How to evaluate an Internet of Things system: models, case studies, and real developments

Sabrina Sicari[*‡], Alessandra Rizzardi[*], Alberto Coen-Porisini[*]

[*]Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria,

via Mazzini 5 - 21100 Varese (Italy)

[‡]Corresponding author

[*]Email: {sabrina.sicari; alessandra.rizzardi; alberto.coenporisini}@uninsubria.it

Abstract

The paper proposes the use of Node-RED, a flow-based programming tool targeted to Internet of Things (IoT), along with a series of case studies related to different IoT contexts, which demonstrate Node-RED's potentialities and outcomings towards the realization of well-structured IoT environments. The analyzed applications potentially include a wide range of domains, ranging from smart cities, smart buildings, smart homes/offices, smart retailing, to smart transportation, smart logistics, smart agriculture, smart health, military scenarios, and so on. The motivations behind the presented work are related to the fact that IoT application fields usually involve the same technologies and communication protocols, which are frequently adopted for totally different purposes. Issues such as systems' interoperabiliy, scalability, security and privacy naturally emerge, due to the huge amount of heterogeneous devices acting in the IoT environment itself and to the wireless nature of information transmissions. As a consequence, it is fundamental to dispose of adequate tools for supporting developers in design the network architecture and messages' exchange, in order to realize efficient and effective IoT network infrastructures.

Keywords

Internet of Things, Application Case Study, Node-RED, Flow-based programming

I. Introduction

In the last decades, the spreading of the Internet of Things (IoT) paradigm interested a wide range of application domains [1], embracing smart cities, smart buildings, smart homes/offices, smart retailing, smart transportation, smart logistics, smart agriculture, smart health, military scenarios, and so on. The IoT revolution has an impact on various aspects of people's every-day life, as well as in business, mainly in relation to the emerging industry 4.0 initiatives [2].

The basic idea behind the IoT paradigm is the possibility of acquiring heterogeneous kinds of information from the environment where IoT devices are placed in. Such devices embed both sensing and actuating capabilities, which make them "smart" and enable them to interact with the surrounding environment. Such features allow the IoT system to share throughout the network and the Internet a huge amount of information, which can be used to provide customized services to the interested users [3].

To do this, numerous technologies and communication protocols must be integrated and orchestrated, in order to enable the realization of efficient IoT infrastructures and to regulate the information exchange process. Also, several issues emerge, which mainly regards the management of such a huge amount of data, interoperability, scalability, security and privacy [4]. As a consequence, a tool or simulator or testing-platform, for supporting the realization of such IoT infrastructures from the design towards the development phase is needed, with the following main goal: representing all the components acting within an IoT environment, so as to give an overview of the whole IoT system before real deployment.

To fill such a gap, we propose the use of Node-RED [5]. Node-RED is an open project, initially developed by IBM, which proposes a flow-based programming tool. The behavior of the application which must be designed/developed can be represented as a network of black-boxes, which may communicate with each others and regulate the flow of the information within the envisioned system. Moreover, the visual representation provided by Node-RED supports the developers in better understanding the interactions happening within the whole IoT network architecture as well as the messages' passing. In fact, different kinds of entity can be modeled, in particular both hardware (e.g., sensors) and software (e.g., services). Such a feature perfectly fits the IoT concept.

To demonstrate the usability and feasibility of Node-RED in representing and validating IoT scenarios and solutions, we present six different case studies: a smart supermarket scenario, a smart parking application, a smart home, a smart health context, a transport logistic application, and a smart agriculture scenario. Such case studies are analyzed with respect to the interacting entities, their relationships, and the IoT-based functionalities.

Note that many solutions, tailored to specific features of the realization of IoT-based systems (e.g., routing algorithms, security mechanisms, data exchange methods, etc.), have been proposed in literature [6], but often they only provide a partial representation of the whole architecture. Such a limit prevents to have a complete understanding of the implications and possible side effects of a designed IoT infrastructure; while the possibility of early (i.e., in a testing environment) analyzing and running the conceived IoT system could allow to save time before the final development, and to avoid mistakes. Moreover, it is worth to note that many approaches and architectures are evaluated in a single application domain with a single technology or protocol; while it is important to dispose of a tool which also facilitates the simulation of different scenarios, resembling different technologies/protocols, for example being able to accept different data formats and information coming from existing data-sets, possibly simulating a real time environment. Also, it is very important the possibility of separately representing the hardware and the software functionalities of an application, in order to let them being interchangeable (e.g., the need of testing an application with different hardware to choose the best one). These are the main motivations behind the proposed work. Finally, note that a tool which is generally recognized for simulating and validating IoT environments does not exist yet, and Node-RED could be a good candidate.

The remainder of this paper is organized as follows. Section II investigates the actual state of the art about the tools and methods used by the researches for validating their proposed solutions in the IoT fields, thus revealing our motivations; moreover, it points out the main application contexts considered until now in the literature. Section III presents the Node-RED's features and the proposed case studies, realized by means of the aforementioned tool. Section IV provides an analysis of the obtained results and a comparison with previous studies. Finally, Section V ends the paper, detailing a discussion about the conducted analysis, drawing some directions for future work.

II. Literature and Motivations

In the last decades, many solutions referred to various facets of the IoT applications and scenarios have been proposed. Some of the proposals regard architectural aspects, other ones refer to communication protocols, or data processing and storage, or security and privacy issues [6] [4]. In general, the works available in literature try to validate and test the performance of their proposed approaches by means of different programming languages and tools, and typical refer to particular case studies, which aim to provide a concrete application context (even if it is only simulated). Note that often researchers make use of data-sets in order to operate with real data (even if not in real time).

What naturally emerges is the need of a tool, able to allow the representation of the IoT infrastructures closer as much as possible to the future working system, in order to provide to designers and developers a complete view of the final architecture before its real deployment. Such a role has been played by wireless sensor networks' (WSN) simulators for many years [7], obviously in a context strictly related to WSN or WMSN (wireless multimedia sensor networks), but, with the advent of IoT, new systems must be adopted, due to the heterogeneity of the involved devices and to the different services provided. Hence, the main goal of the presented paper, as we will clarify in the following, is the adoption of a tool and a methodology able to represent and validate different IoT scenarios (i.e., in a general purpose fashion). In the following sections, we describe how IoT systems are actually evaluated in literature, in different application domains. Finally, we clarify the research question and the motivations behind the presented work.

A. Smart city case studies

First of all, the most investigated IoT context certainly is the smart city, as demonstrated by the recent interest of IBM, which proposes a roadmap of the challenges and applications of IoT in such a context [8]. The importance of smart cities is mainly due to the multiple facets of IoT application in the urban contexts, such as: parking monitoring, control of the traffic congestion, rescue routes, waste management, air pollution and noise control. For example, the work in [9] investigates all the aforementioned aspects within the "smart" city of Padova (Italy). The authors implemented a proof-of-concept deployment to test the effectiveness of their proposed architecture. Typical IoT protocols, such as CoAP and 6LoWPAN, have been used to cope with the resources' requirements of IoT constrained devices.

Interesting is the work proposed in [10] and [11], where a roadmap is proposed as a first step towards establishing a common and empirically valid theory for developing smart city strategies in a urban area. The target cities considered in that case are Barcelona and Amsterdam, and the envisioned strategies mainly regard organizational and financial aspects (i.e., not so focused to technological aspects), which are out of the scope of this paper.

Note that a relevant reference work for many research activities in smart cities is related to the SmartSantander platform [12], which has been conceived for representing an experimental test facility for the research and experimentation of architectures, key enabling technologies, services and applications for the IoT. The differences between the work proposed in this paper and the SmartSantander approach will be definitely clarified in Section IV.

An information framework has been proposed in [13], which focuses on noise mapping in the city of Melbourne (Australia). In such a work, an IoT architecture's proposal for noise mapping has been presented, along with a business model related to the target city of Melbourne, but no simulations or analysis on real data have been carried out. Even without any prototype development, the work in [14] proposes a Smart City IoT Platform based on a micro-service architectural style, thus opposing to the "traditional" SOA approach.

A cognitive management framework has been described in [15], where its cross-application nature has been demonstrated by instantiating a case study in one smart city scenario that horizontally connects several application domains, such as smart health, smart home, smart living, smart transport, and public safety. Such a context supports the interested stakeholders in understanding how the proposed framework may be exploited to protect and facilitate the daily life in the smart city, from an elderly citizen perspective. The analysis is carried out by means of a workflow diagram, and no practical implementation is provided.

Another software-defined IoT architecture for smart urban sensing has been detailed in [16]. Even here, a quantitative analysis has been provided, considering a target region of 5x6 rectangular urban areas, where three types of sensors' platforms are supposed to be deployed: fixed sensor platform, user smartphone based sensor platform, and mobile vehicle based sensor platform. Moreover, five urban sensing applications have been considered: street view, weather monitoring, noise monitoring, environmental monitoring, and dust monitoring.

In [17], the role of smart cities is highlighted in the context of MultiNational Enterprises (MNEs). The growing interest of MNEs in building alliances has, as the main goal, the starting of new projects with public and private city stakeholders aimed at exploring new technologies for smart cities' realization, in order to exploit new IoT-based devices and services to profit from them. What emerges from the analysis of 43 IoT smart city project alliances in Italy is that MNEs need to develop knowledge management capabilities, combined with ICT capabilities, in order to reach the desired objectives. The use of proper tools could help to fulfill such a gap, in an interdisciplinary way. Such an important aspect is also pointed out by the inquiry conducted in [18], where 21 in-depth semi-structured interviews with the smart city managers of 7 large multinational companies have been carried out. The outcomes point out the importance of the open innovation model for the realization of smart cities. The model should provide an efficient support for the involvement of private firms in public collaborations.

If the reader is interested in further studying the evolution of the research in the field of smart cities, the work in [19] presents a complete bibliometric analysis, putting in light the open challenges to be faced in order to reach a full urban innovation.

B. Smart buildings case studies

Other solutions strictly regard smart buildings, which mainly include smart homes and smart offices. The work in [20] uses real data-sets, collected from existing smart homes (reporting information such as energy consumption, lighting, and so on), for testing a middleware conceived to evaluate the security of the information transmitted within the underlying IoT infrastructure. The middleware runs on Raspberry Pi, and it is implemented in Node.js [21], while JSON formal language and MongoDB [22] database are used for data exchanging and storage, respectively.

Another test-bed consisting of a Raspberry Pi is detailed in [23]. Also here, the final goal is to evaluate a security protocol for enforcing the usage control policies. In both the cases (i.e., [20] and [23]) the test-beds consist of a limited number of devices, thus preventing to conduct relevant considerations about the scalability of the proposed approaches.

Instead, the authors of [24] present three different use cases to demonstrate the feasibility and efficiency of their architecture, by measuring home conditions, monitoring home appliances, and controlling home access. Such a solution integrates the IoT paradigm with web services and cloud computing. The following technologies has been adopted for the test-bench: Arduino platform for sensing and actuating functionalities; Zigbee for networking; cloud services; JSON data format for information exchange.

A prototype service for a smart office is provided in [25] to evaluate, from a functionalities' point of view, the proposed Integrated Semantics Service Platform (ISSP). The solution is based on an ontology and a model for semantic interpretations of user inputs through a proper web app. The whole architecture is based on Mobius [26], which is a oneM2M-compatible IoT service platform.

In [27], the work describes a practical realization of an IoT architecture, targeted to the University of Padova (Italy), which allows the interaction of WSN and actuators to standard networks as web services. It is an example of smart building, since the IoT network spans the floors and different areas within the Department of Information Engineering. Basic services, such as environmental monitoring and localization, regulated by roles and authorizations, are provided by means of the proposed approach.

A similar work has been carried out at the University of Bari (Italy), where existing hardware and software IoT solutions have been glued together to provide a reliable monitoring system, able to handle both scalar and media data belonging to either Internet Protocol version 4 (IPv4) and IPv6 realms [28]. More in detail, an IoT middleware, named NOS (NetwOrked Smart object) [29], able to manage heterogeneous data, have been integrated with TLSensing platform, able to efficiently acquire environmental information, and with an IP camera, in charge of acquiring images from the surrounding environment. An experimental test-bed has been been deployed in a university's laboratory, in order to continuously monitor environmental conditions and access control, also against malicious behaviors (e.g., intrusion detection).

Based on a coordinator-based ZigBee network, the smart home control system, presented in [30], has been written as a C# programme in charge of simulating the behavior of the users. A similar approach is that of [31], where ZigBee nodes are simulated by means of a well-known WSN simulator, named NS2. The main drawback, emerged from such solutions, is that an IoT system is too complex for being simulated by a WSN simulator or by a "simple" software.

Without providing any practical test or application, the design model proposed in [32] envisions the adoption of Node.js (i.e., as the application logic) [21] and MongoDB (i.e., as the storage) [22] for developing both the IoT clients and gateways in a smart building scenario.

Sequence diagrams are provided by the authors in order to explain the interactions taking place within the IoT architecture.

C. Transports and logistics case studies

Also transports and logistics fields play a fundamental role among the IoT applications. In [33], a case study is used to clarify the functionalities of a DNS architecture for the IoT. The behavior of a DNS infrastructure, which adopts a three-tier hierarchy of domain name servers and a three-level caching strategy, has been simulated. Preliminary results indicated that such an approach could be feasible for object tracking in the IoT domain, but wide simulations and tests are required.

A logistic cloud-based framework has been presented in [34]. It mainly makes use of RFID technology in order to monitor the goods and pallets in the analyzed warehouse. The main drawback of such a solution is that the experimental evaluation is limited to the simulations of RFID, without considering the whole conceived architecture, which also consists of private and public cloud, and other service provider modules (e.g., a security management and authorization module).

Even adopting RFID, the work in [35] details a web-based embedded system to control vehicles' access from a university campus. Unfortunately, the authors did not carried out a performance evaluation of the proposed application.

The same is for the authors of [36], who propose possible software and hardware technologies to implement their proposed smart public transport system, but no performance evaluation is fulfilled.

[37] limits to envision a future era of "Internet of Vehicles", where advances in the IoT principles' application will foster intelligent transport, and vehicles will be increasingly autonomous and able to take decisions without the human intervention. Hence, only design perspectives are highlighted, promoting the adoption of a vehicular cloud model, but no simulation campaign has been provided.

D. Smart health case studies

Very relevant is the development of smart health applications based on IoT technologies, in order to improve the services provided by healthcare structures to citizens. Starting from an electrocardiogram (ECG) feature extraction case study (useful for preventing cardiac diseases), the work in [38] couples the use of smart gateways with fog computing in order to improve the performance of a "traditional" health monitoring system. Such an approach mainly include the following functionalities: embedded data mining, distributed storage, notification service at the edge of network. Preliminary performance results has been obtained by means of a basic test-bed, making use of an existing data-set.

Similar concepts of "smart health" gateways and fog computing can also be found in [39] and [40] by the same authors, but a complete test-bed is not provided. In fact, a proof of concept implementation of the envisioned IoT-based remote health monitoring system is obtained by means of a demo of the "smart health" gateway. It provides the following local services: repository, compression, signal processing, data standardization, a web socket server, protocol translation and tunneling, firewall, and data mining and notification. The demo includes all the data flow processes from the bio-electrical signal acquisition at sensor nodes to the remote cloud-based healthcare center and web clients.

A wider smart health scenario is presented in [41]. Here, a policy enforcement framework involving doctors, patients and simple visitors is formalized by means of XML syntax, but no implementation or quantitative analysis is provided.

Another theoretical approach [42] consists of a simple and secure IoT architecture aiming at establishing a generic and ubiquitous Ambient Assisted Living (AAL) framework to be used by mobile health applications. The global solution presented here is based on RFID and Electronic Product Code (EPC) for uniquely identify the involved entities (e.g., medicines, patients, caregivers, hospitals, pharmacies, and so on).

Only supported by a trivial case study, the 4-layers model, proposed in [43], discusses the use of smartphone's sensors and body sensors to acquire, process and transmit patients' health data to centralized storage units in the cloud. Such a solution represents the most widely adopted approach today in the e-health field.

E. Smart manufacturing case studies

With the advent of the industry 4.0, many manufacturing companies are starting to adopt the new IoT technologies to regulate and improve their production flows. Such an approach is leading towards a real evolution in the field of industrial automation. As an example, [44] describes a set of proof-of-concept prototypes, which aim at gathering real-time data along the manufacturing processes. Hence, they enable a responsive production management and maintenance, including energy consumption and water usage monitoring at each stage of the production cycle. No performance evaluation is provided, but the proposed system is very promising and its scope is very important.

A system diagram design and hardware requirements are described in [45], to allow the realization of an IoT-based smart industrial automation and control system. It should be able to monitor the industrial data from anywhere in the world using Internet and the cloud. Moreover, it should also remotely control the actuators. The required hardware is very minimal in order to keep the cost low. However, no experiments have been conducted regarding such a proposed solution.

Smart manufacturing solutions for decision making are also important for SMEs (Small and Medium-sized Enterprises). This is the target of the work in [46], which presents the use of Auto-ID, like RFID or Barcode technology, to convert various manufacturing resources into smart objects, enabling the interactions among each other. Hence, minded production information and knowledge are acquired and elaborated to make advanced decisions in the production automation.

IoT and cloud manufacturing are integrated in [47] to develop a dynamic production logistics synchronization management system. A simple company is adopted as a case study to describe the system usage, which, however, does not include a demonstration of the running framework, showing its behavior and possibility to be adapted to other application contexts.

Instead, the work in [48] analyzes the virtualization of the food supply chains' concept. Also such a a feature appears to be fundamental, since increasingly stringent requirements are spreading in the food sector. Controls mainly regards perishable products, unpredictable supply variations and, in general, food safety and sustainability. Virtualization would enable supply chain actors to remotely and promptly monitor, control, plan and optimize business processes, through the Internet. A case study of a fish supply chain is proposed by the authors in order to better explain the functionalities behind their envisioned approach.

F. Research question and motivation

As emerged from the aforementioned analysis, many works and projects related to the IoT are tailored to particular case studies, in different application fields, ranging from smart cities, smart buildings, smart transports, smart logistics, and smart health. Some of them also provide prototypical implementation or use real data-sets to perform a preliminary evaluation of the proposed architecture and mechanisms. Most of them limits to provide a qualitative or quantitative investigation. Moreover, often it is not the whole system to be tested, but only a part of it, and such an aspect represents the main lack in the previous studies and solutions.

Such an issue arises because the adopted evaluation methods are not able to provide the required instruments to embrace all the aspects and life-cycles of the analyzed infrastructures. Note that it is due to the presence of multiple communication protocols, hardware and software technologies, which are difficult to put in act all together in a single test-bench. As a consequence, researches are led to conduct analysis only on a part of the envisioned architectures (usually the core of the application) and omit other interacting parties. However, the response times and behavior of such parts may have a great impact on the performance of the application's core. Hence, it is really important to try to simulate the behavior of the full components of the desired IoT system, in order to better understand the performance and have a complete view on the final solution.

In our opinion, the use of Node-RED tool would fill such a gap and would help researchers in performing a complete preliminary simulation of the designed IoT systems. Being composed of black-boxes, it stays at a higher level than a real implementation, which is usually very expensive. Node-RED makes available, at the same time, many flows acting, for example, as cloud nodes, or Raspberry Pis, Arduino platforms, and so on. In fact, it enables the real connection of hardware devices and APIs. Note that, if the reader is interested, for a comparison among Node-RED and other existing web-based platforms, along with of detailed description of their features and functionalities, we refer to [49].

In order to demonstrate Node-RED potentialities in simulating relevant and complex IoT case studies, we propose, in the followings, different scenarios, realized by means of the Node-RED tool. Furthermore, in Section IV we propose an analysis and comparison of the achieved results with respect to the related works presented in such a section.

III. Methodology

Node-RED [5] is a web-based and event-driven tool, which is implemented in JavaScript, using the Node.js framework. It provides a browser-based flow editor, where data flow programs, consisting of nodes, can be connected by wires, on the basis of the required behavior of the designed application. Node-RED makes available a large set of nodes and flows, mainly developed by the open source community. New flows and sub-flows may be defined by the developers. Moreover, Node-RED can also be deployed on a physical smart device; in fact, the lightweight nature of Node.js and the simplicity of the Node-RED execution engine allows Node-RED flows to be executed with good performance on devices, such as the Raspberry Pi or Arduino.

For such reasons, we chose Node-RED for developing different IoT case studies, related to the followings scenarios: smart retail, smart parking, smart home, smart health, smart transport logistic, and smart agriculture.

Such technologies has been mainly adopted for the developing phase: (i) MongoDB [22], as database engine, instead of a relational database, due to its greater efficiency in responding to a large number of queries in a short time; (ii) Java or Node.js [21] as programming languages, due to their widely adoption in current real implementations; (iii) MQTT [50], as lightweight publish&subscribe method, for information sharing and system's notifications. It is worth to remark that all the information transmitted within the described systems are encrypted, by means of proper functions, made available by Node-RED *CryptoJS* component. It supports many encryption/decryption functions, such as AES, DES, RC4, digest such as MD5, SHA-1, SHA-3, and hash-based message authentication codes (HMAC).

A. *Smart retail case study*

The smart retail case study is focused on the monitoring and management of a supermarket. The shelves (five in our simulation - $S_1 ... S_5$) are equipped with readers (i.e., RFID tags) to detect the presence of products (i.e., identifying when a specific product is picked up or stored away). In addition, the desk fridge (one of the five shelves in our simulation) has a temperature sensor with the relative thermostat to continuously monitor the temperature. Also the carts (20 in our simulation) are equipped with sensors to detect the presence of the selected products. Each cart provides an interface that shows the list of the purchased products along with the total expense. Payments via NFC are also allowed. Furthermore, the carts can be localized in order to show to the supermarket's administrators the people's distribution within the supermarket itself. Figure 1 shows the related dashboard, which is realized by means of the flow reported in Figure 2.

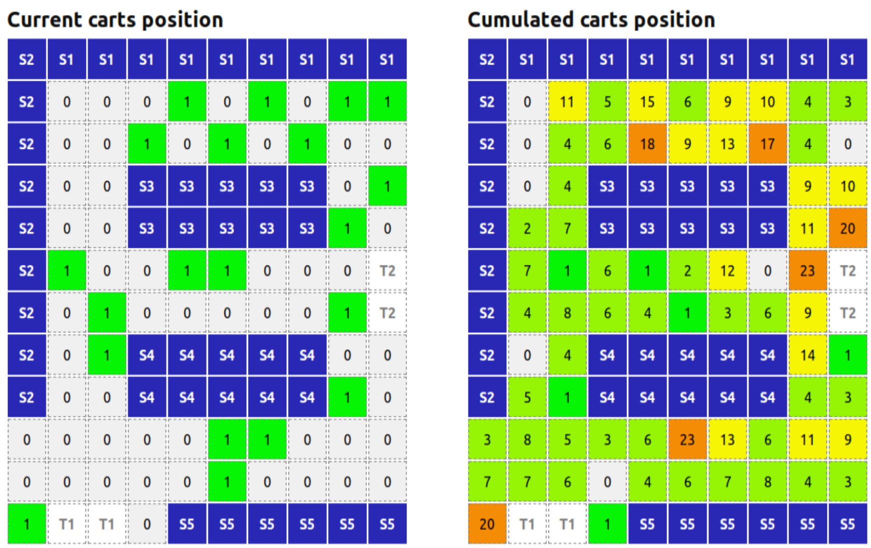

Fig. 1: Smart retail - carts' position

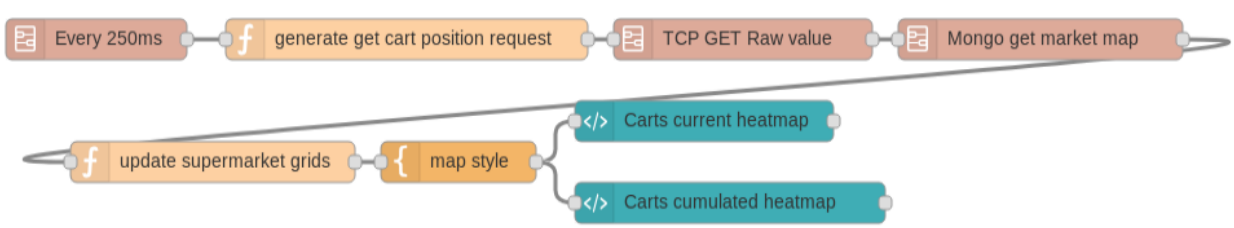

Fig. 2: Smart retail - flow for creating and updating the carts' heatmap

Each cart has four sensors: (i) one for detecting the added products; (ii) one for detecting the removed products; (iii) a localization sensor; (iv) an NFC reader for managing payments. At each entrance/exit to/from the supermarket there are security tags' totems (two in our simulation - $T_1 ... T_2$), in charge of detecting any unpaid products. Security tags' totems are also equipped with alarm signals. Instead, employees are provided with a proper interface, which can be used to lower or increase the products' price. Figure 3 summarizes the scheme of the supermarket.

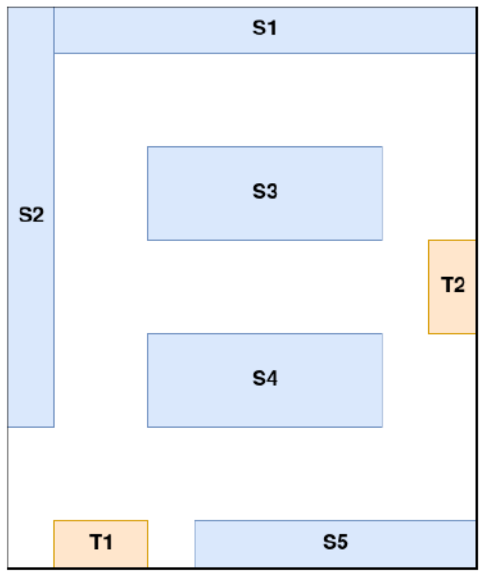

Fig. 3: Smart retail - scheme of the supermarket

All the mentioned sensors are simulated by TCP servers, which act as shelves, carts, and totems, and are implemented in Node.js. MQTT protocol is used to share information related to alerts (e.g., theft attempts), status of the shelves, or changing in the products' price or availability. A real data-set has been used for simulation. In fact, the data related to the supermarket named "Il Gigante" have been obtained from Web Scraper[1]. Figure 4 reports the real-time situation of the supermarket (i.e., the shelves' status, the activities regarding the shopping carts - position, add/remove of a product, payment -, the status of the totems).

[1]https://chrome.google.com/webstore/detail/web-scraper/jnhgnonknehpejjnehehllkliplmbmhn

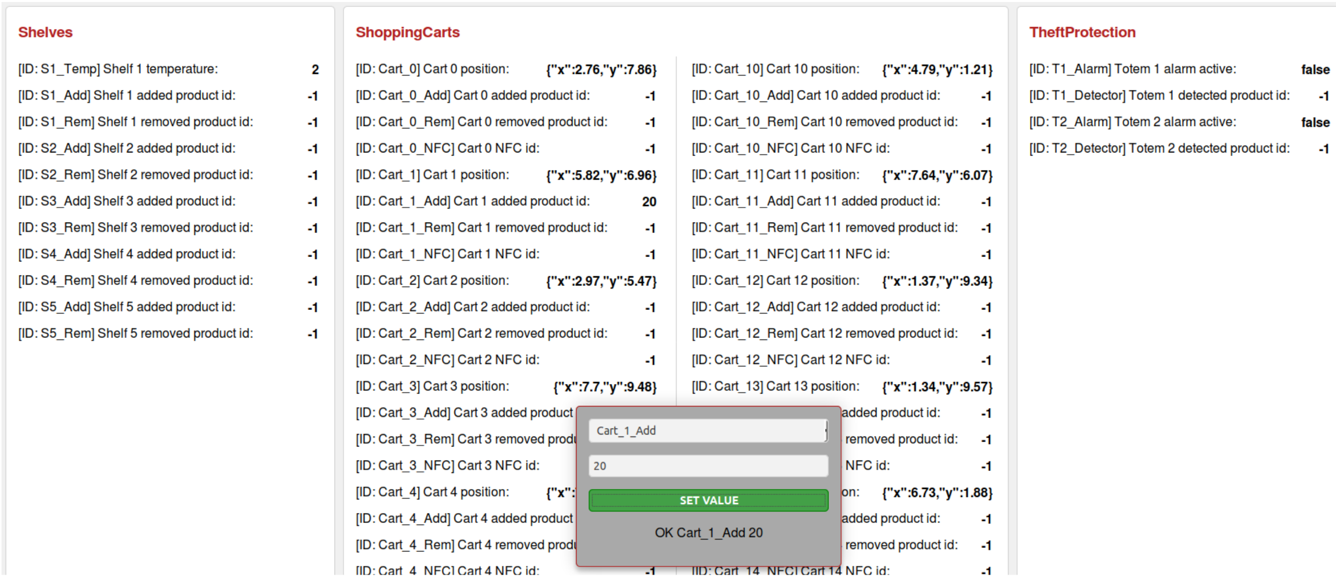

Fig. 4: Smart retail - dashboard of the real-time situation of the supermarket

B. Smart parking case study

The smart parking application, presented herein, deals with the control of the signals of a two-story car park. The IoT system controls two kinds of traffic lights: (i) the entrance light indicating if the parking is full or not; (ii) the traffic light at each floor that indicates the current floor to enter.

Some rules applied in the described scenario are the followings:

- The entrance light is green when there is at least one free place in one of the two floors;
- Cars in the first floor are accepted until such a floor has no more than 70% of busy places; once exceeded such a threshold, cars are sent at the second floor;
- If also the second floor has more than 70% of busy places, then the cars are sent to the first floor until it is complete;
- If the first floor is full and the second floor has free places, the cars are sent to the second floor.

The application provides five sensors to signal the input/output of vehicles: (i) one sensor marks the entry of a car into the parking; (ii) two sensors, one per floor, mark the entry of a car in a certain floor; (iii) two sensors, one per floor, mark the exit of a car from a certain floor. Moreover, registered users can reserve a place for their car at a certain date and time; a reservation is considered as a busy place, without specifying the floor.

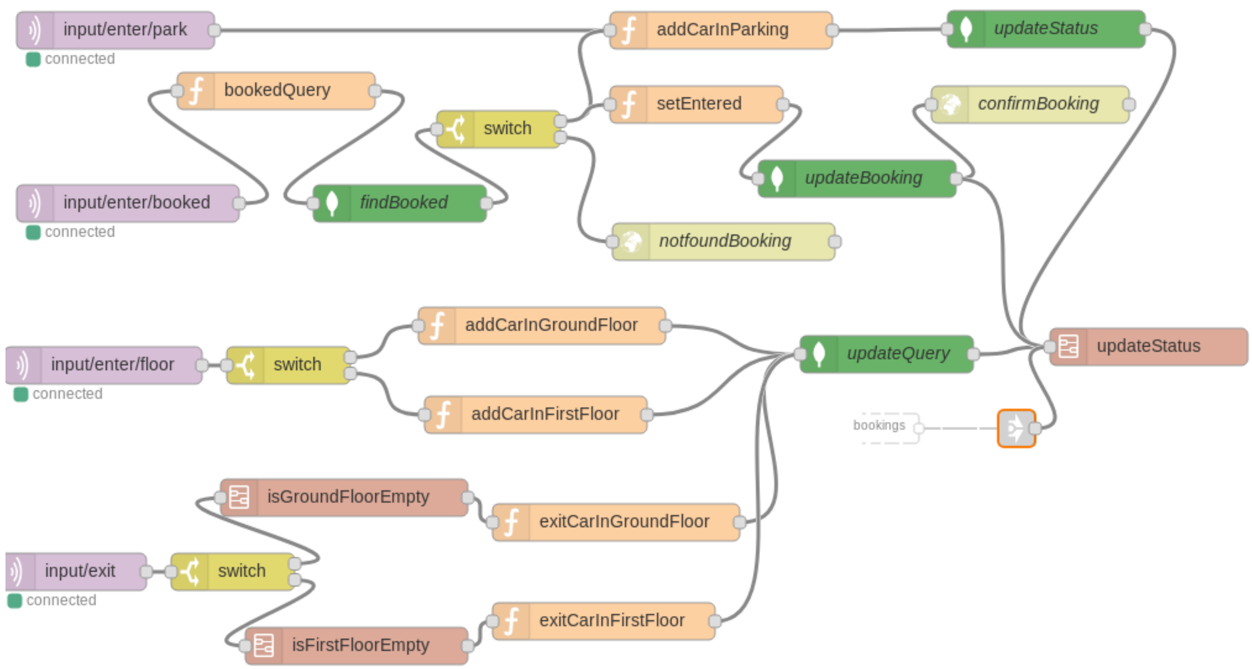

Fig. 5: Smart parking - flow

The logic, just described, is managed by a Node-RED application (see the flow in Figure 5); an HTML interface, provided by a flow of Node-RED itself, allows the users' registration and reservations' requests; data are stored in a MongoDB database (three collections are disposed for storing the users' information - username and password -, the reservations, and the status of the parking). The sensors are implemented on a Raspberry Pi 3 that allows the sending of analogical signals to set traffic lights, and to receive analogical inputs to signal the entry/exit of the cars. The application is written in Python and allows: (i) to activate the switching on of analogical outputs in response to MQTT notifications; (ii) to publish entry/exit messages on the MQTT channel in response to analogical inputs. The application does not handle the entry of users who have a reservation, since it requires the use of an optical or NFC reader to recognize the users themselves. The Raspberry Pi directly communicates with the Node-RED application. The test-bed is illustrated in Figure 6. As an alternative to the Raspberry Pi, the simulation of the sensors can be also implemented with Node.js and Express.js.

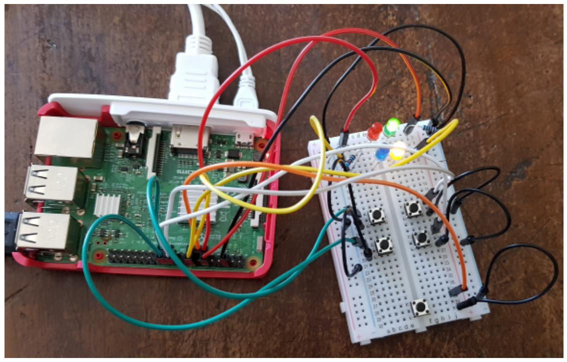

Fig. 6: Smart parking - test-bed

Note that users' registrations and reservations are managed via HTTP protocol; while other signals regarding the status of the smart parking are managed via MQTT protocol. The dashboard in Figure 7 shows the current status of the car park.

C. Smart home case study

The aim of such a case study is to put in act heterogeneous sensors (both digital and analogical) for the automated management of different facilities of a smart home, keeping, at the same time, the user informed about the occurring events. The smart home model considered herein consists of four rooms (i.e., kitchen, living room, bathroom and bedroom) and the garden. The arrangement of the sensors for each room is described in the following:

- Kitchen: temperature, humidity, brightness, magnetic
- Living room: temperature, humidity, brightness, magnetic, NFC
- Bathroom: temperature, humidity, brightness, magnetic
- Bedroom: temperature, humidity, brightness, magnetic
- Garden: brightness, infrared, irrigation, rain detector, NFC

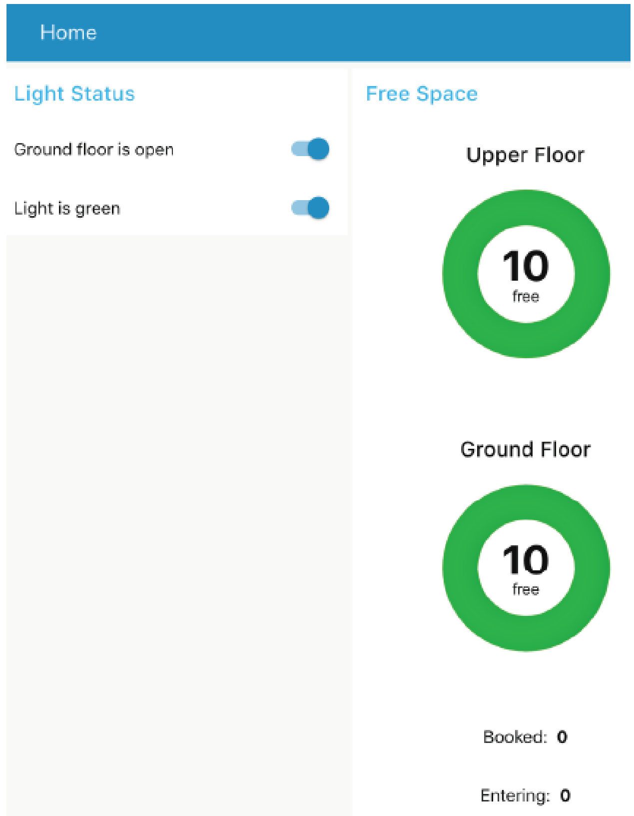

Fig. 7: Smart parking - dashboard showing the status of the car park

The functions of each kind of sensor are:

- Temperature sensors: located in all the rooms, they monitor the temperature, in order to manage the conditioning system
- Humidity sensors: located in all the rooms, they monitor the humidity present in the air, in order to manage the conditioning system
- Brightness sensors: located in all the rooms, they detect the degree of brightness
- Magnetic sensors: installed on the windows, they detect their open/close status
- NFC: installed on the gate and on the entrance door, it allows their opening via smartphone
- Infrared: placed in the garden, it detects the presence of people inside
- Irrigation device: a controller useful to manage the switching on and off of the irrigation

system

- Rain detector: placed in the garden, it detects the rainy weather, so as to avoid the start of the irrigation system

The adopted technologies are: (i) a Java application for simulating the behavior of the aforementioned digital and analogical devices; (ii) the TCP protocol to acquire data from such simulated sensors and transmit them to the Node-RED application; (iii) a MongoDB database to manage the history of acquired data; (iv) MQTT to notify the changes happening within the IoT system; (v) the Node-RED flows, which manage the conditioning system within the rooms, the irrigation system in the garden, the intruders' detection both in the garden (by means of infrared) and in the rooms (e.g., brightness due to a violated window, or the unusual open of a window). Figure 8 exemplifies the irrigation system's management. More in detail, before starting the irrigation system, some checks are performed: (i) if it will rain during the day (the weather forecast is obtained by means of the library *OpenWeatherMap*, made available by Node-RED); (ii) the temperature and humidity values. If the irrigation system must be started, then an MQTT message is sent to the broker to notify the interested subscriber (in this case, the irrigation system itself). Hence, it runs for certain period of time and, then, stops, notifying the IoT system application (see Figure 9). It is worth to remark that each task is logged into the database, and information are transmitted in a ciphered way. Moreover, the smart home system is programmed to keep the light on in the garden during the night or if someone is in the garden.

Furthermore, an alert on the smartphone of the user is sent in case of intrusion detection, as performed by the flow in Figure 10.

The user can also selectively decide to activate only the external alarm, only the internal alarm or both of them (the dedicated web page is shown in Figure 11).

All the alarms are notified under pre-defined topics on MQTT. In case of gate opening or of the entrance door by means of the NFC, the respective alarms are disabled to allow the user to entry in the home, as sketched in the flow of Figure 12.

The developed IoT system also allows the user to manually add new sensors to the application or modify the existing ones, by associating relevant information, such as: the location (a room or the garden), the type (digital or analogical), the status (enabled or disabled). In this way, the

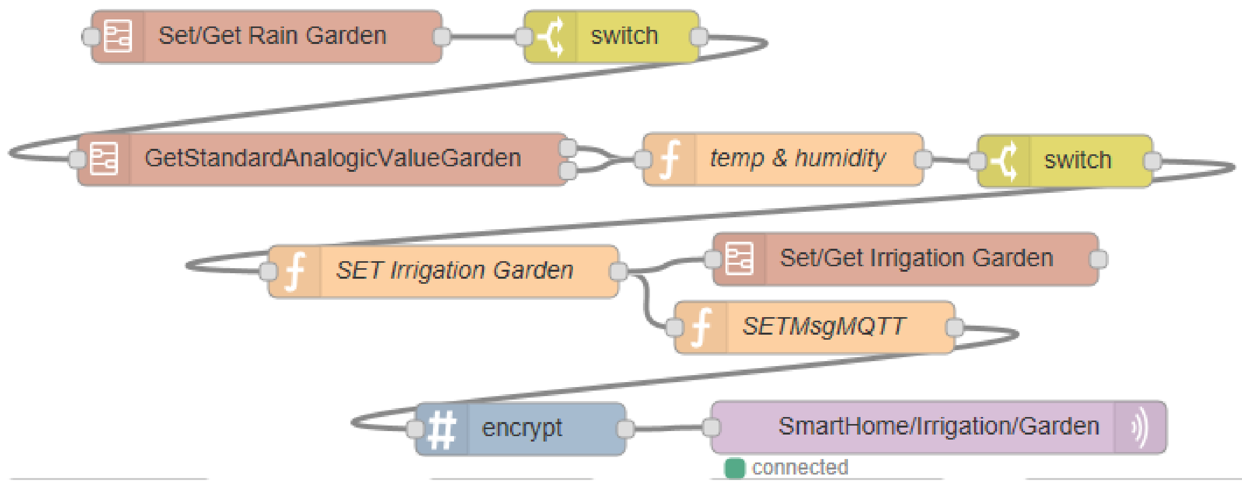

Fig. 8: Smart home - start of the irrigation system management's flow

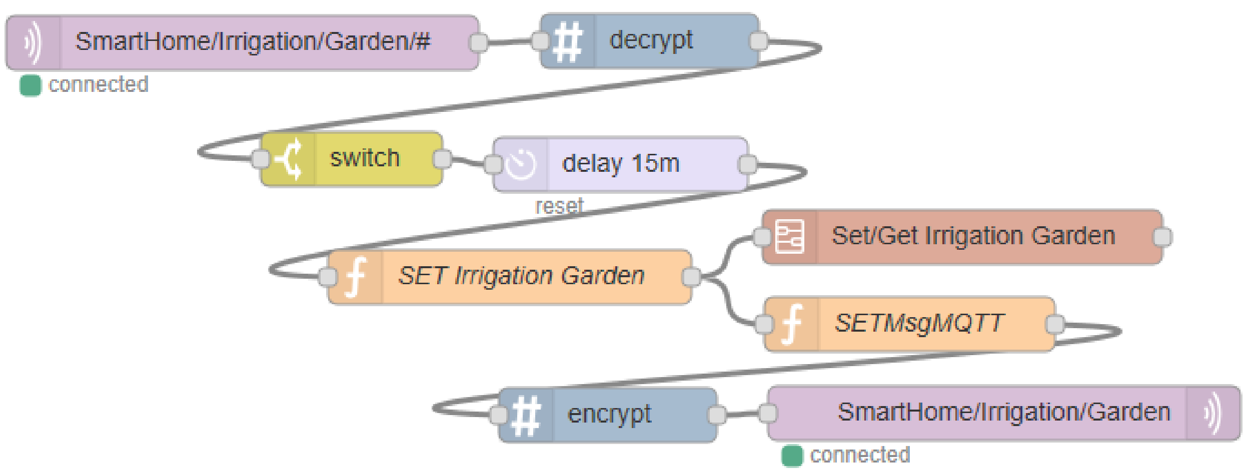

Fig. 9: Smart home - stop of the irrigation system management's flow

presented case study becomes very dynamic and close to a real application, where users may need to modify the sensors' configuration of their home.

Figure 13 shows a sample dashboard for the kitchen, which is very similar to the other rooms and to the garden. In fact, they use gauges and simple charts to show the current values and the trend of the analyzed parameters.

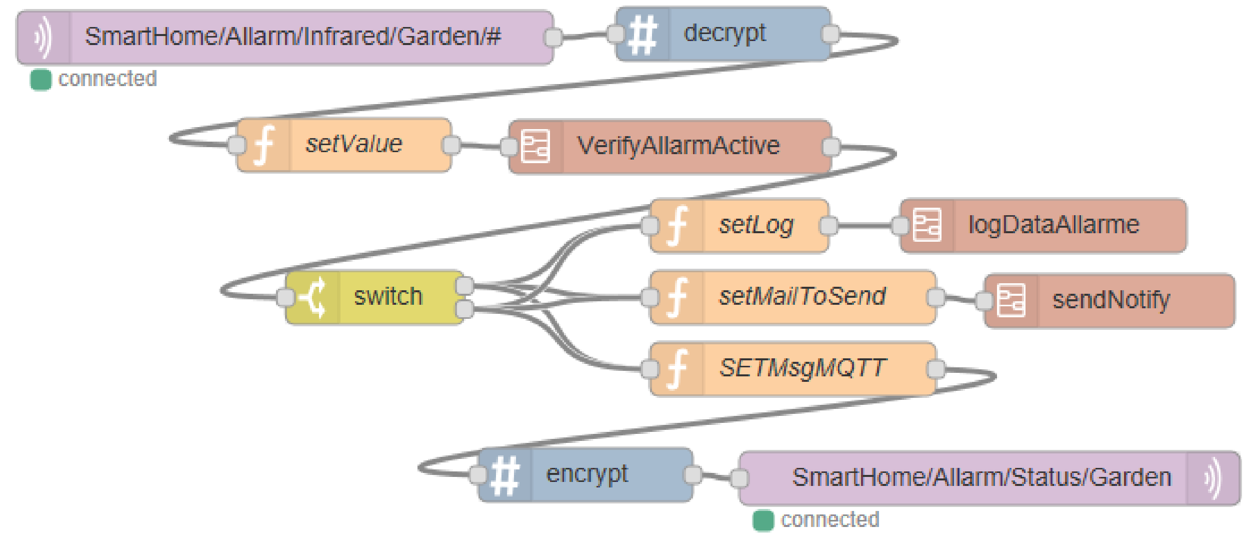

Fig. 10: Smart home - flow related to alarm activation in case of intrusion detection

Fig. 11: Smart home - form for the setup of the web page for alarms' selection

D. Smart health case study

The case study related to the smart health domain is referred to the monitoring of diabetic patients. The main goal is to provide a system able to dynamically calculate the amount of insulin to be administered to patients having diabetes of type 1 and 2 [51]. Many parameters are involved in such a scenario, as detailed herein.

The existing kinds of insulin varies, depending on the action's speed and duration; hence, we have insulin: (i) very fast; (ii) fast; (iii) intermediate; (iv) long-lasting; (v) premixed. The needle, used for injecting the insulin, must be changed every time; moreover, the possible injection sites are the followings: (i) the abdomen, for the injection of fast, very fast and premixed insulin; (ii) the arms, for the injection of intermediate and long-lasting insulin; (iii) the thighs for the injection of intermediate and long-lasting insulin; (iv) the buttocks, for the injection of intermediate and long-lasting insulin. The injection technique depends on the length of the needle: (i) up to 5

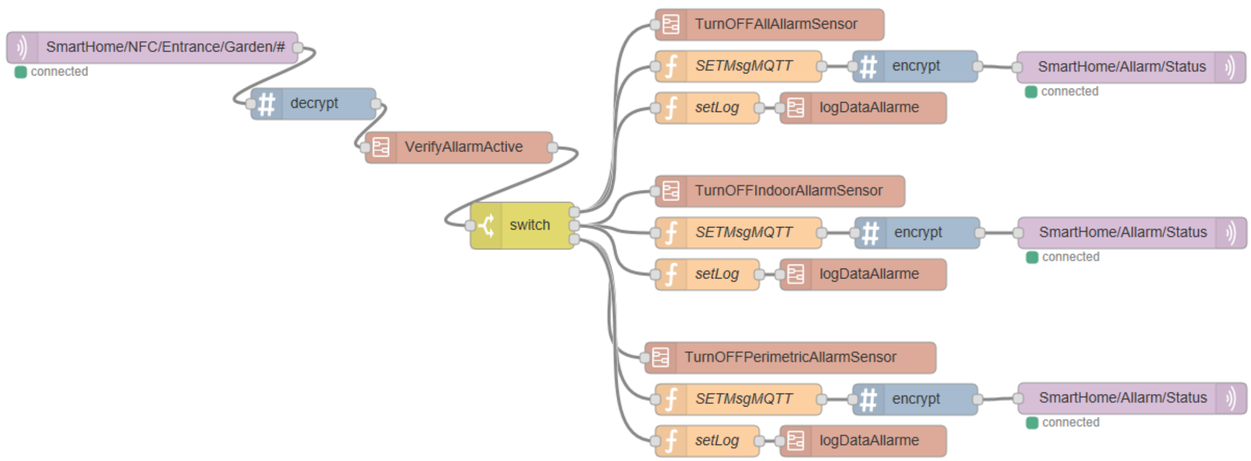

Fig. 12: Smart home - flow of the deactivation of the alarms in case of user's entry by means of NFC

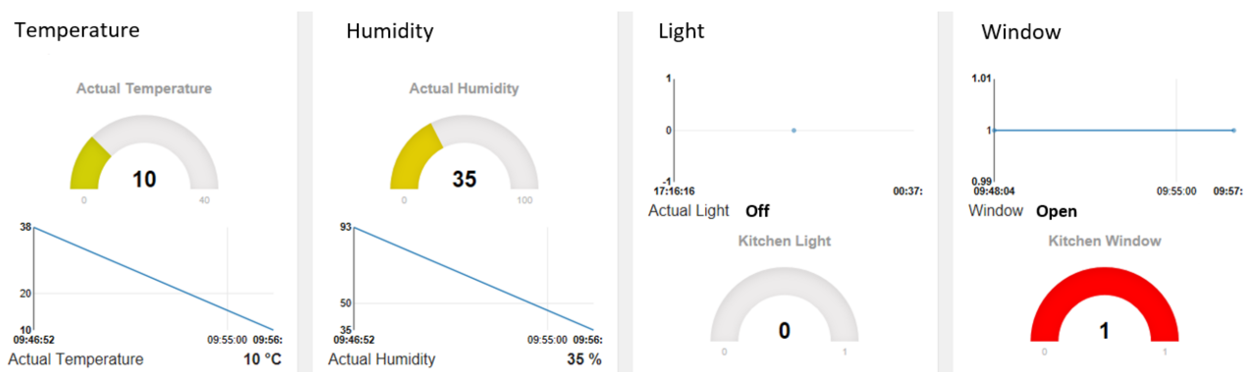

Fig. 13: Smart home - sample dashboard of the kitchen

mm, so the needle should be positioned 90 degrees from the injection site; (ii) over 5 mm, so the needle should be positioned at 45 degrees.

The calculation of the insulin dosage also takes into account the grams of carbohydrates, which will be consumed by the patient in the next meal; for the calculation of each kind of carbohydrate, the proportion is as follows:

$$x : gc = cyin100gr : 100gr \quad (1)$$

where gc represents the grams to be consumed, and cy represents the grams of carbohydrates.

We only considers, as examples, the grams of carbohydrates in 100 gr of the following foods: (i) flour: 10 gr; (ii) sugar: 15 gr; (iii) Parmesan cheese: 10 gr. Finally, the symptoms of hypoglycemia are: (i) sweating; (ii) tachycardia feeling; (iii) blurred view; (iv) tremors. Other factors that can influence the insulin's dosage are the presence of allergies, physical activity, or events such as trauma, surgical operations, concomitant diseases.

For the calculation of the insulin's dosage, we take into account the blood's sugar, a correction factor (for simplicity, equals to 40 mg/dl), the ratio I/CHO (insulin/carbohydrates) equals to 1/16, and the case of intense physical activity; the following outcomes may verify:

- If the blood's sugar is 151-180 mg/dl, the calculation of the insulin's dose is recommended for that meal;
- If the blood's sugar is 121-150 mg/dl, the insulin dose should be reduced of $1U$ (U = unit);
- If the blood's sugar is 90-120 mg/dl, the insulin dose should be reduced of $2U$;
- If the blood's sugar is 181-210 mg/dl, the insulin dose should be increased of $1U$, keeping the same diet;
- If the blood's sugar 211-240 mg/dl, the insulin dose should be increased of $2U$, keeping the same diet.

Taking in mind such features, the following rules have been listed, which take into account the actions undertaken by the doctor, by the patient, and those automated by the IoT system:

- The doctor, through a form (see Figure 14), enter the patient's data (type of diabetes, type of insulin, tolerance to drugs, events - such as those indicated above: trauma, surgical operations, concomitant diseases -, allergy to insulin, type of the needle);
- The doctor, through a form, inserts the date of the next visit;
- The patient, through a form, inserts useful information for calculating the final insulin's dosage, such as the intention of performing intense physical activity, or the carbohydrates consumed in the next meal;

- The patient, through a form, declares to have changed the needle;
- The patient, through a form, indicates if he/she presents the symptoms of hypoglycemia, mentioned above (see Figure 15);
- Each time interval, the patient's blood sugar is measured, by the system, as well as the insulin's dosage;
- If the patient presents one or more symptoms of hypoglycemia, together with a low blood's sugar level (e.g., less than 80 mg/dl), a family member will receive an alert via e-mail by the IoT system;
- If the blood's sugar is 61-80 mg/dl, this value will be published, by the IoT system, with an AES encryption, under the topic *device/alarm/blood sugar/low*;
- If a new data is published under the topic *device/alarm/blood sugar/low*, the patient will receive an e-mail with the following text: ”Take 1 cup of milk with 2 rusks or 1 cup of milk with 1 cracker”;
- If the blood sugar is less than 60 mg/dl, this value will be published, by the IoT system, with an AES encryption, under the topic *device/alarm/blood sugar/very low*;
- If a new data is published under the topic *device/alarm/blood sugar/very low*, the patient will receive an e-mail with the following text: ”Take 3 candies or 3 sachets of sugar or 1 can of coca-cola or 1 can of fruit juice”;
- If the patient presents a blood's sugar greater than 250 mg/dl, this value will be published, by the IoT system, with an AES encryption, under the topic *device/alarm/blood sugar/high*;
- If a new data is published under the topic *device/alarm/blood sugar/high*, the patient will receive an e-mail with the following text: ”Take 1/2 liter of water” and the alarm signal will be activated;
- If the blood's glucose level is right (i.e., between 81 ad 249 mg / dl), this value will be published, by the IoT system, with an AES encryption, under the topic *device/log/glycemia*;

- If a new data is published under the topic *device/log/glycemia*, the alarm will be turned off;
- Reports of the data published under the topics *device/alarm/blood sugar/+* and *device/log/glycemia* can be downloaded as separate files.

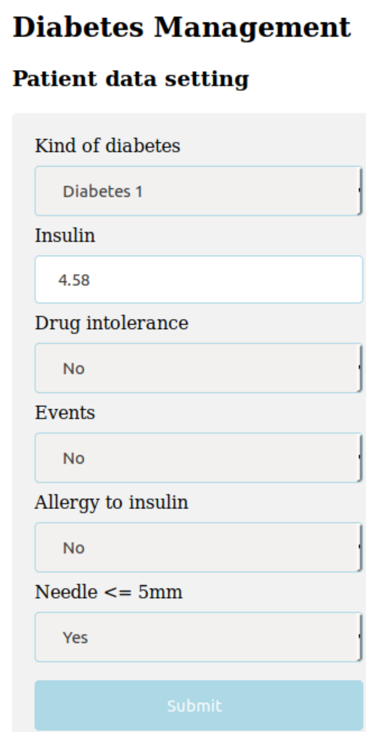

Fig. 14: Smart health - doctor's form to enter the patient's data

The flows, related to the management of the glycemia's alarms, are shown in Figures 16 and 17.

The IoT system continuously provides and updates a dashboard that displays the blood's glucose trend (see Figure 18), and a dashboard to visualize the value of the last administered insulin. The useful information, such as the patient's blood's sugar, the values of administered insulin, the dates of the visits, and so on, are stored in a MongoDB collection. MQTT is used as publish&subscribe method to manage both alarms and logs; while Node-RED implements the rules listed above in a set of flows. Hence, such flows include the logic of the whole application and set up the proper actions in response to the situation in real time. Instead, the device, which

Fig. 15: Smart health - patients' form to enter the symptoms of hypoglycemia

should be owned by the patient in order to perform the real monitoring and actuating activities of the diabetes, is simulated by means of a Java program. Note that the Java program separately runs and communicates with the Node-RED application via HTTP protocol. In this way, it is possible to test all the system's functionalities, and, then, in a second time, attach a real hardware, instead of the Java program. Nothing must be changed in the logic of the application, but only the connected I/O device.

E. Smart transport logistic case study

Referring to the smart transport logistic field, the goal of this case study is to create an IoT application for guaranteeing the control of products' delivery, through the management of smart vehicles and tracking of the products themselves.

Products will be tracked by means of RFID tags; while smart vehicles will be equipped with a set of sensors that will automatically perform different tasks depending on information contained in the delivery package tag. The fundamental system's components are: (i) the smart vehicles; (ii) the RFID tags; (iii) the RFID scanners; (iv) the warehouse. Communications among such

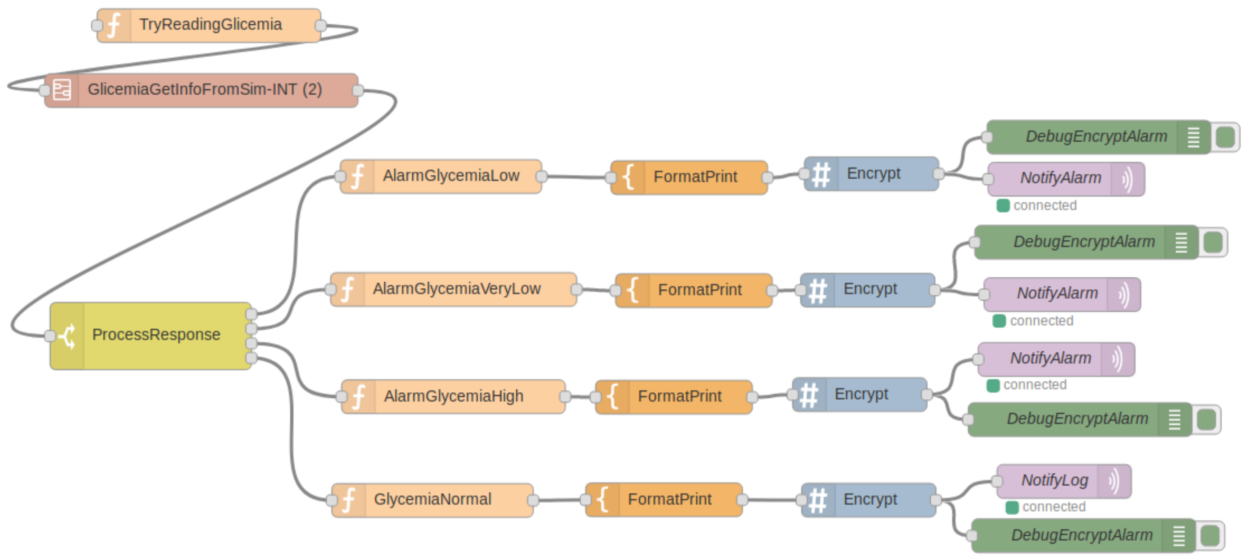

Fig. 16: Smart health - flow related to the logging and alarms to dedicated topics in response to glycemia values

components will only take place via MQTT protocol, exploiting a well-defined topic hierarchy. MongoDB will have the active role of identifying the status of the products (e.g., to sell, in delivering, and so on) and the the status of the smart vehicles (e.g., available, in delivering). A Java program will simulate the communications, via proper TCP connections, among sensors and RFID tags.

The smart vehicles will be equipped with the following sensors and devices: GPS, vibration sensor (inspired by Arduino SW-420), humidity sensor, temperature sensor, dehumidifier, motor fridge. Such kinds of delivery can be carried out: (i) *STANDARD*, which only provides the execution of a geographical coordinates' check; (ii) *FRIGO*, which provides a check on the coordinates, the temperature and the humidity of the smart vehicle; (iii) *FRAGILE*, which provides a check on the coordinates and on vibration. The anomalies are reported with alarms via MQTT notifications to the warehouse. The topics' hierarchy is summarized in Figure 19, revealing that not only errors are managed via MQTT, but also messages useful for the management of the logistics.

Four Node-RED flows are needed to manage the just presented scenario:

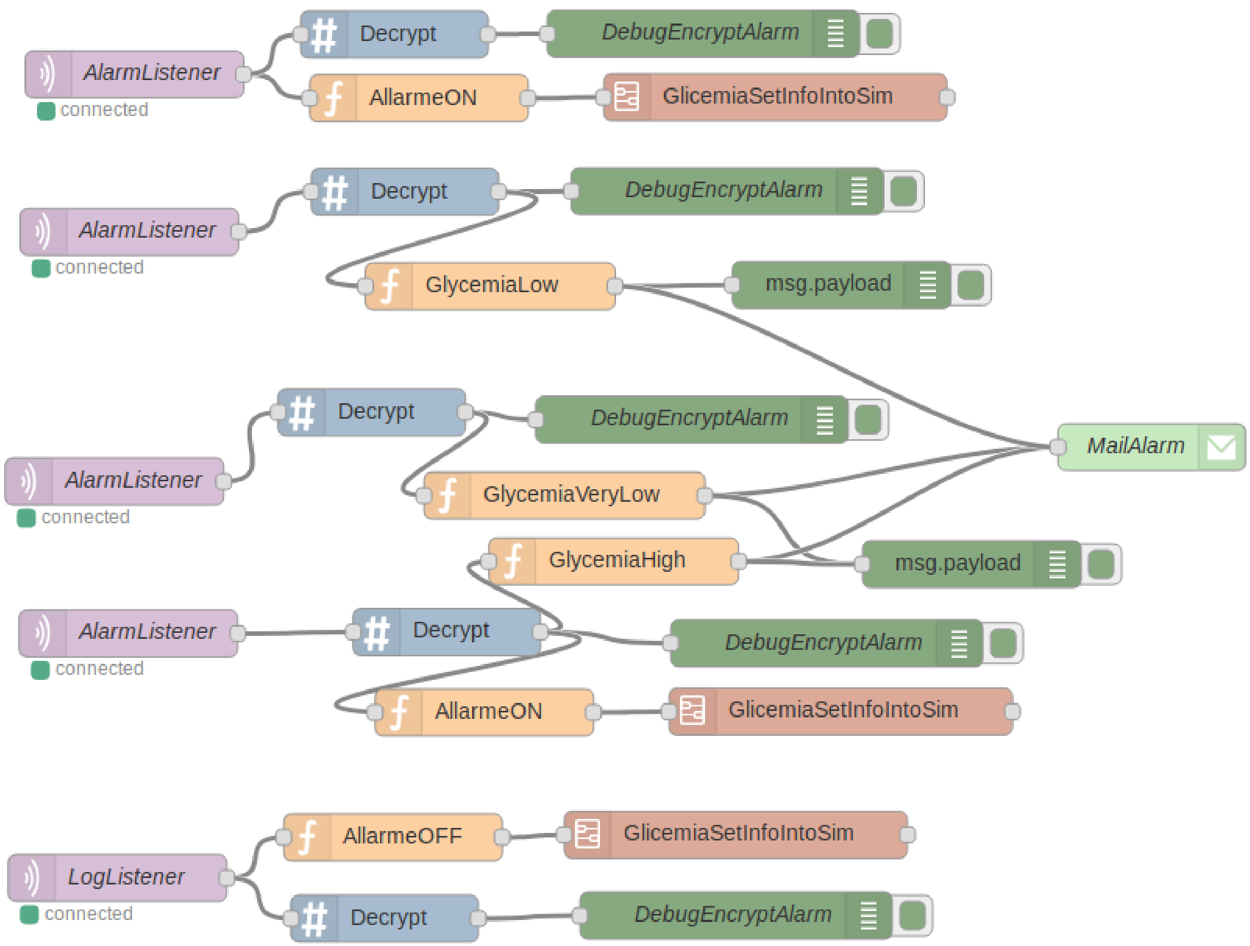

Fig. 17: Smart health - flow related to listening on the dedicated topics of glycemia situation

- The flow related to the central system that manages the users' orders; such orders are then sent to the warehouse which is notified by the fact that one or more products have been sold, and, then, are ready to be delivered (see Figure 20);
- The flow related to the management of the warehouse; it executes the proper checks to verify the presence of some products to be delivered;
- The flows associated to the RFID scanner, which are in charge of keeping updated the information related to the status of the products in the warehouse;
- The flow associated to the behavior of the smart vehicles.

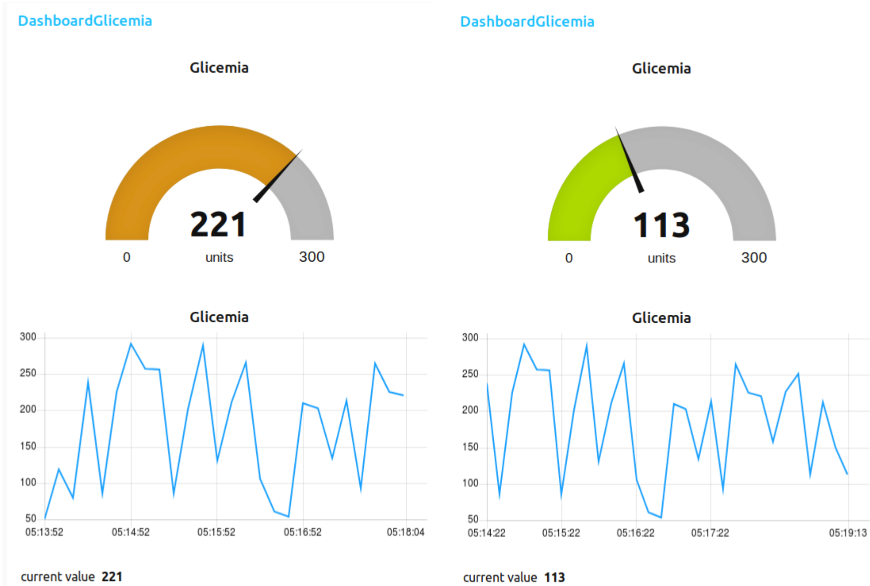

Fig. 18: Smart health - dashboard showing information about the patients' glycemia

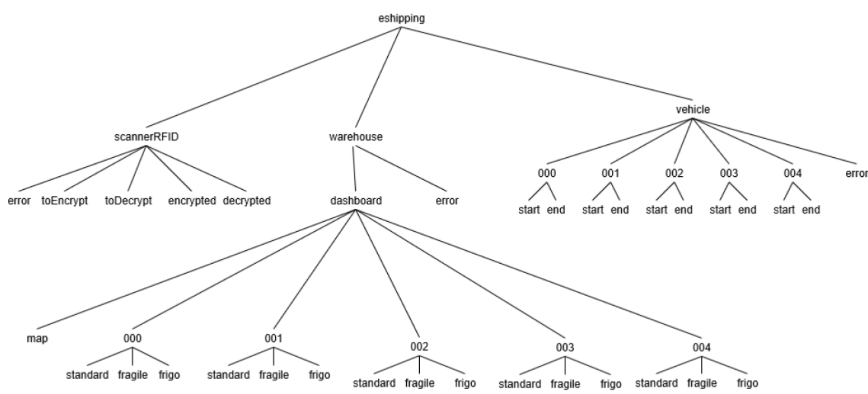

Fig. 19: Smart transport - topics' hierarchy

Proper dashboards have been defined in order to dynamically simulate: (i) a purchase notify; (ii) a delivery attempt; (iii) the visualization of the vehicles' location. Another dashboard provides real-time information about the status of the vehicles (coordinates, temperature, humidity, vibration), as sketched in Figure 21. Figure 22 shows statistics about the status of the products and the alarms. A visualization map, finally, shows the position of the vehicles in real-time.

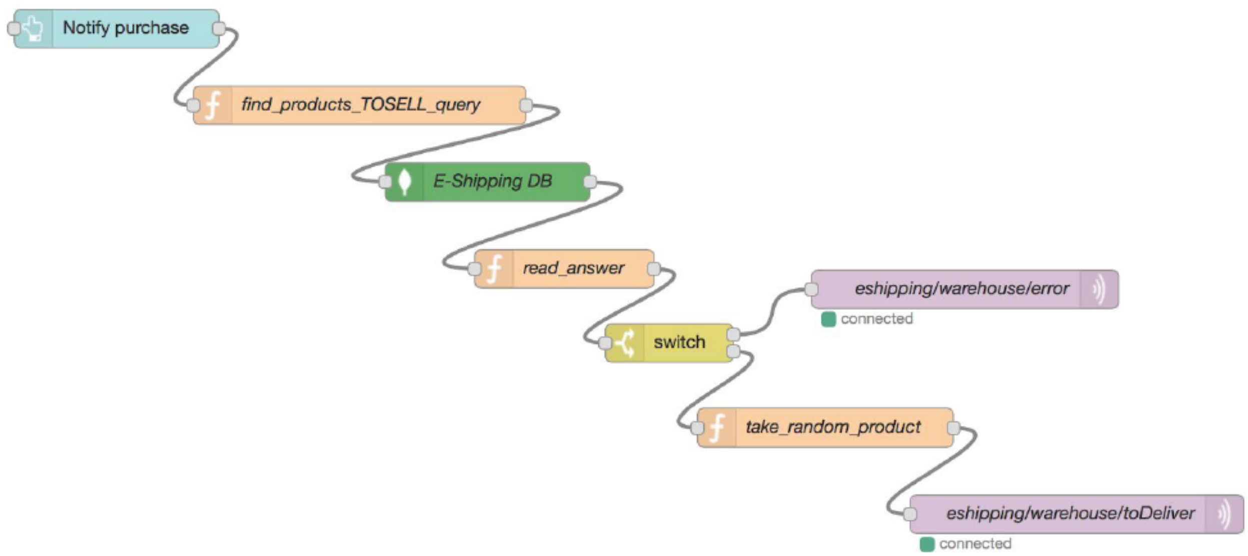

Fig. 20: Smart transport - flow for notifying a deliver to the warehouse

F. Smart agriculture case study

With regards to smart agriculture, the case study consists of an application in charge of automatically managing a greenhouse. At regular interval times (every three hours in our simulation), the application obtains the weather forecast (by means of the library *OpenWeatherMap*, made available by Node-RED), collects data on the soil conditions from the sensors placed within the greenhouse and queries a MongoDB database containing useful information on the kinds of plant situated inside the greenhouse. All these data are analyzed by the IoT system, which decides whether to activate one or more greenhouse actuators. Note that the weather information are periodically published on a dedicated topic */weather* via MQTT, in order to notify the interested flows for actuating proper countermeasures depending on the current environmental conditions.

The greenhouse is divided into four areas. In each area there can only be one type of plant. The information about the four types of plants inside the greenhouse are stored in a MongoDB database, within a single collection. The followings information is stored for each type of plant: (i) the area of the greenhouse where the plant is located; (ii) the ideal temperature for that kind of plant; (iii) how often it must be watered (i.e., in terms of days); (iv) the level of moisture to be reached by the soil in order to water the plant; (v) how many days have passed since the last

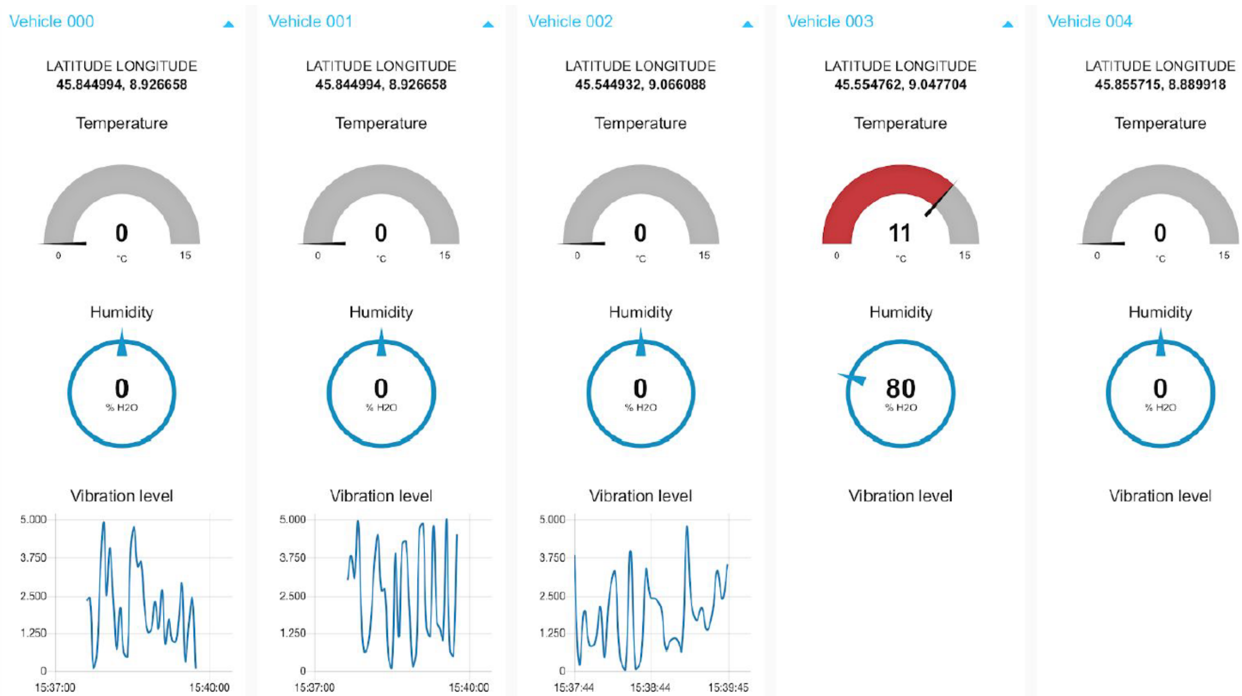

Fig. 21: Smart transport - dashboard showing information about the vehicles

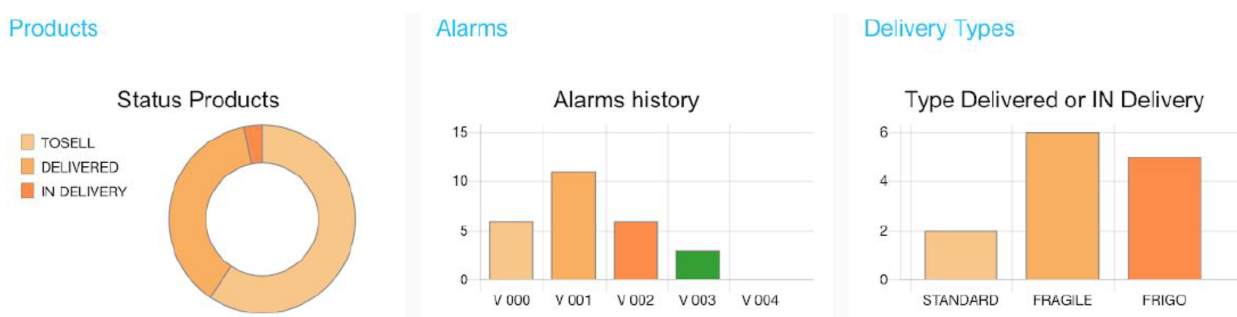

Fig. 22: Smart transport - dashboard showing information about products and alarms

time the plant was watered; (vi) the ideal quantity of light for exposing the plant; (vii) the kinds of insect that are harmful for the plant. Three types of insects are considered. Each of them can damage one or more types of plants. In each area there can only appear one type of insect.

A Java application simulates the greenhouse. In each of the four areas, the following sensors are placed: temperature detector, brightness detector, humidity detector, insect detector and insecticide detector. Instead, the actuators placed in each area are the followings: irrigation system, protection dome, insecticide dispenser, thermostat and artificial light.

There is also a central node, which acts as a controller, able to communicate with all the greenhouse's sensors and actuators. Each sensor, actuator and controller is simulated by a separate Java application class. The Node-RED application communicates with the class that simulates the controller to receive data from the sensors and change the status of the actuators, using the TCP communication protocol. The pattern *Skeleton* has been applied.

To test the correct behavior of the IoT system, some environmental changes are simulated:

- The soil temperature is periodically modified so that it is always closer to the ambient temperature. Note that the value of the ambient temperature is provided by the Node-RED application;
- The soil moisture level is periodically reduced depending on the soil temperature. If the Node-RED application indicates that it is raining, then the humidity level is increased;
- Periodically, the level of natural brightness is determined using the percentage of cloud cover. This information is always provided by the Node-RED application. At night-time the brightness level is reduced;
- Periodically, if insects were previously detected, the number of insects in the soil is randomly increased. If before there were no insects, randomly a certain kind of insect may appear;
- If the number of insects exceeds the value 100, it is reset to zero. In this way, in every area sooner or later will appear insects that can damage the plants;
- Periodically, the amount of insecticide present in the soil is decreased. If a new type of insecticide is added, the old type is completely eliminated.

In such a way, the Node-RED flows can provide the logic for the whole greenhouse's management, being able, by means of the defined rules and the parameters set into the database, to regulate the use of: (i) the irrigation system in case of lack of water in certain areas; (ii) the opening or closing of the protection dome, depending on the current weather conditions (e.g., hailstorm in progress or good weather); (iii) the insecticide dispenser, depending on the presence of insects or insecticide itself; (iv) the thermostat, depending on the requirements of the plants in each area; (v) the artificial light, even depending on the requirements of the plants in each area. The status of the greenhouse can be monitored in real-time by means of the dashboard shown in Figure 23.

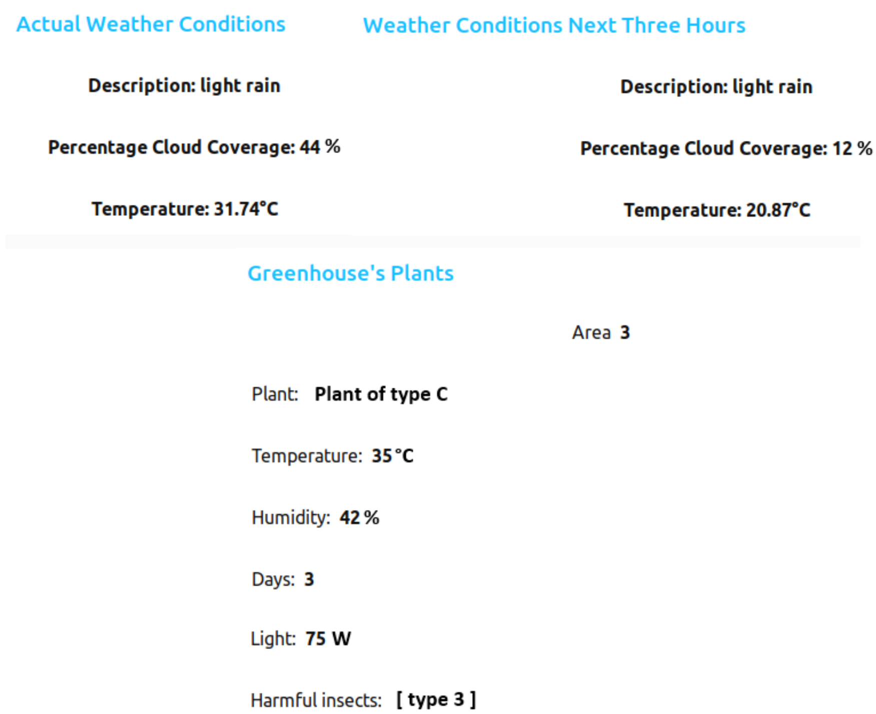

Fig. 23: Smart transport - dashboard showing the status of the greenhouse

IV. Analysis and comparison

By means of the case studies presented in Section III, we pointed out different features of the IoT application domains. Note that the presented scenarios require the adoption of different technologies as well as different data flows and life cycles are put in act. The setup and description of the involved entities, protocols, and functionalities for each of the investigated case studies compose the design phase. However, another fundamental step is required before real deployment, which is the testing in a simulation environment.

To this end, we proposed the use of Node-RED tool. With Node-RED, it is possible to provide a complete view of the IoT application case studies, allowing to analyze and simulate them, considering all the involved entities and messages' passing. In the analysis conducted in relation to the proposed case studies, note that the hardware devices (i.e., digital and analogical sensors) are simulated by means of Java or Node.js programs; while the software and logic of the application are completely contained in the Node-RED flows, which allow the designers of the IoT system to monitor and, practically, see the data and events' flow of the whole application. Different technologies, such as Java language, XML, JSON, Node.js, MQTT, TCP, and MongoDB, are grouped together without worrying about interoperability issues, thanks to the libraries offered by Node-RED. In that sense, we state that such an approach represents a viable solution for performing preliminary tests on heterogeneous IoT scenarios. In fact, it should also reveal errors or possible malfunctions in the application's logic or in the devices' interactions, which can be corrected before deploying the whole system. Hence, the support provided by a tool such as Node-RED could help in wasting time in deployment's attempts, without already having a clear and complete view of the entire IoT system.

With respect to the state of the art, as presented in Section II, and to the proposed case studies (Section III), we derive the following considerations:

- The case study related to the smart parking represents a facet of a smart city. With respect to the solutions described in Section II-A, the approach, presented here with Node-RED, certainly allows an easier configuration and test of the investigated context; for example, the framework, presented and developed in [9] in the city of Padova (Italy), is currently targeted to such an environment, and a huge effort would be required if the same solution were evaluated and further deployed in another city. Also, the SmartSantander platform [12] is targeted to real deployments in existing cities; until now, such cities are Belgrade, Guildford, Lubeck and Santander, where 20,000 sensors will be deployed. Hence, the scope of our evaluation and of that proposed by the SmartSantander platform is totally different: the former represents a higher level methodology to evaluate new solutions and data flows' management before real deployment; while the latter aims at practically deploying the conceived solution in real environments. Note that the other approaches, mentioned in Section II-A, do not provide any simulation or test-bed to verify the envisioned functionalities; but it is fundamental to propose new architectures and methodologies inspired to IoT principles, as we do by means of Node-RED.

- With respect to the works presented in Section II-B, some of them tried to test the proposed smart home systems by means of real test-beds [20] [23] [24]. The main drawback is that such test-beds, due to their limited dimension, do not allow to perform a preliminary evaluation about the scalability of the architecture's entities and interactions. Instead, Node-RED tool allows to simulate the presence of a lot of devices and their interactions. Other solutions, as for the smart cities, are targeted to specific environments, such as the University of Padova (Italy) [27] and the University of Bari (Italy) [28]. Hence, they are so specifically, while it is important to define general-purpose IoT solutions. Some approaches are strictly related to a platform (e.g., [25] is based upon the M2M Mobius) or to a particular technology ([30] and [31] are based upon ZigBee). Finally, with respect to such solutions, Node-RED tool serves as a mean for evaluating/testing new designed methods, to be used/applied inside the smart home/building, without directly exploiting the real deployment. In fact, a preliminary analysis in a controlled simulation environment is important in avoiding to waste time in inaccurate deployments, as previously hinted.
- Maybe, the smart health is the most investigated field in coupling medical facilities management with the IoT. Real implementations of remote patients' monitoring systems and e-health architectures have already been envisioned [42] by the interested parties. The works [38] [40], presented in Section II-D, only include limited test-beds. In such a direction, the use of a tool as Node-RED would help in designing and testing new solutions, targeted to specific medical structures, before the deployment. As an example, due to the strictly security and privacy requirements of medical information, it is fundamental to perform a preliminary evaluation of security and privacy protocols, as the ones proposed in [41], which is currently only presented in an analytical way, without a practical vision of its effectiveness.
- With regards to the transports and logistics fields, as revealed by Sections II-C and II-E, literature is not mature enough, since it is mainly limited to theoretical works without providing an in-depth analysis of performance and usability, such as [36] [37] [45] [47] [48]; furthermore, the current deployments are mainly based on the use of RFID technologies [34] [35] [46]. Hence, the use of a proper tool enables to simulate

both simple and complex environments, by means of the same application logic. Such an approach would be a great opportunity for achieving advances in small and medium enterprises (SMEs) in the industry 4.0 era, which could potentially involve heterogeneous cooperating technologies and protocols.

V. Discussion and conclusions

The paper investigated different application case studies conceived according to the IoT paradigm's principles. Such scenarios have been designed and developed by means of the Node-RED flow-based programming tool, due to its important features, such as: the data flow programs, the browser-based editor, the lightweight nature of the platform, the possibility of deployment on real devices. Hence, we demonstrated the feasibility and usability of Node-RED in representing the behavior of typical IoT environments, considering all the entities and interactions involved. In fact, at the end of the investigated IoT case studies' overview, it is worth to remark that Node-RED is a tool which allows to perform an in-depth analysis of the behavior of different IoT scenarios, before real deployment. The following pros of the proposed approach can be pointed out:

- The application's logic is encapsulated in the Node-RED flows and it is kept separate from the underlying simulated hardware.
- Real data-sets can be integrated and used as sources for testing the analyzed scenarios, as well as real-time data incoming from connected physical devices or obtained from web services.
- The interactions between the application's front-end and back-end can be effectively simulated, enabling an easy evaluation on how the two parts interact with each others.
- Different programming languages can co-exist as well as different communications protocols, whose functioning can be tested in a single simulation environment inside Node-RED tool, without requiring to enable some interoperations among different tools or platforms.
- Different database engines can be used.

- The connection of the application's logic with real devices (e.g., Raspberry Pi, Arduino) is possible.
- Node-RED tool natively provides options for generating well-structured dashboards, charts, gauges, and so on, which help in defining the output of the analyzed application.

Summarizing, with respect to the open issues pointed out in Sections I and II-F, the conducted study revealed that Node-RED enables to set up a complete overview of the desired scenarios; in the real world, such a feature certainly helps in taking into account all the possible variants and failures. From a theoretical viewpoint, Node-RED tool acts as a support in the design phase of complex and heterogeneous IoT scenarios (i.e., not limited to single technologies, protocols, or scenarios); useful demos can be obtained by using the flows made available by the tool itself. Such demos can then be used, from a practical viewpoint, as test environments, in order to check if the designed system behaves as expected. Such a step is fundamental before carrying out a real deployment of the conceived solutions. Also, the tool allows the separation of hardware and software functionalities, which is important in case it emerges the need of changing some system's features. Maybe, in the next future, Node-RED could acquire a prominent place as a universally accepted general purpose simulation environment for the IoT.

Naturally, some limitations also emerge in the use such a tool, which mainly regards its usability, in presence of large systems; in fact, the organization of the various sub-systems in different projects is not so easy as well as to allow them to work together; moreover, a versioning mechanism is not provided, in order to trace the changes operated during the time.

As a future research direction, we aim to extend the presented work by defining re-usable modules and components, which will be made available to the interested research community, in order to speed up the design and development of other innovative scenarios, always related to the IoT paradigm. In our opinion, the availability of ready-to-use modules should encourage further investigations in such an area. Another important aspect regards the possibility to carry on a performance analysis concerning execution times, delays, storage occupancy, CPU load, and so on; the obtained results should also depend of the kind of the hardware which will be deployed after the design phase.

Hence, note that IoT applications suffer of four main issues: scalability, interoperability, security and privacy. Concerning the last two ones, it is worth to remark that complex security and privacy mechanisms and policy management systems could be integrated, where required, in novel or in existing IoT scenarios. Clearly, proper tests must be conduced, in order to verify the feasibility of the proposed approaches. Also in this sense, Node-RED could help researches in have a preliminary vision over the future real systems to be realized.

REFERENCES

[1] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad hoc networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[2] N. Jazdi, "Cyber physical systems in the context of industry 4.0," in *Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–4.

[3] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[5] "Node-red, "flow-based programming for the internet of things"," https://nodered.org/.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[7] I. Minakov, R. Passerone, A. Rizzardi, and S. Sicari, "A comparative study of recent wireless sensor network simulators," *ACM Transactions on Sensor Networks (TOSN)*, vol. 12, no. 3, p. 20, 2016.

[8] V. Scuotto, A. Ferraris, and S. Bresciani, "Internet of things: Applications and challenges in smart cities: a case study of ibm smart city projects," *Business Process Management Journal*, vol. 22, no. 2, pp. 357–367, 2016.

[9] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.

[10] R. Bolici and L. Mora, "The development process of smart city strategies: the case of barcelona," 2016.

[11] L. Mora and R. Bolici, "How to become a smart city: Learning from amsterdam," in *International conference on Smart and Sustainable Planning for Cities and Regions*. Springer, 2015, pp. 251–266.

[12] L. Sanchez, J. A. Galache, V. Gutierrez, J. M. Hernandez, J. Bernat, A. Gluhak, and T. Garcia, "Smartsantander: The meeting point between future internet research and experimentation and the smart cities," in *Future Network & Mobile Summit (FutureNetw), 2011*. IEEE, 2011, pp. 1–8.

[13] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things journal*, vol. 1, no. 2, pp. 112–121, 2014.

[14] A. Krylovskiy, M. Jahn, and E. Patti, "Designing a smart city internet of things platform with microservice architecture," in *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*. IEEE, 2015, pp. 25–30.

[15] P. Vlacheas, R. Giaffreda, V. Stavroulaki, D. Kelaidonis, V. Foteinos, G. Poulios, P. Demestichas, A. Somov, A. R. Biswas, and K. Moessner, "Enabling smart cities through a cognitive management framework for the internet of things," *IEEE communications magazine*, vol. 51, no. 6, pp. 102–111, 2013.

[16] J. Liu, Y. Li, M. Chen, W. Dong, and D. Jin, "Software-defined internet of things for smart urban sensing," *IEEE communications magazine*, vol. 53, no. 9, pp. 55–63, 2015.

[17] S. Bresciani, A. Ferraris, and M. Del Giudice, "The management of organizational ambidexterity through alliances in a new context of analysis: Internet of things (iot) smart city projects," *Technological Forecasting and Social Change*, vol. 136, pp. 331–338, 2018.

[18] A. Ferraris, G. Santoro, and A. Papa, "The cities of the future: hybrid alliances for open innovation projects," *Futures*, vol. 103, pp. 51–60, 2018.

[19] L. Mora, R. Bolici, and M. Deakin, "The first two decades of smart-city research: A bibliometric analysis," *Journal of Urban Technology*, vol. 24, no. 1, pp. 3–27, 2017.

[20] S. Sicari, A. Rizzardi, D. Miorandi, and A. Coen-Porisini, "Securing the smart home: A real case study," *Internet Technology Letters*, vol. 1, no. 3, p. e22, 2018.

[21] "Node.JS," http://nodejs.org/.

[22] "MongoDB," http://www.mongodb.org/.

[23] A. La Marra, F. Martinelli, P. Mori, and A. Saracino, "Implementing usage control in internet of things: a smart home use case," in *Trustcom/BigDataSE/ICESS, 2017 IEEE*. IEEE, 2017, pp. 1056–1063.

[24] M. Soliman, T. Abiodun, T. Hamouda, J. Zhou, and C.-H. Lung, "Smart home: Integrating internet of things with web services and cloud computing," in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2013, pp. 317–320.

[25] M. Ryu, J. Kim, and J. Yun, "Integrated semantics service platform for the internet of things: A case study of a smart office," *Sensors*, vol. 15, no. 1, pp. 2137–2160, 2015.

[26] "Mobius onem2m, "onem2m-compatible iot service platform"," http://wiki.onem2m.org/index.php?title=Open˙Source.

[27] A. P. Castellani, N. Bui, P. Casari, M. Rossi, Z. Shelby, and M. Zorzi, "Architecture and protocols for the internet of things: A case study," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*. IEEE, 2010, pp. 678–683.

[28] D. Costantino, G. Malagnini, F. Carrera, A. Rizzardi, P. Boccadoro, S. Sicari, and L. A. Grieco, "Solving interoperability within the smart building: A real test-bed," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[29] S.Sicari, A. Rizzardi, D. Miorandi, C. Cappiello, and A. Coen-Porisini, "A secure and quality-aware prototypical architecture for the Internet of Things," *Information Systems*, vol. 58, pp. 43–55, 2016.

[30] M. Khan, B. N. Silva, and K. Han, "Internet of things based energy aware smart home control system," *IEEE Access*, vol. 4, pp. 7556–7566, 2016.

[31] P. Xiang, "Design of smart home system based on the technology of internet of things," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 14, pp. 2236–2240, 2012.

[32] Y. Jiang, X. Liu, and S. Lian, "Design and implementation of smart-home monitoring system with the internet of things technology," in *Wireless Communications, Networking and Applications*. Springer, 2016, pp. 473–484.

[33] B. Karakostas, "A dns architecture for the internet of things: A case study in transport logistics," *Procedia Computer Science*, vol. 19, pp. 594–601, 2013.

[34] S.-L. Chen, Y.-Y. Chen, and C. Hsu, "A new approach to integrate internet-of-things and software-as-a-service model for logistic systems: A case study," *Sensors*, vol. 14, no. 4, pp. 6144–6164, 2014.

[35] R. M. Cardoso, N. Mastelari, and M. F. Bassora, "Internet of things architecture in the context of intelligent transportation systems - a case study towards a web-based application deployment," in *22nd International Congress of Mechanical Engineering (COBEM 2013)*, 2013, pp. 7751–7760.

[36] S. Apsara, G. A. Rashmi, K. V. Mohankumari, L. Anitha, and B. Jyothi, "Smart public transport system using internet-of-things," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 3, no. 2, 2017.

[37] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 241–246.

[38] T. N. Gia, M. Jiang, A.-M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare internet of things: A case study on ecg feature extraction," in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 356–363.

[39] A.-M. Rahmani, N. K. Thanigaivelan, T. N. Gia, J. Granados, B. Negash, P. Liljeberg, and H. Tenhunen, "Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems," in *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*. IEEE, 2015, pp. 826–834.

[40] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[41] S. Sicari, A. Rizzardi, L. Grieco, G. Piro, and A. Coen-Porisini, "A policy enforcement framework for internet of things applications in the smart health," *Smart Health*, vol. 3, pp. 39–74, 2017.

[42] A. Santos, J. Macedo, A. Costa, and M. J. Nicolau, "Internet of things and smart objects for m-health monitoring and control," *Procedia Technology*, vol. 16, pp. 1351–1360, 2014.

[43] K. Ullah, M. A. Shah, and S. Zhang, "Effective ways to use internet of things in the field of medical and smart health care," in *Intelligent Systems Engineering (ICISE), 2016 International Conference on*. IEEE, 2016, pp. 372–379.

[44] P. Brizzi, D. Conzon, H. Khaleel, R. Tomasi, C. Pastrone, A. Spirito, M. Knechtel, F. Pramudianto, and P. Cultrona, "Bringing the internet of things along the manufacturing line: A case study in controlling industrial robot and monitoring energy consumption remotely," in *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on*. IEEE, 2013, pp. 1–8.

[45] P. Pariyumbud, R. Samkaria, N. Karnatak, R. Singh, A. Gehlot, and S. Choudhary, "Smart factory automation using internet of things to ensure quality manufacturing," *Journal of Embedded System & Applications*, vol. 5, no. 3, pp. 1–6, 2018.

[46] R. Zhong, "Internet of things (iot) enabled smart manufacturing for smes," in *Manufacturing and Design for the Future 2017: A National Conference for Innovation in Manufacturing and Design*, 2017.

[47] S. Lei, T. Qu, Z. Wang, D. Nie, H. Luo, and G. Huang, "A smart implementation framework of physical internet with the integration of internet of things and cloud manufacturing," in *2nd International Physical Internet Conference*, 2015.

[48] C. N. Verdouw, J. Wolfert, A. Beulens, and A. Rialland, “Virtualization of food supply chains with the internet of things,” *Journal of Food Engineering*, vol. 176, pp. 128–136, 2016.

[49] M. Blackstock and R. Lea, “Toward a distributed data flow platform for the web of things (distributed node-red),” in *Proceedings of the 5th International Workshop on Web of Things*. ACM, 2014, pp. 34–39.

[50] “IBM and eurotech, ”mqtt v3.1 protocol specification”,” http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html.

[51] P. C. Davidson and H. R. Hebblewhite, “Determining insulin dosing schedules and carbohydrate-to-insulin ratios in diabetic patients,” Jan. 2018, uS Patent 9,872,890.