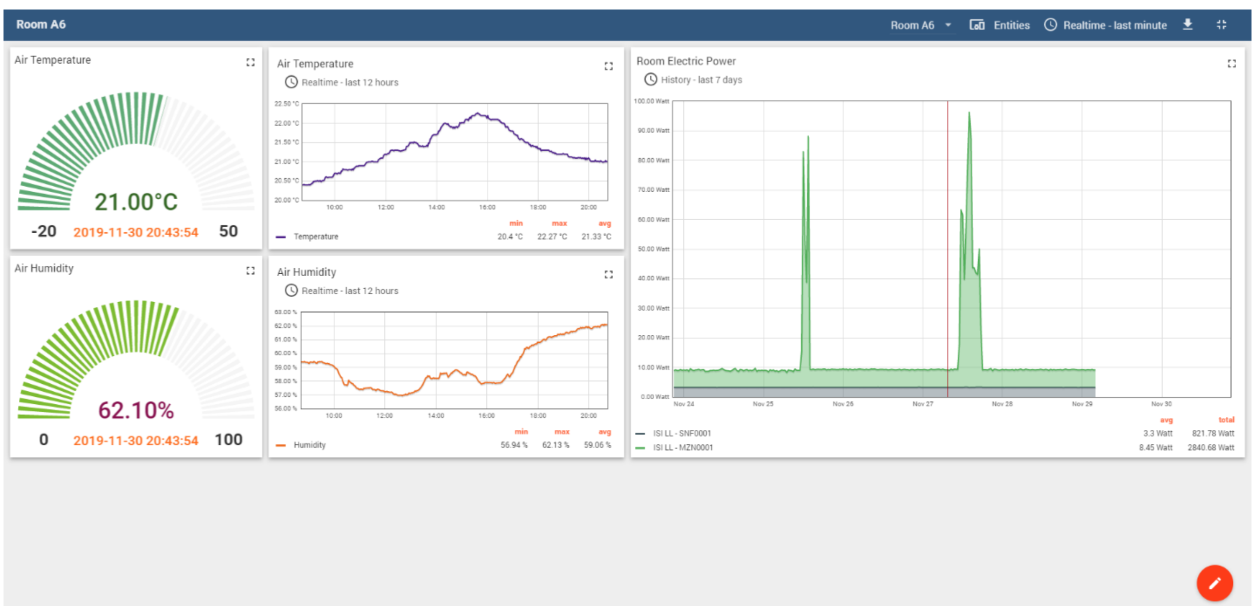

Figure 35: Screenshot of the Room A6 Dashboard of the installed ThingsBoard

Project co-financed by the European
Regional Development Fund

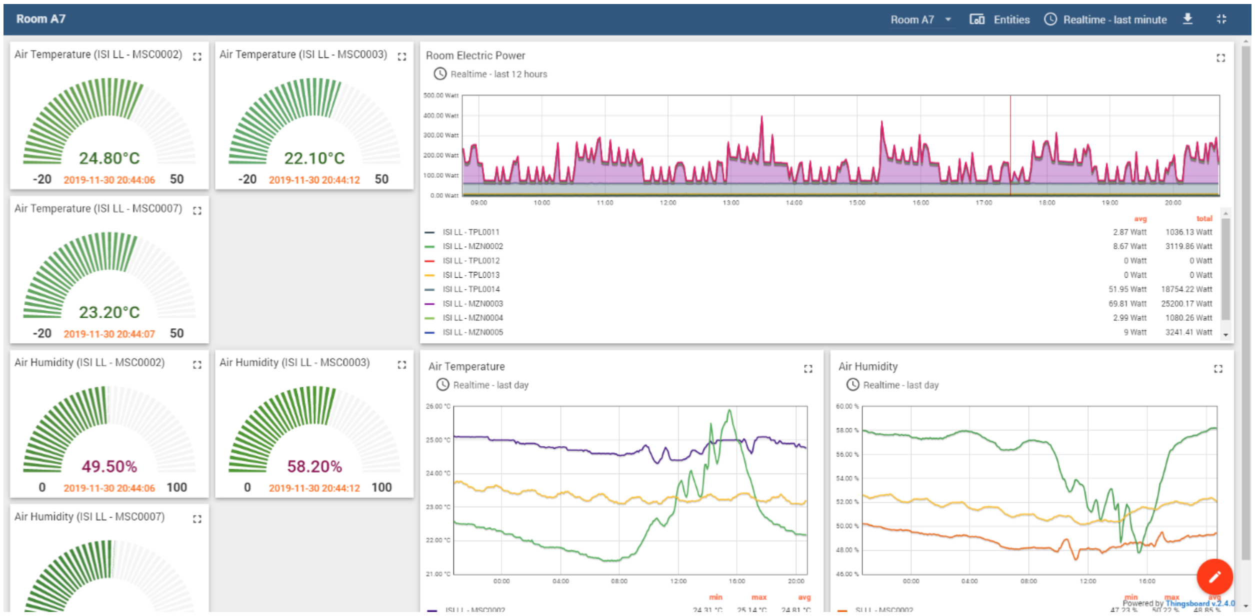

Figure 36: Screenshot of the Room A7 Dashboard of the installed ThingsBoard

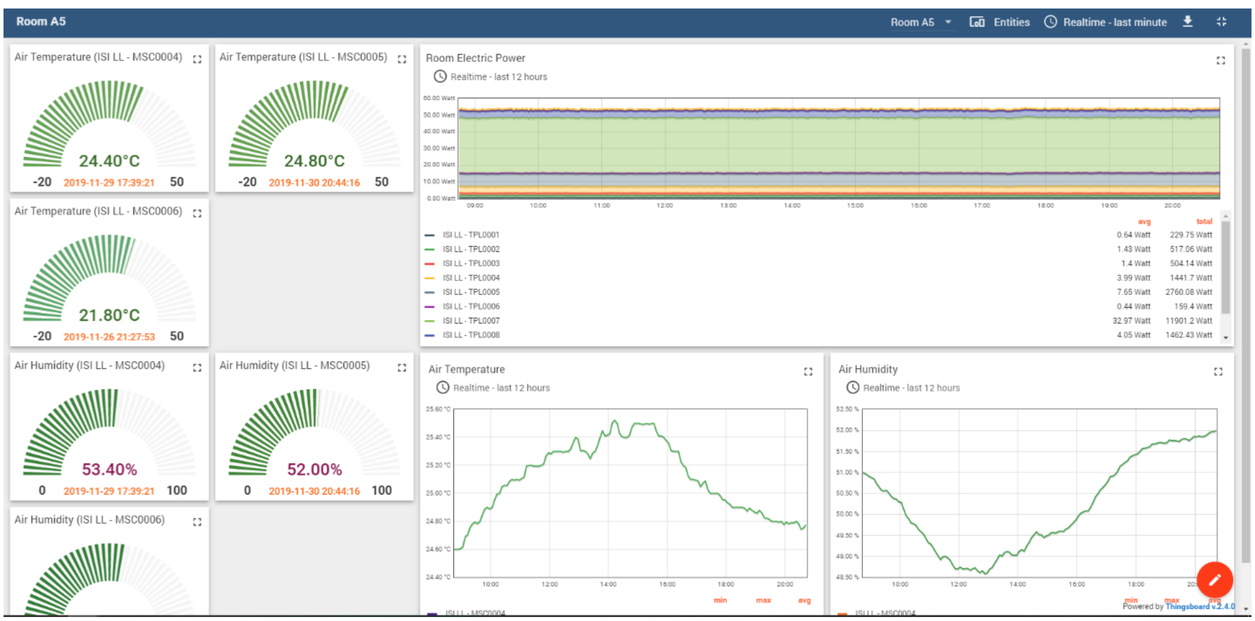

Figure 37: Screenshot of the Room A5 Dashboard of the installed ThingsBoard

Configuration of Rules

The IoT Rule Engine of ThingsBoard was used in order to automate the control of some devices, the pre-process of telemetry data and the generation email alarms about the state of the overall Living Lab environment.

Project co-financed by the European Regional Development Fund

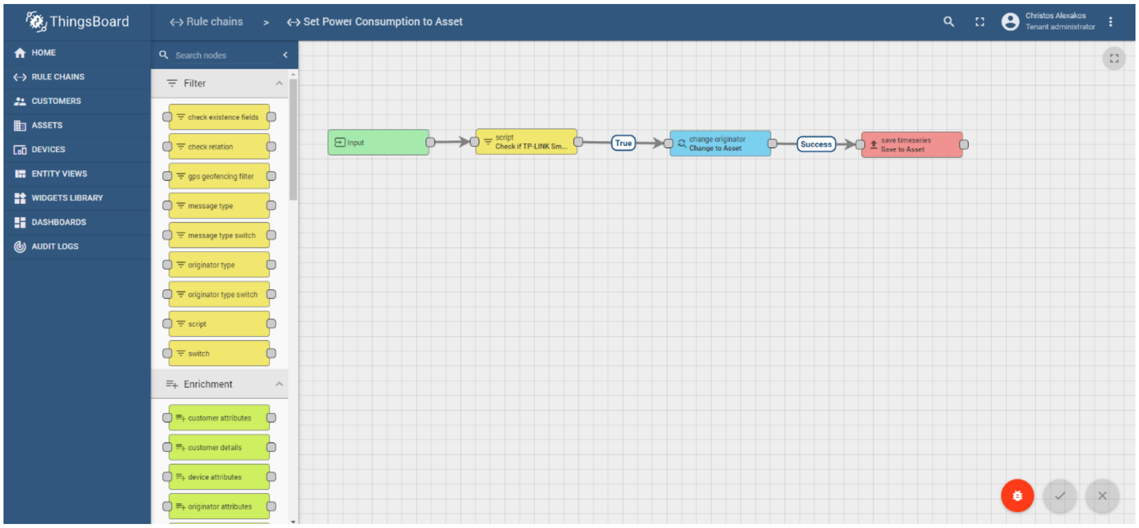

Figure 38: Screenshot of a rule of the installed ThingsBoard which accumulates the power consumption of a room

Multi-Monitor Panel

For the purpose of continuous monitoring of the several variables a monitoring panel consisting PC Monitors and Rasberry Pi[17] was installed. Using this multi monitor panel, the users are able to watch real time and interact with the overall Living Lab system.

Figure 39: ISI's Living Lab Multi-Monitor Panel

[17] https://www.raspberrypi.org/

Project co-financed by the European Regional Development Fund

4 Conclusion

The pilot of ESMARTCITY project for smart buildings was successfully deployed at the ISI's premises at the Patras Science Park Building.

The pilot deployment will be used as a practical paradigm for applying an IIoT (Industrial Internet of Things) reference architecture -in our case the IIRA- utilizing open source software in combination with industrial products that can be found in the market. Through this pilot, we demonstrate that the implementation of reference IIoT architectures using open-source software is not only feasible, but in fact a highly scalable and flexible approach for IIoT implementation, combining custom and COTS (Commercial off-the-shelf) hardware. The deployment included the installation of fifty two (52) devices in five (5) rooms for monitoring of 152 physical variables. Furthermore, there is the automated control of 5 smart switches.

5 Future Work

An extension of this Living Lab is envisaged as future work, mainly with reference to the Enterprise Layer, where an additional platform for big data analytics including Apache Spark will be deployed in a cloud computing infrastructure. This platform will enable the users to proceed with complex data analysis tasks, and use the results to create new rules which will feed the IoT platform rule engine. More importantly, we are looking at distributing the data analytics intelligence capabilities to reduce the reliance on single-points of failure and increase the robustness, responsiveness and scalability of the system. To this end, we investigate the implementation of multi-tier fog-based analytics, in the form of a hierarchy of individual AI-enabled boards at the edge (e.g. Google Coral or Raspberry Pis with AI hats working at the room level), local analytics using Spark on Raspberry Pi nano-clusters (e.g. at floor level), and global analytics using the cloud Apache Spark configuration described in this paper (e.g. at building level).

6 List of Figures

Project co-financed by the European Regional Development Fund