# Annexes

Collection of the author’s remaining publications at the time of this thesis:

**Federated Learning on 5G Edge for Industrial Internet of Things**

IEEE Network, 2024 https://doi.org/10.1109/mnet.2024.3469988

**Virtual Reality and Internet of Things Based Digital Twin for Smart City Cross-Domain Interoperability**

Applied Sciences, 2024 https://doi.org/10.3390/app14072747

**Monitoring Framework for the Performance Evaluation of an IoT Platform with Elasticsearch and Apache Kafka**

Information Systems Frontier, 2023 https://doi.org/10.1007/s10796-023-10409-2

**Alexa-Based Voice Assistant for Smart Home Applications**

IEEE Potentials, 2021 https://doi.org/10.1109/mpot.2020.3002526

**Digital Twins for Street Lighting: Challenges for a Virtual Reality solution based on Internet-of-Things Devices and Photometry Rendering**

LS18, 2023 https://doi.org/10.1109/ls1858153.2023.10170533

**Towards a Digital Twin for Smart Street Lighting Systems Using a Virtual Reality interface**

EEDAL/LS17, 2021 https://ieeexplore.ieee.org/document/9932212

**Management and Monitoring IoT Networks through an Elastic Stack-based Platform**

FiCloud, 2021 https://doi.org/10.1109/ficloud49777.2021.00034

**A Novel, Self-Powered, Non-Intrusive, Sigfox-Enabled Smart Meter for Challenging Scenarios**

IE, 2020 https://doi.org/10.1109/ie49459.2020.9154930

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3469988

# Federated Learning on 5G Edge for Industrial Internet of Things

Xiaoli Liu, Xiang Su, *Member, IEEE,* Guillermo del Campo, *Member, IEEE,* Jacky Cao, Boyu Fan, Edgar Saavedra, Asunción Santamaría, Juha Röning, Pan Hui, *Fellow, IEEE,* Sasu Tarkoma, *Senior Member, IEEE*

***Abstract*—Industry 4.0, leveraging the Internet of Things (IoT) and Artificial Intelligence (AI), is a key enabler for many automated processes in modernized industrial applications. This paper addresses significant challenges pertaining to sensing and data analytics by connecting a large number of industrial IoT (IIoT) devices and deploying federated learning on 5G edge networks. We envision a federated learning-based 5G edge architecture for IIoT and develop an AI algorithm, i.e., an LSTM autoencoder algorithm for anomaly detection, on the 5G edge. We conduct comprehensive scalability analytics of communication and computation resources on our 5G edge IoT testbed. Our experimentation verifies that 1) federated AI algorithms can be deployed on 5G edge servers for latency-sensitive analytics, and 2) 5G edge supports scalable deployment of IIoT devices with low latency.**

***Index Terms*—Internet of Things, networked edge systems, federated learning, anomaly detection, 5G.**

## I. INTRODUCTION

INDUSTRY'S transition to the digital world has already reaped several benefits, such as fully automated processes and predictive maintenance, which have paved the way for novel service developments and business models. Industry 4.0, supported by the Internet of Things (IoT) and machine intelligence, is a key enabler for many automated processes in modernized industrial applications. IoT enables production monitoring and controlling by collecting data from numerous sensors to increase manufacturing productivity, logistics, and other industrial contexts. Machine intelligence applies Artificial Intelligence (AI) algorithms to understand processes in industrial plants, predict events, and eventually support decision-making.

5G with low latency and high bandwidth plays a crucial role in delivering the data required by AI algorithms for real-time analytics. Additionally, 5G offers the capacity to connect many devices that intermittently transmit data. Edge servers, typically co-located with 5G base stations, allow for AI algorithms to be developed and deployed in proximity to sensors, actuators, and end-users [1]. By processing data collected from resource-constrained IoT devices leveraging computation-intensive AI algorithms at edge, Industrial IoT (IIoT) systems enables efficient decision-making by reducing contact frequency with cloud servers, thus reducing round-trip delay; adhering to local identity management and access control policies; reducing lower communication costs through local processing, and load balancing between the application and network requests based on changes in the edge or core infrastructure, as well as adapting to temporary failures or maintenance.

The proliferation of sensors and connected devices in IIoT has brought heightened privacy concerns, as it necessitates the secure handling of vast amounts of sensitive operational data. For example, IoT-enabled medical devices collect real-time data on users' vital signs and health parameters, which are essential for tracking users' health conditions. However, due to the potential privacy risks, users may be hesitant to allow ML algorithms on centralized servers to analyze their personal data. Therefore, it would be ideal for storing sensitive data on users' trusted edge servers instead of a remote or cloud server. Federated learning (FL), a distributed ML approach, enables the training of ML models on trusted edge servers without transferring data to centralized cloud servers, which enhances the data owner's privacy [2]. Meanwhile, FL is well suited for edge computing applications and can leverage the computation power of edge servers [3]. Significant fundamental challenges exist in designing and implementing distributed FL architectures and systems [4] for IIoT while fully exploiting the computation and communication capacities of the 5G edge. In [5], the authors have proposed integrating blockchain into FL to enhance security and privacy, where blockchain offers permission control for user participation and data encryption, in application scenarios beyond 5G. In [6], the authors target optimization approaches of heterogeneity challenges of FL in 6G network by designing incentive mechanisms, network resource management, and personalized FL approaches. Limited research efforts have been made on deploying FL on 5G edge nodes for IIoT.

Our contributions are twofold. First, we propose an FL-based 5G edge architecture for IIoT connecting a large number of industrial resource-constrained IoT devices and develop FL with an LSTM autoencoder algorithm for anomaly detection on the 5G edge. To the best of our knowledge, this is one of

This research was supported in part by the Academy of Finland (345008 and 326305) and Nordforsk Nordic University Cooperation on Edge Intelligence (168043).

X. Liu, J. Cao, B. Fan, P. Hui and S. Tarkoma are with University of Helsinki, 00014 Helsinki, Finland. E-mail:{xiaoli.liu, jacky.cao, boyu.fan, pan.hui, sasu.tarkoma}@helsinki.fi.

X. Su is with Norwegian University of Science and Technology, 2815 Gjøvik, Norway. Email:xiang.su@ntnu.no.

G. d. Campo, E. Saavedra, and A. Santamaria are with Universidad Politecnica de Madrid, Madrid, Spain. E-mail: {guillermo.delcampo, edgar.saavedra, asun.santamaria}@upm.es.

J. Röning is with University of Oulu, 90014 Oulu, Finland. Email:{juha.roning}@oulu.fi.

P. Hui is also with The Hong Kong University of Science and Technology (Guangzhou), China.

X. Liu and X. Su contribute equally to this work (co-first authors)

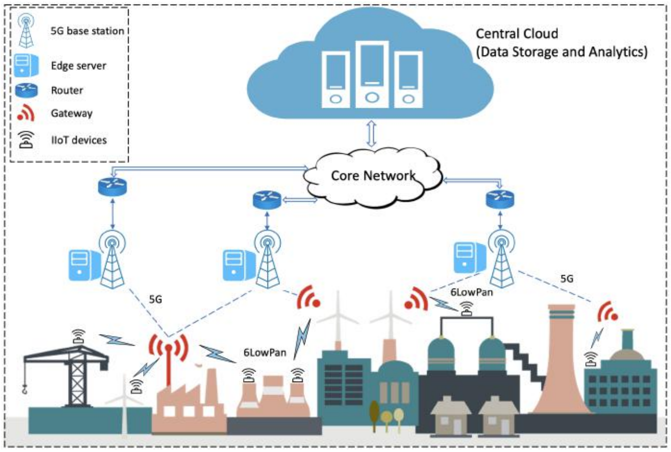


Fig. 1. A conceptual FL 5G edge intelligence architecture for IIoT. Large volumes of data generated by IIoT devices in industrial processes is delivered to edge servers at their proximity for fast data analytics with reliable and high-speed data connections through gateways. 5G and 6LowPan networks allow for massive numbers of connections and provide support for large data transmission. Intelligent FL algorithms facilitate learning and decision-making at the edges of 5G networks by keeping measured data on edges without sharing with the cloud server. The ML model parameters are sent to cloud server for model aggregation.

the first efforts to enable FL on the 5G edge for IIoT. Second, we further demonstrate the feasibility of deploying FL with AI algorithms on a real-world 5G Test Network (5GTN) and conduct comprehensive scalability analytics of communication latency, throughput, and computation resources on 5G edge servers.

## II. FL ON 5G EDGE FOR IIOT: CHALLENGES AND ENABLING TECHNOLOGIES

### A. Challenges

We have witnessed the proliferation of IoT devices in Industry 4.0. Numerous devices in industrial processes monitor and control production by creating a digital twin of the product being realized or the operation being achieved to increase manufacturing productivity, logistics, and many other industrial contexts. The core of the distributed automation systems in Industry 4.0 is reliable information exchange and decision-making [7]. In greater detail, we consider the following critical challenges:

- **Ubiquitous, high-speed, and reliable connectivity for IIoT devices**. IIoT networks present unique challenges compared to conventional IoT networks due to the critical nature of industrial operations and the complexity of industrial environments. Firstly, IIoT applications often require real-time monitoring and control, demanding low latency and high reliability to support critical processes. Secondly, industrial environments can be noisy, with various sources of electromagnetic interference that can degrade wireless communication signals. IIoT networks must overcome signal attenuation and interference challenges to maintain reliable communication between devices. Besides, IIoT networks have to scale effectively to accommodate the growing number of connected devices without sacrificing performance. Novel high-speed and reliable communication technologies are required to provide connectivity to heterogeneous, multi-vendor devices, enable interoperability by offering common software interfaces and compatible protocols, and handle data heterogeneity.
- **Big data streams processing capacities**. IIoT data is a type of big data, which is both large in scale and volume and is also continuous, often with rich time and location dependencies [8]. The potential subsequent integration of multiple sources further amplifies this challenge. Analyzing big data with AI algorithms extracts higher levels of information, guides the understanding of complex situations, and enables real-time analytics to provide user insights. Furthermore, data storage, management, confidentiality, and security also introduce significant challenges.
- **Distributed latency-sensitive, privacy-preserving, and reliable training and inference**. IIoT devices often have constrained and heterogeneous resources, complicating the deployment of complex ML tasks. Privacy-preserving training and inference ensure the protection of user data, but further amplify this challenge. Latency-sensitive decision-making generates insights timely before becoming obsolete. Reliable decision-making typically enforces the system to process significant amounts of data with interpretable AI models. Therefore, efficiency, privacy, and reliability are crucial, and different data models and data aggregation strategies in distributed training and inference must be considered.

### B. Enabling Technologies

AI-powered systems are envisaged to overcome the emerging challenges of Industry 4.0 by fully unleashing the potential of edge intelligence. Figure 1 presents a conceptual architecture of FL on the 5G edge for IIoT.

**Reliable IIoT networks.** Reliable decision-making depends on reliable data from reliable IoT networks. The reliability of IIoT networks requires minimum losses and a low, bounded latency. Table I presents the experimental results from the main IIoT network technologies: Zigbee, NB-IoT, 6LoWPAN, LoRaWAN, Sigfox, BLE and WiFi. Results show the average of 1000 messages that have been obtained by means of a validated Testbed device [9]. The Stability function determines the reliability of a communication technology, accounting for the variability of the latency and the number of losses. Energy consumption values given in milliwatts-second (mWs) represent the specific consumption of only the message transmission for an IoT node. The other energy consumptions are not relevant to compare since they are all the same: a microcontroller in sleep or duty mode with the wireless transceiver deactivated. Note that Sigfox consumption is orders of magnitude higher due to its slow transmission rate (10 seconds on air for a single message) and distance to the receiving end. In fact, energy consumption increases as the range of coverage gets wider although not linearly due to modulation scheme, message size, frequency band, etc. Among them, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN)

TABLE I
COMPARISON OF IOT NETWORK TECHNOLOGIES

| Technology | Zigbee | NB-IoT | 6LoWPAN | LoRaWAN | Sigfox | BLE | WiFi |
|---|---|---|---|---|---|---|---|
| End-to-end latency (ms) | 48 | 1797 | 22 | 397 | 3695 | 27 | 32 |
| Latency standard deviation (ms) | 5 | 1352 | 9 | 5 | 294 | 13 | 9 |
| Error Rate (%) | 0 | 0 | 0.02 | 0.66 | 0 | 0 | 0 |
| Stability (0–1) | 0.592 | 0.036 | 0.924 | 0.993 | 0.922 | 0.002 | 0.036 |
| Energy consumption per message transmission (mWs) | 12.57 | 44.40 | 0.92 | 31.65 | 4024 | 5.05 | 17.02 |

provides a balance between technical properties (latency, error rate) and implementation features (energy consumption, cost). 6LoWPAN enables the use of IPv6 over the IEEE 802.15.4 standard, allowing each IoT device to be accessible via IP address [10]. The IEEE 802.15.4-2015 revision of the standard defines three MAC protocols, including Low Latency Deterministic Network (LLDN), Deterministic and Synchronous Multi-channel Extension (DSME), and Time Slotted Channel Hopping (TSCH), all of which are suitable for industrial applications [11]. LLDN is intended for single-hop low latency networks, such as factory automation. DSME is designed for highly reliable mesh networks, such as predictive maintenance. TSCH provides multi-hop and multi-channel communications, such as process control. Alternatives at the routing layer are mesh protocols, providing higher range and reliability, and star configuration, providing lower latency.

**5G networks.** 5G is an M2M type communications enabler, featuring up to 10 Gbps speed, 1 ms latency, and 100% coverage and reliability. 5G networks aim to support IIoT and provide quality of experience-aware services for industrial applications. The dense deployment of 5G base stations, equipped with antenna arrays, significantly increases the available line-of-sights between IoT devices and 5G antennas. 5G networks have three characteristics, including massive machine-type communications (mMTC), enhanced mobile broadband (eMBB), and ultra-reliable and low-latency communications (uRLLC). 5G networks, with their mMTC characteristic, fulfill the requirement of ultra-dense machine communications by supporting connection densities of one million devices per square kilometer and fulfilling certain quality of service requirements. Thus, mMTC enables the connection of thousands of IoT devices simultaneously. eMBB supports high data rates (exceeding 10 Gbps), which fulfills the throughput requirement when thousands of IoT devices transmit large volumes of data simultaneously and may require gigabytes of network capacity. Finally, 5G offers uRLLC, which fulfills the requirements for extremely low latency communication, allowing for fast data transmission and control messages. Moreover, 5G networks are designed to enhance energy efficiency through lower power consumption for communication compared to other technologies, e.g., LTE-4G networks. Hence, equipping IoT devices with 5G will drastically decrease power requirements.

**FL on edge.** FL is a distributed ML paradigm where a consortium of devices contributes collaboratively to a global neural network model instead of centralizing the data to train a global model. Edge intelligence [12] for IIoT requires integrating heterogeneous IoT data streams, analyzing IoT data with AI algorithms on edge devices, and deriving system-level understanding and knowledge for decision-making. FL on edge servers is optimal for IIoT due to its ability to process data locally on edge devices, diminishing the necessity for data transmission to a central server. This minimizes latency and enables real-time decision-making. Additionally, FL guarantees data privacy and security by keeping sensitive information on-premises, addressing compliance concerns. Its distributed nature also enhances scalability and system robustness, as model training can continue even if certain devices are offline. FL on 5G edge networks harness the advantage of 5G such as high bandwidth, low latency, and increased connectivity, to improve the efficiency and effectiveness of FL system for improved participation, fast model updates and inference, continuous model training without disruption, and enhanced model robustness, which further enable reliable IIoT applications.

## III. EDGE INTELLIGENCE ALGORITHM: A CASE STUDY OF ANOMALY DETECTION

### A. FL Architecture for Anomaly Detection

In this section, we present the development of FL algorithms, i.e., anomaly detection for IIoT data, on edge servers in proximity to IoT devices. Anomaly detection plays an important role in IIoT, such as facilitating early identification of deviation from normal operations in industry machinery, supporting predictive maintenance strategies, and identifying unusual network behaviors within IIoT systems. Implementing federated anomaly detection in edge devices for IIoT data offers several significant benefits. Firstly, it allows for real-time analysis without the necessity of transmitting raw data to a centralized server, thereby reducing network bandwidth usage and latency, enabling quicker anomaly detection and response. Secondly, it enhances data privacy and security by keeping sensitive information localized to the edge devices, mitigating the risk of data breaches and ensuring compliance with privacy regulations. Furthermore, federated anomaly detection improves scalability and fault tolerance, leveraging processing power of edge devices.

Figure 2 presents an overview of the federated anomaly detection architecture for energy monitoring. The collected sensor readings on different edges are denoted by $D_1, D_2, ..., D_k$, where $k$ is the number of edge nodes. The amount of computation is controlled by: *C*, the fraction of edge nodes that perform computation on each round ($C = 1$ corresponding to full-batch gradient descent meaning all the edge nodes participating in the training process); *E*, the number of training passes each edge node makes over its local dataset on each round; and *B*,

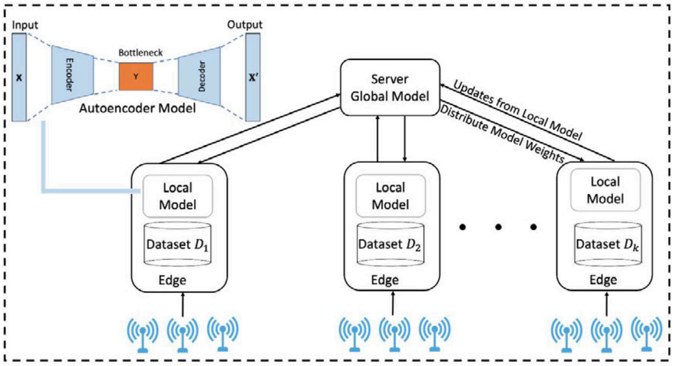


Fig. 2. Federated architecture for anomaly detection where federating participating devices using local data to train a global anomaly detection model instead of sending the data to a central server. The cloud server distributes the current model parameters to selected edge nodes in each training round. Each selected edge node locally trains and updates the local model's parameters by calculating the stochastic gradient descent (SGD). Then the server takes a weighted average of the updates for global model's updates

the local minibatch size used for the edge node updates ($B = \infty$ indicates that the fully local dataset is treated as a single minibatch).

### B. Model

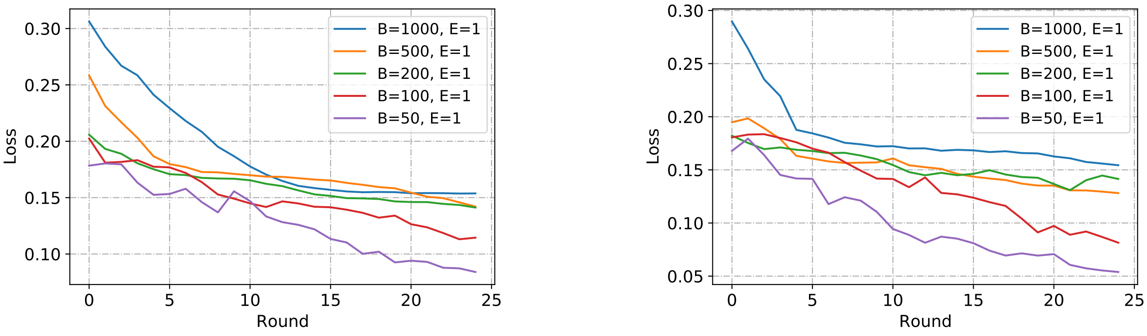
Our anomaly detection model is based on reconstructing the sensing data shown in the top left corner in Figure 2. We design an LSTM autoencoder architecture, a stacked LSTM network consisting of LSTM layers and TimeDistributed dense layers, for energy consumption anomaly detection. Our collected data from IIoT devices is temporal with continuous time series exhibiting periodic (or cyclic behavior) patterns and showing long-term trends. Anomaly is often defined as long-term trends and it is essential that the anomaly detection model could capture the dependence and patterns over continuous time steps. LSTM is selected due to its ability to learn long-term dependence and handle inputs of varying lengths which make it particularly suitable for sequence modeling problems. Meanwhile, the features of LSTM make it easy to train under FL architecture compared to alternative anomaly detection methods. Considering a time series $X = \{x^{(1)}, x^{(2)}, \cdots, x^{(n)}\}$, where each point $x^{(t)}$ in the time series refers to an observed value at time $t$. The model expects inputs of a sequence with $K$ time steps and outputs a sequence with $K$ time steps. The reconstruction errors between reconstruction values $X'$ and real measurements $X$ from sensors are minimized. Specifically, the mean absolute error (MAE) is defined as the loss function for finding the optimized parameters. Adam optimization algorithm with hyperparameters setting $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ is deployed for the LSTM autoencoder. Adam is an adaptive learning rate optimization algorithm that combines the advantages of AdaGrad and RMSProp and has many appealing qualities such as computational efficiency and invariant to diagonal rescale of the gradients. It is widely regarded as being fairly robust to the choice of hyperparameters [13]. On each edge node, the data is partitioned into two parts: the former 70% of the whole data set is used for training, and the remaining 30% is used for testing. Meanwhile, 10% of the training set is used for validation when the best parameters are selected. The reconstruction errors on the training data set are used to formulate the threshold for anomaly detection. A data point on a data set is labeled as an anomaly if the reconstruction error for the data point is greater than the threshold.

### C. Experiments and Results Analysis

We use data from the CeDInt IoT network [14] to design and evaluate the performance of our proposed federated algorithm for anomaly detection. The CeDInt IoT network monitors energy consumption and ambient parameters (temperature, humidity, and presence) within the CeDInt building, including HVAC, lighting and other systems. The federated anomaly detection model is used to detect anomalies within HVAC systems by analyzing related energy consumption and ambient data.

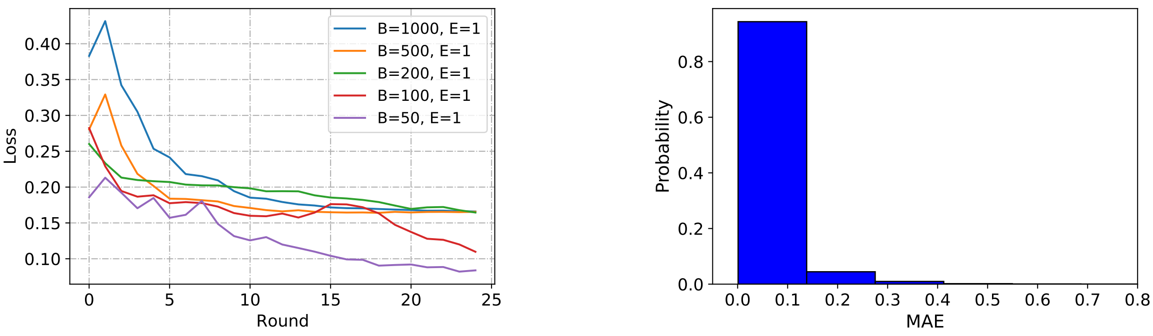
We consider three model structures: 1) two layers with 32 units; 2) two layers with 64 units; and 3) four layers with 64, 32, 32, 64 units. Meanwhile, in each model structure, we fix $C = 1$, $E = 1$, and add more computation per client on each round by decreasing $B$ [2]. Figures 3a-3c present the loss curves of the three model structures with different combinations of ($B$, $E$). Our results demonstrate that 1) adding more local SGD updates per round can significantly decrease training loss. The expected number of updates per client per round is $u = En/B$. We see that increasing $u$ by varying $B$ is effective. We also note that decreasing $B$ is taking advantage of available computation resources of edge devices and this in practice, should be a primary parameter to be tuned; and 2) the model with two LTSM layers with both 64 units has the best learning effect, which means that the neural network structure also affects the training results.

We demonstrate how to use a federated model to identify the anomaly, specifically formulating a threshold. As we assume that all the data points on the training data set are normal, and the max value of the reconstruction errors on the training data set is the worst for our proposed model to reconstruct the data point. Therefore, it is suitable to be used as the threshold. Figures 3d-3f present the process of detecting anomalies in the HVAC scenario. The MAE distribution on the scaled training data set is shown in Figure 3d, and the max MAE value of 0.3 is used for the threshold. Figure 3e presents the dataset where a sudden jump is added to the original dataset, which is used to evaluate the performance of the proposed model and show whether the proposed model could detect the sudden jump as an anomaly and to what extent. The difference is that the values for the time interval $[560, 680]$ in Figure 3e suffer from a sudden jump with 100 W more when compared to the normal testing data set. We calculate the true positive ($TP$), false negative ($FN$), false positive ($FP$), and true negative ($TN$), and use four performance measures, namely precision ($\frac{TP}{TP+FP}$), recall ($\frac{TP}{TP+FN}$), F1 score ($\frac{2TP}{2TP+FP+FN}$), and detection accuracy ($\frac{TP+TN}{TP+FN+FP+TN}$) to evaluate the proposed federated model. $TP$ is the number of points that lie in time interval $[560, 680]$ detected as anomalies, and $FN$ is the



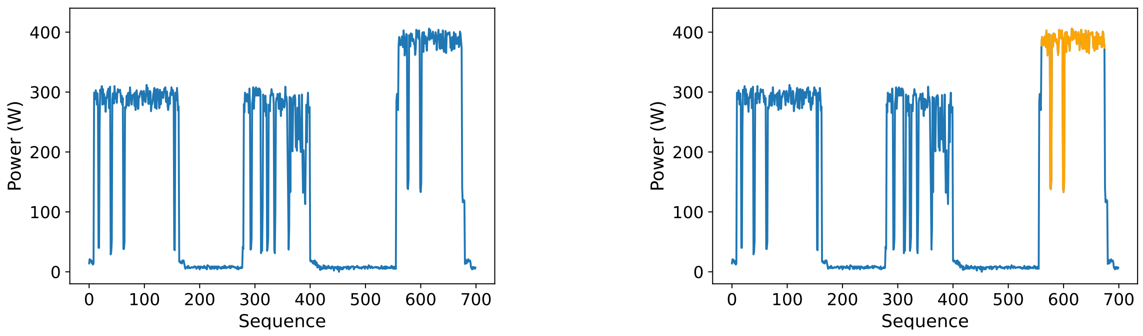
(a) Two layers with 32 units

(b) Two layers with 64 units



(c) Four layers with 64, 32, 32, 64 units

(d) MAE distribution



(e)

(f)

Fig. 3. (a)-(c) Performance of the proposed federated anomaly detection model with different model structures with ($B$ = 1000, $E$ = 1), ($B$ = 500, $E$ = 1), ($B$ = 200, $E$ = 1), ($B$ = 100, E = 1), ($B$ = 50, $E$ = 1); (d) MAE on the scaled training data set in HVAC scenario; (e) an example of the testing dataset containing anomaly; and (f) anomaly detected marked with orange color using the proposed model on testing dataset (e).

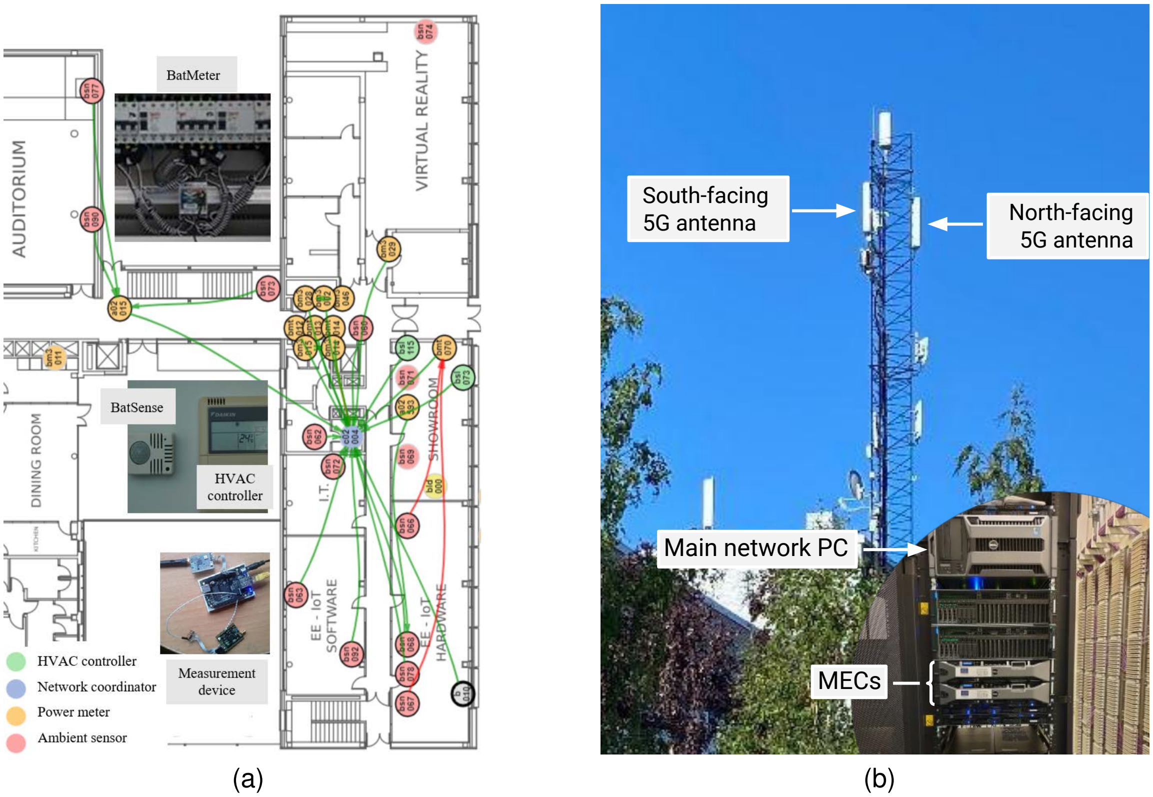
number of points that lie in time interval $[560, 680]$ detected as normal points. Similarly, $FP$ and $TN$ are the numbers of points in time interval $[0, 559]$ detected as anomalies and normal points, respectively. The precision, recall, F1 score, and detection accuracy of the proposed federated detection model on the dataset shown in Figure 3e are 0.954, 0.983, 0.968 and 0.950, respectively, which demonstrate that the proposed federated model can effectively detect anomalies. Figure 3f shows the detected anomalies in time interval $[560, 680]$ by using the proposed federated model marked with an orange color, which further demonstrates the effectiveness of the proposed federated model.

## IV. 5G EDGE EXPERIMENTATION AND ANALYTICS

### A. *Testbed*

This section presents the performance of deploying FL model on a real-world 5G edge IoT testbed. Our testbed is composed of IoT networks and 5G edge testbed (5GTN).

**IoT network:** The IoT network at the CeDInt building is used 1) to evaluate 6LoWPAN reliability and compliance with industrial constraints and 2) to collect IoT data for anomaly detection algorithm deployable on edge servers. Based on previous experimental evaluations, 6LoWPAN outperforms alternative protocols regarding communication latency: 6LoWPAN-measured single node latency is 20 ms,



(a) (b)

Fig. 4. (a) Floor map of IoT sensor deployment in CeDInt building, with the location of the different IoT nodes and communication paths, power meters (BatMeter) installed at electric panelboards, ambient sensors (BatSense) close to a HVAC controller, and the measurement device. (b) 5GTN with MEC server in our experiments.

which outperforms BLE (˜26 ms), WiFi (˜32 ms), Zigbee (˜40 ms), LoraWAN (˜290 ms), and Sigfox (˜3.7 s). Besides, the mesh network topology allows for a dynamic routing configuration, increasing communication range while reducing overhead and error rate. Figure 4(a) presents different elements of the experimental IoT testbed.

**5GTN:** 5GTN is a full-scale 5G micro operator, providing both standalone and non-standalone 5G and LTE connectivity. We conduct our experimentation at the University of Oulu implementation of 5GTN, which has air interfaces of two 5G macrocells (n78), several LTE macrocells (B28, B7, B42), and a LoRa network supporting frequencies [0.7, 2.1, 2.6 and 3.5] GHz. Moreover, our testbed supports heterogeneous wireless technologies, including IEEE 802.11, Bluetooth LE, LoRa, NB-IoT, UWB and LTE evolutions. Edge servers are deployed on 5GTN to support latency-sensitive data analytics. Figure 4(b) presents the 5GTN cell tower and the Multi-access Edge Computing (MEC) server. The edge server in this experimentation has an Intel Core i7-8700 CPU, 32 GB memory, and an NVIDIA GeForce RTX 2080 Ti GPU. To support these experiments, we deploy a Stockholm-based cloud Amazon EC2 server, which has comparable specifications as our edge server, i.e., 8 virtual CPUs, 32 GB memory, and an NVIDIA T4 GPU.

A PC (Intel Core i7-8700K CPU, 32 GB memory, and an NVIDIA GeForce RTX 2080 with Max-Q Design GPU) relays the real collected data from IoT devices. With this approach, we can evaluate massive deployments of devices through multiple parallel threads (one thread representing one device). Data is sent from each thread to a MEC server connected to 5GTN. The PC relaying the data can access the network through a 5G modem.

### B. *Results and analytics*

To analyze the performance of deploying the FL model on the 5G edge for IIoT, we study:

- Data transmission latency, throughput, error rate, and jitter. We separate overall latency into the latency of transmitting data from IoT sensors to a 5G modem with 6LoWPAN and the latency of forwarding the data from the 5G modem to an edge server with a 5G connection.
- Scalability, by increasing the number of emulated devices to deliver data from a 5G modem to an edge server.
- Required computation resources (CPU, GPU, and memory) by the FL trained LSTM autoencoder algorithm on edge servers.

We analyze different communication parameters while increasing the number of neighbors from 1 to 50 in increments of 5. Figures 5(a)-(d) presents results of the 6LoWPAN experimentation. Node hopping increases average latency values compared to single communications (60-70 ms vs 20 ms). Latency and jitter value oscillations are caused by node-hopping and aleatory event-triggered messages and added traffic. We observe that the message error rate is less relevant

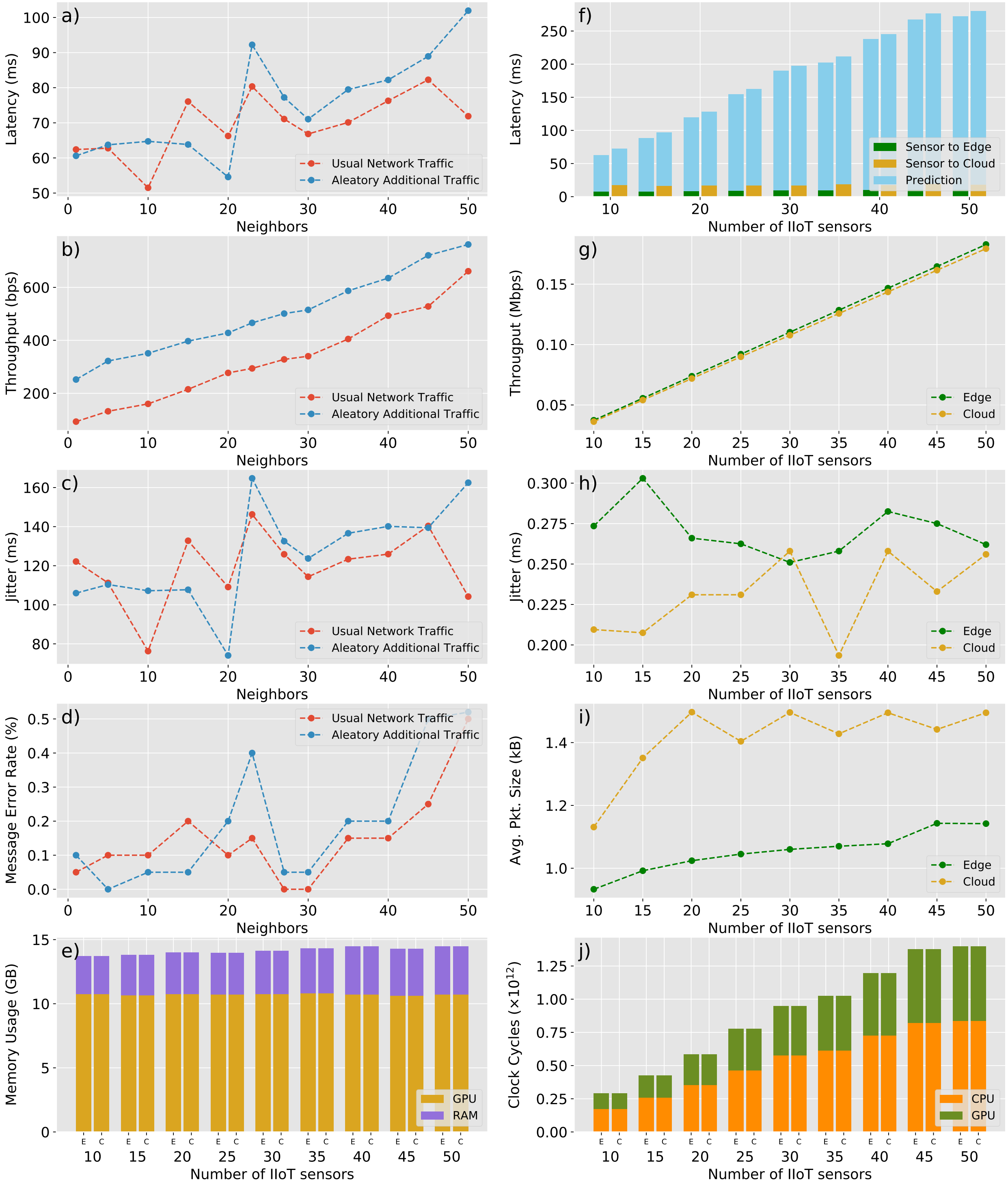


Fig. 5. Left: 6LoWPAN results of the (a) latency, (b) throughput, (c) jitter, and (d) error rate. The red lines represent the usual network traffic (configuration messages, periodic measurement requests from higher-level applications, and event-triggered message transmissions). We add additional aleatory traffic (57 byte messages every 4 seconds, blue lines) to simulate real-world environments and test reliability [15]. Then (e) are the results of the GPU memory and RAM usage for the edge- (E) and cloud- (C) deployed algorithm, with the cloud server requiring an average of 12.7% more memory resources than the edge. Right: (f) cellular 5G latency and the computation time for prediction where the data connection to the edge server has an average latency of $10.5 \pm 2.0$ ms, while the connection to the cloud is approximately 40.3% larger at $17.6 \pm 0.8$ ms, and the prediction latency for both edge and cloud servers is similar, with an average percentage difference of 1.30%. Then the cellular 5G results of the (g) throughput, (h) jitter, and (i) average packet size are presented; as well as (j) the GPU and CPU clock cycles results for the edge- (E) and cloud- (C) deployed algorithm. The edge executes 9.62% more complete clock cycles than the cloud server.

(below 0.5%). Experimental results of latency, jitter, and error rate validate that utilizing 6LoWPAN in industrial applications ensures high-speed, reliable IIoT communications. For experimentation of FL on the 5G edge, we measure the latency, throughput, jitter, and average packet size from sending data from the sensors to a receiving MEC server and a cloud server with a 5G network connection. We vary the number of sensors from 10 to 50 in increments of 5. Figures 5(f)-(i) present the results of the 5G experimental tests. We observe that as the number of IoT sensors increases, the data transfer latency, throughput, and average packet size increase for both edge and cloud connections. For timely data transfer, an edge server is, therefore, the most suitable. For both connections to the edge and cloud server, the throughput has approximately the same values for each sensor number. The jitter shows more variability, e.g., increased jitter at 15 sensors at the edge and at 35 sensors for the cloud, which reflects the dynamic nature of a live cellular network. To test the capabilities of the proposed LSTM autoencoder algorithm, we deploy the algorithm on both an edge and cloud server and compare the required computation resources, i.e., the total prediction latency, memory usage, and the number of clock cycles during algorithm execution. Figures 5(f), (e), and (j) present the results of this experimentation.

## V. DISCUSSION AND OPEN CHALLENGES

5G edge intelligence combines advanced connectivity, compact processing power, and AI at the edges of 5G networks, which presents an emerging trend for Industry 4.0. Federated optimization enables increasingly complex networked systems involving heterogeneous devices, service providers, and network operators to become more intelligent and autonomous in network management with privacy protection. In this paper, we focus on sensing and data analytics by connecting many resource-constrained IIoT devices and deploying a federated AI algorithm on the 5G edge. Our contributions are twofold: 1) we envision FL on 5G edge-enabled IIoT architecture; and 2) we develop and deploy federated LSTM autoencoder anomaly detection on the 5G edge and conduct comprehensive scalability analytics of communication and computation resources on our 5G edge IoT testbed. Our experimentation verifies that 5G edge supports scalable deployment of IIoT devices with low latency and federated algorithm deployed on 5G edge servers for privacy-sensitive analytics.

This study has several open challenges that need further investigation. First, how to efficiently process continuous data streams locally arrived at 5G edge servers, which is vital for real-time analytic and decision making. Second, how to optimize the FL algorithm while to minimize the energy consumption during the FL model training and communication process if energy consumption of edge servers or devices is a significant concern. Third, how to address potential attacks of FL systems on 5G edge, such as gradient leakage and model poisoning attacks, is still a significant concern. Even data is kept locally without sharing and only parameter updates (gradients) are sent during the training process, it is still possible for the adversaries to disclose valuable information or even reconstruct the raw data from the leaked parameter gradients. Adversaries can also attack the model aggregation process to alter the parameters of the global model to corrupt the model, causing the global model to behave undesirably and produce inaccurate predictions.

## ACKNOWLEDGMENTS

This research was supported in part by the Academy of Finland (345008 and 326305) and Nordforsk Nordic University Cooperation on Edge Intelligence (168043).

## REFERENCES

[1] Y. Liu, M. Peng, G. Shou, Y. Chen, and S. Chen, "Toward edge intelligence: Multiaccess edge computing for 5g and internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6722–6747, 2020.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[3] R. Yu and P. Li, "Toward resource-efficient federated learning in mobile edge computing," *IEEE Network*, vol. 35, no. 1, pp. 148–155, 2021.

[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[5] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for 5g beyond," *IEEE Network*, vol. 35, no. 1, pp. 219–225, 2021.

[6] B. Luo, P. Han, P. Sun, X. Ouyang, J. Huang, and N. Ding, "Optimization design for federated learning in heterogeneous 6g networks," *IEEE Network*, vol. 37, no. 2, pp. 38–43, 2023.

[7] P. K. Malik, R. Sharma, R. Singh, A. Gehlot, S. C. Satapathy, W. S. Alnumay, D. Pelusi, U. Ghosh, and J. Nayak, "Industrial internet of things and its applications in industry 4.0: State of the art," *Computer Communications*, vol. 166, pp. 125–139, 2021.

[8] X. Liu, S. Tamminen, X. Su, P. Siirtola, J. Röning, J. Riekki, J. Kiljander, and J.-P. Soininen, "Enhancing veracity of iot generated big data in decision making," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2018, pp. 149–154.

[9] E. Saavedra, L. Mascaraque, G. Calderon, G. Del Campo, and A. Santamaria, "A universal testbed for iot wireless technologies: Abstracting latency, error rate and stability from the iot protocol and hardware platform," *Sensors*, vol. 22, no. 11, p. 4159, 2022.

[10] J. Tournier, F. Lesueur, F. L. Mouël, L. Guyon, and H. Ben-Hassine, "A survey of iot protocols and their security issues through the lens of a generic iot stack," *Internet of Things*, vol. 16, p. 100264, 2021.

[11] L. Alkama and L. Bouallouche-Medjkoune, "Ieee 802.15.4 historical revolution versions: A survey," *Computing*, vol. 103, no. 1, pp. 99–131, Jan 2021.

[12] D. Xu, T. Li, Y. Li, X. Su, S. Tarkoma, T. Jiang, J. Crowcroft, and P. Hui, "Edge intelligence: Empowering intelligence to the edge of network," *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1778–1837, 2021.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[14] G. del Campo, E. Montoya, J. Martin, I. Gomez, and A. Santamaria, "Batnet: A 6lowpan-based sensors and actuators network," in *Ubiquitous Computing and Ambient Intelligence*. Springer, 2012, pp. 58–65.

[15] M. Herlich and C. Maier, "Measuring and monitoring reliability of wireless networks," *IEEE Communications Magazine*, vol. 59, no. 1, pp. 76–81, 2021.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3469988

## VI. BIOGRAPHY SECTION

**Xiaoli Liu** is a research coordinator at the Department of Computer Science, University of Helsinki, Finland. Her research interests include federated learning, edge intelligence, and augmented reality.

**Xiang Su** is an Associate Professor at Norwegian University of Science and Technology, Norway. He has extensive expertise in the Internet of Things, edge intelligence, and extended reality.

**Guillermo del Campo** is an Assistant Professor, Universidad Politecnica de Madrid and the head of the IoT/EE research group at CEDINT-UPM, Madrid, Spain. His research interest include Internet of Things, wireless communications, visible light communications and smart environments.

**Jacky Cao** is a PhD student at the University of Oulu, Finland. He received his MPhys. degree from Durham University in 2019. His research interests include mobile augmented reality, edge computing, and 5G networks.

**Boyu Fan** is a PhD student at the Department of Computer Science, University of Helsinki, Finland. He received a master's degree from Beihang University, Beijing, China in 2020. His research interests include pervasive computing, Internet of Things, and federated learning.

**Edgar Saavedra** is a Ph.D. student at CEDINT-UPM, Madrid, Spain, in the field of sensor wireless networks for energy efficiency in industry, homes, and cities. He earned his master's degree in telecommunications engineering in 2018 from Universidad Politécnica de Madrid, Madrid, Spain.

**Asunción Santamaría** is a Professor at the Universidad Politécnica de Madrid Telecommunications School, Madrid, Spain. She is the director of CEDINT-UPM and has extensive expertise in Internet of Things, data communications, network communication, network architecture, cloud computing, and virtual and augmented reality.

**Juha Röning** is Professor of Embedded System at the University of Oulu, Finland. He is principal investigator of the Biomimetics and Intelligent Systems Group (BISG). He is currently serving as a Board of Director for euRobotics aisbl. He is also a steering board member of ARTMIS-IA. His research interests include computer vision, robotics, intelligent signal analysis, and software security.

**Pan Hui** is a Chair Professor of Computational Media and Arts (CMA), a Chair Professor of Emerging Interdisciplinary Areas, Director of the Center for Metaverse and Computational Creativity, and also Director of the HKUST-DT Systems and Media Laboratory (SyMLab) at the Hong Kong University of Science and Technology. He received his PhD from the Computer Laboratory at University of Cambridge. He has extensive experience on augmented reality, virtual reality, and metaverse. He is an IEEE Fellow, an ACM Distinguished Scientist, and a member of Academia Europaea.

**Sasu Tarkoma** is the Dean of Faculty of Science and a Full Professor at the Department of Computer Science, University of Helsinki, Finland. He is also the Director of the Helsinki Center for Data Science (HiDATA). He completed his PhD in Computer Science at the University of Helsinki in 2006. His research interests include distributed systems, data analytics, mobile and ubiquitous computing, Artificial Intelligence, and 6G.

applied sciences

MDPI

*Article*

# Virtual Reality and Internet of Things Based Digital Twin for Smart City Cross-Domain Interoperability

**Guillermo del Campo *, Edgar Saavedra, Luca Piovano, Francisco Luque and Asuncion Santamaria**

CEDINT-UPM, Universidad Politecnica de Madrid, Campus de Montegancedo sn,
28223 Pozuelo de Alarcon, Spain; e.saavedra@upm.es (E.S.); luca.piovano@upm.es (L.P.);
fp.luque@upm.es (F.L.); asun.santamaria@upm.es (A.S.)
* Correspondence: guillermo.delcampo@upm.es

**Featured Application: Management of Smart City services using Digital Twin concept with the integration of Internet of Things and Virtual Reality. Special focus in interoperability to enable cross-service applications.**

**Abstract:** The fusion of Internet of Things (IoT), Digital Twins, and Virtual Reality (VR) technologies marks a pivotal advancement in urban development, offering new services to citizens and municipalities in urban environments. This integration promises enhanced urban planning, management, and engagement by providing a comprehensive, real-time digital reflection of the city, enriched with immersive experiences and interactive capabilities. It enables smarter decision-making, efficient resource management, and personalized citizen services, transforming the urban landscape into a more sustainable, livable, and responsive environment. The research presented herein focuses on the practical implementation of a DT concept for managing cross-domain smart city services, leveraging VR technology to create a virtual replica of the urban environment and IoT devices. Imperative for cross-domain city services is interoperability, which is crucial not only for the seamless operation of these advanced tools but also for unlocking the potential of cross-service applications. Through the deployment of our model at the IoTMADLab facilities, we showcase the integration of IoT devices within varied urban infrastructures. The outcomes demonstrate the efficacy of VR interfaces in simplifying complex interactions, offering pivotal insights into device functionality, and enabling informed decision-making processes.

**Keywords:** smart cities; Internet of Things; Digital Twin; virtual reality; interoperability

**Citation:** del Campo, G.; Saavedra, E.; Piovano, L.; Luque, F.; Santamaria, A. Virtual Reality and Internet of Things Based Digital Twin for Smart City Cross-Domain Interoperability. *Appl. Sci.* **2024**, *14*, 2747. https://doi.org/10.3390/app14072747

Academic Editors: Ryan Gibson and Hadi Larijani

Received: 7 March 2024
Revised: 21 March 2024
Accepted: 22 March 2024
Published: 25 March 2024

## 1. Introduction

Smart cities represent a paradigm shift in urban living, leveraging advanced technologies to enhance efficiency, sustainability, and quality of life for citizens [1]. However, urban services often operate within isolated silos, each functioning independently with minimal coordination or integration. These silos emerge due to historical development, bureaucratic structures, and specialized expertise, resulting in fragmented delivery of services (e.g., mobility, waste management and street lighting). Each service typically operates within its own department or agency, focusing solely on its specific mandate and objectives, without considering the broader urban context or potential synergies with other services [2]. Breaking down these silos is essential for creating more efficient, resilient, and sustainable urban environments. The integration of Internet of Things (IoT), Digital Twins (DTs), and Virtual Reality (VR) tools plays a pivotal role in shaping the landscape of these futuristic urban environments, enabling not only optimization of municipal services, but also providing valorization of cross-domain applications [3].

The IoT serves as the backbone of smart city infrastructure, connecting devices and systems to gather and analyze data for informed decision-making [4]. IoT facilitates the creation of intelligent transportation systems, environmental monitoring, and smart infrastructure, contributing to enhanced resource management and reduced environmental impact [5]. For example, IoT sensors deployed in traffic lights, public transit vehicles, and roadways can communicate seamlessly to optimize traffic flow, reduce congestion, and enhance the commuter experience. Likewise, IoT-enabled environmental sensors monitor air quality, noise levels, and weather conditions, providing valuable data for urban planners to implement measures to mitigate pollution and improve overall environmental health. [6]. On the other hand, DTs, virtual replicas of physical entities or processes, have emerged as powerful tools in smart city planning and management. These digital replicas enable real-time monitoring, simulation, and analysis, fostering a deeper understanding of complex urban systems and facilitating data-driven decision-making and adaptive strategies [7]. One of the key advantages of digital twins in smart cities is their ability to model and simulate various scenarios, allowing stakeholders to anticipate the impact of different interventions or policies before implementation. For example, city officials can use digital twins to simulate the effects of changes in traffic patterns, urban development projects, or environmental policies, enabling them to make informed decisions that optimize outcomes and minimize risks [8]. Furthermore, digital twins enable cities to proactively manage and maintain infrastructure assets, such as roads or public lighting, by providing real-time insights into their condition and performance [9].

Additionally, VR tools bring immersive experiences to smart cities, offering innovative solutions for urban visualization, management interfaces, citizen engagement, and training. Integrating virtual reality technology with digital twins for smart cities and IoT applications profoundly enhances data management and interpretation, promoting wider participation in urban planning and management. VR crafts a dynamic, data-driven environment that is interactive, immersive, and conducive to collaboration, effectively bridging the physical and the digital, abstract worlds [10,11]. Additionally, it affords unique and enriched perspectives that augment contextual awareness of the represented scenarios. By superimposing extra information onto replicated scenarios, VR deepens environmental/operational understanding through pertinent data and graphical representations [12,13]. Moreover, VR boosts collaboration by facilitating interaction among users within a common space, enhancing collective deliberation and decision-making, and thereby fostering participant synergy [14–16]. Altogether, the synergy between IoT, DTs, and VR tools amplifies the impact on urban planning, governance, and sustainability [17].

While each of these technologies (IoT, DTs, and VR) individually contributes to the smart city ecosystem, their true potential is achieved through seamless integration, i.e., interoperability [18]. The use of a common data model for IoT devices enables consistent interpretation and utilization of data across various platforms, applications, and smart city services. Current efforts among standardization bodies and industry associations are focused on the importance of metadata to build comprehensive data models that enable interoperability [19]. There seems to be an alignment between many of them, creating an ecosystem definition where oneM2M, FIWARE, ETSI ISG CIM, and OMA LwM2M are coexisting and playing a clear complementary role [20]. There are a few research works focused on erasing urban services silos. Most of them aim at the use of semantic interoperability in different application fields such as energy [21], agriculture [22], or building management [23]. Other approaches concentrate on the integration of BIM and IoT, both for building [24], public facilities [25], and cities [26]. Finally, there are initiatives that work above the IoT platform level, focused on data integration [27], analytics [28], and applications [29]. However, to the best of our knowledge, there is not a previous work that proposes the combined use of IoT, DTs, and VR technologies towards breaking down the barriers between different urban service domains to enable integrated and collaborative approaches to city management.

The objective of this paper is to present the design principles and practical implementation of a DT concept for management of cross-domain smart city services. The DT solution, which has been developed in the framework of the IoTMADLab initiative, consists of a VR tool that creates a virtual replica of the real environment and the main functional features of the IoT devices. This virtual environment offers intuitive interaction to perform typical monitoring tasks through Head Mounted Displays (HMDs) and its controllers.

The rest of the paper is organized as follows: Section 2 introduces the IoTMADLab initiative, explains the methodology followed to develop the work, and provides some insights about the proposed IoT network architecture. Section 3 describes the main components of the DT platform, at both physical and virtual level, the technologies used for its deployment and the interconnections between the different elements. Section 4 presents the services currently implemented for both monitoring and controlling urban assets. Finally, Section 5 summarizes the main achievements and benefits of the proposed work while Section 6 discusses the future research lines to expand the DT capabilities.

## 2. IoTMADLab and Methodology

### *2.1. The IoTMADLab*

The IoTMADLab is joint initiative between the Madrid city Council (through the Digital Office) and the Universidad Politecnica de Madrid (through the research center CEDINT), aimed at defining a standardized IoT network model that is open, neutral, and interoperable to facilitate the direct connection between devices (equipment, sensors, and actuators) from different manufacturers and services [30]. It serves as a platform for research, development, and innovation in the field of the Internet of Things (IoT) technology, particularly focused on applications for smart cities. Collaborating closely with the technicians from the different municipal services, service awarded companies, and their respective technological providers, the aim is to define a reference architecture to establish a framework that allows cities to implement and use IoT technology efficiently and effectively while ensuring consistency and interoperability among different devices and systems.

The IoTMADLab facilities are divided into three areas: the IoT Laboratory, the Controlled Environment, and Virtual Reality Laboratory.

- IoT Laboratory: an indoor laboratory equipped with measurement and testing equipment and data representation interfaces. It is where the compatibility, interoperability, and cybersecurity IoT devices and sensors are tested. Compatibility certification requires that the devices fulfill the requirements of the specific municipal service and comply with the communication protocols from the IoTMADLab IoT network reference architecture. Interoperability certification, which demands compiling with the data model, may be achieved at different degrees (depending on the communication protocol. For both certifications, minimum cybersecurity requirements must be met.
- Controlled Environment: an outdoor replica of the urban environment, integrating operational city elements such as streetlights, waste containers, parking lots, and green areas. Once the equipment has been validated in the laboratory, it is deployed and tested in a controlled yet real-world environment. In this area, the devices are exposed to variables and conditions they might encounter in actual urban situations, allowing for monitoring and evaluation of their performance in a more genuine setting. This ensures that issues, incompatibilities, or unforeseen challenges can be identified and resolved before the devices are deployed in an operational urban setting. The impact of interoperability across municipal services is closely examined.
- Virtual Reality laboratory: a facility equipped with the most advanced XR hardware and software for the design and development of pioneering solutions that address complex problems and technological challenges across various disciplines. It specializes in creating immersive VR experiences, being the creation of the digital twin for the IoTMADLab. This initiative seeks to construct an interactive virtual counterpart of the laboratory, utilizing Virtual Reality to achieve precise representation, visualization, and interaction within both tangible and digital domains. To achieve its goal, developers undergo a detailed development process encompassing several key

stages: 3D modeling, realistic texturing, and integrating communication protocols with IoT platforms. The design of user interactions is carefully considered to provide intuitive and engaging experiences.

### 2.2. Methodology

Cities are actively engaging in a digital transformation strategy that positions municipal services as both catalysts and beneficiaries. Interconnection among these services is a must to improve management and enable added value applications. In this work, we propose the combined use of IoT, DTs, and VR to foster collaboration among services.

- As a first step, we conducted a detailed review of the city's current technological infrastructure and services. This assessment aimed to identify areas where enhancements could be made and potential gaps in technological capabilities, particularly in terms of IoT connectivity. This process included mapping out municipal devices, facilities, and equipment that could benefit from IoT technologies, assessing the existing network and communication systems, and evaluating the degree of digital integration within municipal services. The collaboration with the Digital Office of Madrid City Hall was crucial in defining a starting point for strategic planning and the deployment of IoT solutions aimed at enhancing urban life.
- Following this groundwork, the focus shifted to establishing a common and interoperable framework that would allow for seamless communication between IoT devices and other elements within the city's infrastructure and their respective control centers. An IoT network reference architecture and an open data model was proposed to facilitate clear and effective data interpretation and management across various municipal services, thereby improving service efficiency and quality. The main results from this analysis are described throughout Section 2.3.
- In addition to establishing this framework, the development of a digital twin enhanced with virtual reality technology was initiated. This digital twin is designed to simulate real-world scenarios, test IoT functionalities, and exchange information between physical and virtual components in a controlled environment. The objectives of this model include creating a virtual space for managing and controlling an IoT system equipped with interoperable sensors, enhancing decision-making processes through immersive experiences and data visualization, and conducting thorough tests on IoT communications and services before their widespread implementation in smart urban settings. The main features of such proposal are outlined in Section 3.
- A phase of pilot testing and evaluation was outlined to rigorously assess the practical implementation of IoT technologies and the functionality of the digital twin with virtual reality. This phase took place at the IoTMADLab facilities and is crucial for refining data models, communication protocols, and the synergy between the physical and digital worlds, ensuring that the technological infrastructure and its virtual counterpart operate seamlessly together. This step is vital for confirming the system's effectiveness and readiness for a larger-scale deployment, aiming to establish a robust foundation for real-time data exchange and analysis in the context of Smart Urban Spaces (SUS). SUS are pilot areas within Madrid, and are designed in collaboration with city technicians and awarded service companies and answer to real needs of municipal services. The aim of SUSs is to show city governors, technicians, and citizens the potentials of collaboration cross municipal services leveraging IoT interoperability.

### 2.3. IoT Network Reference Architecture

The objective of the IoTMADLab IoT network reference architecture is to establish the requirements in terms of connectivity and semantics for achieving direct interoperability between IoT devices, which may be from different manufacturers, and even from different municipal services. The IoTMADLab focuses on the lower layers of the IoT ecosystem, i.e., at the network level, including the connectivity layer and the data model (see Figure 1).

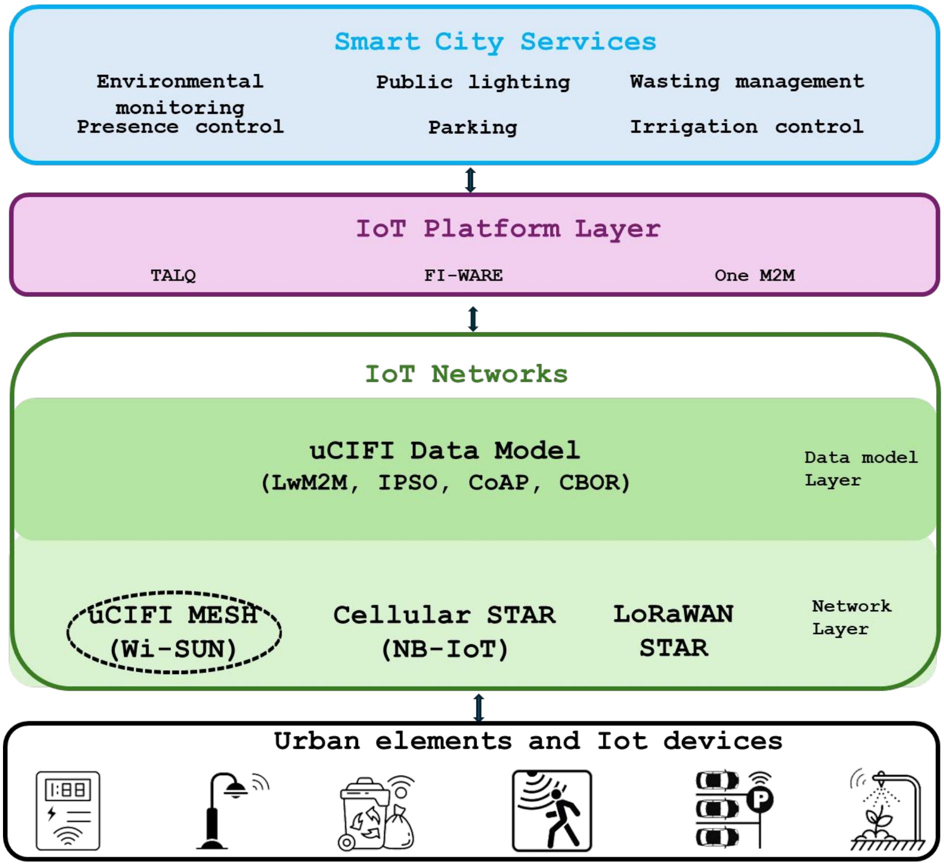


**Figure 1.** IoTMADLab IoT network reference architecture: the focus is put at the network and data model layers (green-colored box). For the data model layer, we use the uCIFI model, while for the network layer, initially Wi-SUN mesh technology is proposed.

In IoT applications for Smart City, one of the necessary requirements is a reduced overhead in the information encoding and the use of lightweight communication protocols to reduce the amount of information transmitted and processing needs so that simple, low-cost, low-power devices can be used. It is also important that the data model and protocols used should be based on open, mature standards that are widely implemented, allowing the development of an ecosystem of products and projects that favors competitiveness and innovation.

### 2.3.1. Data Model

The data model defines the logical structure of the information associated with an asset monitoring, management, and control application such as a city's public lighting system. It can also define the protocols for accessing and communicating this information. To ensure the interoperability of the systems that make up the application and to avoid dependence on a single vendor lock-in, the data model must be based on open specifications standardized by independent entities not affected by commercial interests.

After analysing different approaches, the IoTMADLab has decided to align with the uCIFI data model [31]. The uCIFI alliance is an open, not-for-profit organization whose goal is to develop open standards for enabling device interoperability of devices in Smart City applications. The uCIFI data model defines the use of four protocols: LwM2M, IPSO, CoAP, and CBOR.

- LwM2M (Lightweight M2M), promoted by the Open Mobile Alliance (OMA), is a protocol designed for efficient management of IoT devices, providing standardized communication and data exchange. It enables remote device management, firmware updates, and monitoring, optimizing IoT deployments. LwM2M's lightweight nature ensures compatibility with resource-constrained devices, enhancing scalability and interoperability in IoT ecosystems. This protocol streamlines device management processes, facilitating seamless integration and operation of IoT networks [32].
- The IPSO (Internet Protocol for Smart Objects) Alliance provides a framework for interoperability among IoT devices, defining common data models and communication protocols. By standardizing data representation and exchange, IPSO simplifies device integration and application development in IoT ecosystems. It facilitates seamless interoperability between heterogeneous devices, fostering scalability and innovation. IPSO's approach enhances efficiency and reliability in IoT deployments, promoting the widespread adoption of smart technologies [33].
- CoAP (Constrained Application Protocol) is a specialized web transfer protocol designed for constrained IoT devices, offering lightweight communication with low overhead and high efficiency. It enables devices to exchange data over the internet in a constrained environment, optimizing resource usage. CoAP's simplicity and flexibility make it ideal for IoT applications, supporting reliable, asynchronous communication and resource discovery [34].
- CBOR (Concise Binary Object Representation) is a compact data serialization format derived from JSON, designed for resource-constrained devices in IoT applications. It efficiently encodes complex data structures into binary form, reducing payload size and transmission overhead. CBOR's simplicity and efficiency make it suitable for constrained environments, facilitating seamless data exchange between devices and applications [35].

### 2.3.2. Connectivity Layer Technologies

After analysing the current IoT connectivity technologies, the IoTMADLab has selected those that are (or may be in the future) compatible with the uCIFI data model and that, due to their characteristics, cover all the requirements of the possible use cases from smart city services: Wi-SUN, NB-IoT, and LoRaWAN. For the first phase of the implementation of SUSs, WI-SUN has been chosen as its compatibility with the data model has already been validated.

The Wi-SUN (Wireless Smart Ubiquitous Network) Alliance defines the Wi-SUN FAN (Field Area Network) specification, which has resulted in the IEEE 2857-2021 standard, a mesh network protocol, which works over 6LoWPAN and is based on several IETF, IEEE, and ANSI/TIA standards. 6LoWPAN (IPv6 over Low Power Wireless Personal Area Network) is a communication standard designed to enable connectivity of low-power and resource-limited devices over wireless personal area networks (WPAN) based on IEEE 802.15.4. As it is based on Internet Protocol Version 6 (IPv6), it allows assigning unique IP addresses to each connected device, facilitating its identification and communication in the network.

Hence, communications at the lowest layer of the stack are carried out by means of 6LoWPAN-based technology up to the IoT gateway, where the next step until the server-side platform coordinator is usually conducted via standard Ethernet connectivity, falling back to Wi-Fi or LTE when the lack of wiring becomes a factor. Regardless, this piece within the full-stack communications path is always tunnelled inside a VPN connection so that the actual connectivity infrastructure can be transparent in the deployment process while at the same time ensuring security.

Future IoTMADLab deployments will also be targeting NB-IoT and LoRaWAN at the lowest layer of the stack. This means that the network might even drop the gateways or coordinators when NB-IoT is used. For LoRaWAN use cases, a proper gateway is usually required, so that the stack and logical behaviour of the network shapes that of Wi-SUN's use cases. Nonetheless, that one hop is avoided in the case of NB-IoT, where network gateways become nil, at least functionally and in terms of provisioning.

Wi-SUN implements a meshed network topology, which enables direct communication between devices (expanding coverage) and uses a hierarchical, self-organizing routing approach, where nodes act as routers to relay packets between devices. On the security side, encryption and authentication mechanisms (128-bit AES) are used to protect the communication between the devices and the network. The following communication parameters within the Wi-SUN FAN specification have been defined, ensuring compatibility with uCIFI reference mesh implementation:

- Frequency bands: EU1 (863–870 MHz) and EU2 (870–876 MHz).
- PHY mode of operation: 2a (100 kbps).

## 3. System Description and Main Components

The foundational elements of the Digital Twin infrastructure for management and operational oversight are illustrated in Figure 2. At the Campus of Montegancedo (Universidad Politecnica de Madrid, Spain), the urban controlled environment encompasses pedestrian zones, green areas, a parking zone, and the building where the IoTMADLab is located. At the IoTMADLab facilities, the following urban elements co-exist: LED streetlights, parking lots, watering systems, garbage bins and containers, a bike sharing station, and building systems (HVAC, lighting, electric panel board, and power plugs). The integration of IoT sensors and actuators within these elements provides the data to feed the virtual/digital world. Within this digital world, simulations and predictive analyses of physical elements are conducted. Acting as a conduit between the tangible and digital worlds, the communication network and IoT platforms facilitate seamless integration. The service layer enriches the Digital Twin experience by offering value-added features such as monitoring and control functionalities, which are accessible through virtual interfaces like Head-Mounted Displays (HMDs).

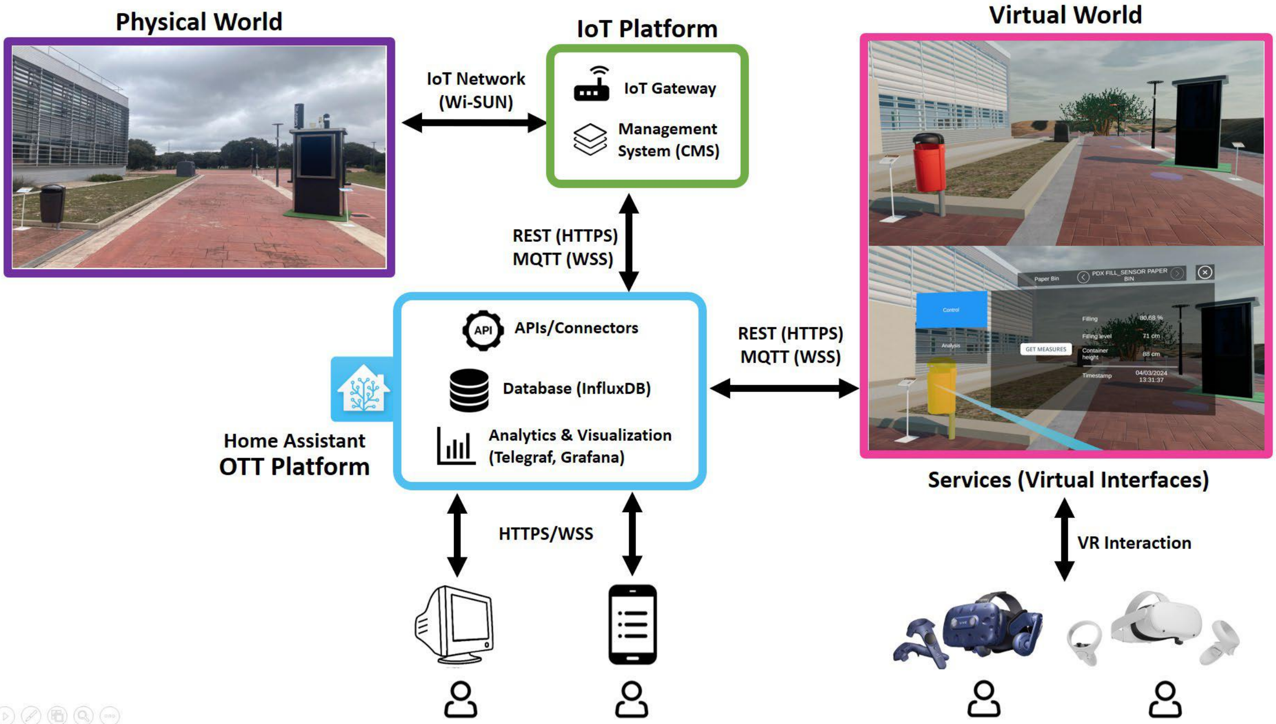


**Figure 2.** Main building blocks for the Digital Twin implemented for the controlled environment at IoTMADLab. Black arrowed lines highlight the communication channels, and they are labeled with the employed protocols or means**.**

### *3.1. The Physical World*

The IoTMADLab is a physical space designed to work with various teams, IoT devices, and protocols to develop a standardized IoT network reference architecture. This architecture aims to establish an open, neutral, and interoperable IoT network that facilitates the direct connection of devices from different manufacturers within the same service, as well as direct connectivity with devices from other services and manufacturers. To achieve this, it mainly consists of an integration and testing laboratory and a pilot installation area. In the laboratory, a series of technical tests are conducted to verify the connectivity and interoperability of devices linked to various IoT sensors and actuators. The pilot installation areas comprise a controlled real-world environment where equipment, whose interoperability has been previously verified in the laboratory, is deployed. In this area, their performance under real conditions is tested, monitored, and the impact of interoperability on city services is assessed.

Figure 3 presents the IoT laboratory and two areas of the controlled environment of the IoTMADLab. The controlled environment encompasses diverse urban elements that are equipped with IoT devices enabling the digitalization interface (sensor/actuator) and the communications to the Internet:

- Streetlights: there are 12 LED streetlights from different manufacturers, each one equipped with IoT nodes, illumination, and motion detection sensors (from different vendors). The IoT nodes are connected through a Zhaga interface, allowing plug&play replacement, while sensors use both Zhaga and internal interfaces. The streetlights can work autonomously, being triggered by illumination and motion/presence values, controlled manually (with remote control commands), or depending on values from other sensors following pre-configured rules.
- Parking lots: there are four parking lots equipped with IoT parking sensors, both below and above ground. The sensors, which are battery-powered, detect if a vehicle stands at parking lot by means of radar and earth magnetism sensors. When a change in the parking status is detected, the device sends a message indicating the new status. To save battery, these sensors remain in sleep mode most of the time, waking up periodically to check the parking occupancy or to send a periodic keep alive message if there is no change in the parking status.
- Waste containers: there are two garbage disposal bins and a public paper recycling container equipped with waste IoT sensors. The devices, which are radar-based volumetric sensors, measure the filling level of the bins/containers. These devices also have GPS to generate alarms in case of detecting large movements with respect to the installation location, a temperature sensor with configurable thresholds to generate alarms in the event of fire, and accelerometers to detect emptying operations and alarms due to vandalism. These devices are battery powered and will provide up to eight years of operation depending on how frequent measurements and transmissions are configured.
- Environmental information: different IoT sensors monitor environmental parameters such as outdoor temperature, relative humidity, wind speed, and solar radiation. They are connected to the electrical mains and allow both periodic measurement and data request.
- Watering systems: three irrigation valves regulate water flow for green areas watering, which may be scheduled depending on weather forecast and actual soil moisture values. Soil temperature and pH sensors provide additional information. Both irrigation valves and soil sensors are plugged to IoT nodes to enable communications.
- Building facilities: the IoTMADLab indoor laboratory activities are monitored and optimized using diverse IoT devices. On one side, IoT power meters are connected to the building electric panel boards, measuring the energy consumption of the individual electric lines. On the other side, HVAC, lighting, and other energy consumers

(e.g., workstations) are managed by means of IoT smart plugs. Additionally, environmental IoT sensors collect indoor temperature, relative humidity, and presence data.


(**a**) (**b**) (**c**)

**Figure 3.** The different physical spaces of the IoTMADLab. (**a**) The integration and testing laboratory; (**b**) One area of the controlled environment with urban elements as streetlights and bike stations; (**c**) Another area with parking lots and green zones.

### *3.2. The IoT Platform*

Synergies between various technologies are essential to conform the backbone of advanced digital twin interactions. The IoTMADLab controlled environment IoT platform is founded on Home Assistant, which serves as the main top-layer application and integrator for the whole system. This thoughtful choice was motivated by several critical factors that align with the goals and requirements of our project, emphasizing interoperability, scalability, and ease of integration with a wide array of IoT devices and protocols. Home Assistant provides an ideal platform to incorporate data from different vendors and third-party platforms, as it is highly expandable, open-source, and modular, while also prioritizing privacy and allowing molding of the local-cloud frontier as one may wish [36].

Some of the decisive key characteristics and benefits of Home Assistant are the following:

- Open-Source and Community-Driven: Home Assistant's open-source nature fosters a vibrant community contributing to its continuous improvement. This aspect ensures the platform evolves in response to emerging IoT trends and technologies, providing a rich ecosystem of plugins, integrations, and support for a broad spectrum of devices and services, being virtually infinite.
- Privacy-Centric and Local (or not) Control: Unlike many IoT platforms that solely rely on cloud-based logic, operation, and algorithms, Home Assistant pushes privacy and local control. This philosophy ensures that data generated by smart city services remain within Local Government's boundaries, mitigating privacy concerns, keeping data close, and lowering dependency on third-party cloud services.
- Extensive Device Compatibility, Flexibility, and Customizability: Home Assistant supports a wide range of devices and protocols, making it an ideal choice for smart city applications. The platform's ability to seamlessly integrate devices across different manufacturers and communication protocols facilitates the development of cohesive and interoperable smart city services. The modular architecture of Home Assistant allows for high degrees of customization and flexibility. Home Assistant can be tailored to meet use cases' specific needs, ranging from environmental monitoring to urban mobility solutions. This adaptability, along with its rapid growth and constant improvement over the last years, extends the platform's applicability beyond home automation to more complex smart city infrastructures.
- Scalability: Home Assistant's basal lightweight and efficient design ensure it can scale to accommodate the growing number of IoT devices and services within smart city

ecosystems. This scalability is crucial for sustaining the dynamic expansion of smart city services and their evolving requirements.

The IoT platform makes use of RESTful APIs and MQTT to communicate with IoT network gateways and management systems (CMS), retrieving data and sending commands. In addition, it integrates a state-of-the-art InfluxDB database for time-series data, which allows a very responsive, efficient, and accountable interaction with live events. By integrating Home Assistant with InfluxDB and leveraging the communication capabilities of RESTful APIs and MQTT, the IoT platform bridges the gap between the physical and virtual worlds in real time, retrieving data to mirror the real world (MQTT) and sending commands to control the physical world from the virtual interface (REST). This integration provides a comprehensive solution for monitoring, controlling, and visualizing the events from IoT devices within the digital twin ecosystem.

The IoT platform also integrates Grafana and Telegraf, allowing for the creation of dynamic and interactive dashboards. These dashboards provide real-time insights into the IoT ecosystem, showcasing trends, patterns, and potential anomalies. Grafana's flexibility in data visualization makes it an indispensable tool for users who seek to understand and analyze the vast amounts of data generated by their IoT devices, while Telegraf provides a rapid and reliable manner of monitoring the performance of the InfluxDB-based system.

### 3.3. The Virtual World

The Virtual Reality (VR) environment, developed using the Unity 3D engine, is centered on a detailed 3D digital representation of the IoTMADLab where the different devices and sensors are located (as depicted in Figure 4). This virtual model accurately reflects the demonstrator layout and is geographically precise as it utilizes GPS coordinates to position urban elements and associated sensors within the virtual landscape. Moreover, the model boasts visual realism, achieved by texturing the 3D assets with high-resolution images of real materials. To ensure a seamless and immersive user experience, the model has been meticulously optimized to meet the hardware specifications and constraints of popular commercial VR headsets. For instance, when running on standalone VR headsets with limited computational power, such as the Meta Quest 2, the model employs simplified geometry and lower-resolution textures. Various optimization techniques have been implemented to maintain smooth performance and minimize VR-induced discomfort, including object culling to render only visible objects, the use of different levels of detail for distant objects, and reducing computationally intensive visual effects like shadows and transparency effects without compromising overall visual fidelity. The software has been implemented following the OpenXR API [37], which supports interacting with VR system in a platform-agnostic way. Nevertheless, it has been fully tested on some of the most prominent headsets of the market like HTC Vive Pro, HTC Focus 3, and Meta Quest 2.

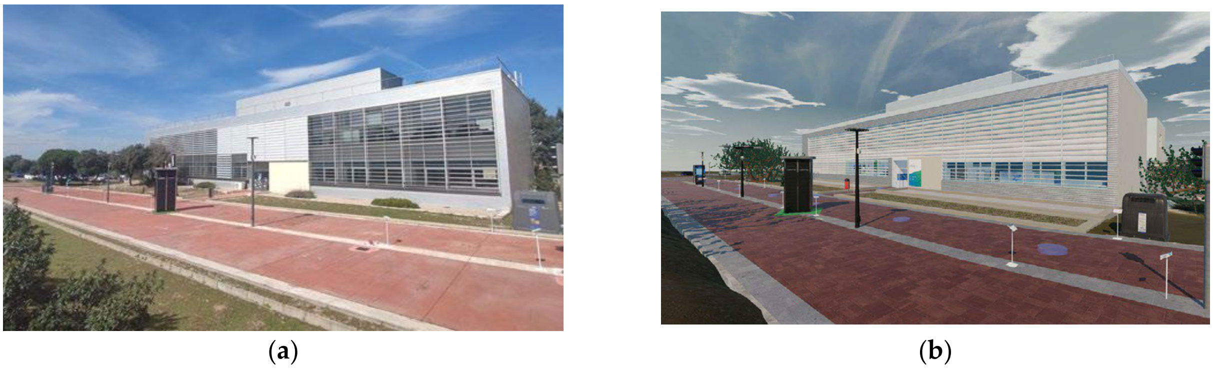

(**a**) (**b**)

**Figure 4.** A view of the IoTMADLab area (Madrid, Spain) where the IoT solution have been deployed: (**a**) The CEDINT building at Campus of Montegancedo hosting real world facilities; (**b**) The realistic virtual world digital version of such environment.

### 3.4. The Virtual Interfaces

Navigation within the virtual environment is optimized for an immersive experience, employing a first-person perspective to enhance realism. Users wearing Head-Mounted Displays (HMDs) can freely explore the virtual space, with the virtual camera adjusting its view based on the user's movements tracked by the HMD's sensors (see Figure 5). To traverse larger distances, users can employ VR controllers to teleport seamlessly to specific locations, situated near the devices (represented by blue teleport cylinders). Alternatively, users can switch to predefined spaces that agglutinate IoT elements: the laboratory space inside the building, the streetlights path, and the parking lot area. The VR controllers facilitate interaction with operational devices; hovering over a device highlights it in yellow, and enabling interaction is as simple as pulling the controller's trigger. Active devices are indicated by a green border and offer a 3D user interface displaying status, control options, and historical charts.

(**a**) (**b**) (**c**)

**Figure 5.** User interfaces for navigation and interaction in the VR-based Digital Twin: (**a**) selecting a scenario to visit. Possible choices are the internal laboratory, the outdoor streetlight zone, and the parking lots. (**b**) Interaction with 3D virtual objects by pointing the controller ray. Interactive objects are highlighted in yellow when hovering. (**c**) Selecting a teleport cylinder to move to a different position (about 4 m away from the current position).

### 3.5. Communication Channels

Figure 6 depicts the interaction between the VR application and the Home Assistant. Utilizing the RESTful API, the login module posts user credentials for authentication and receives an authorization token in return, which must be included in the HTTP header of subsequent requests. The initial state for all devices is acquired through a GET message, comprising GPS coordinates for accurate localization within the virtual environment and the current state values of the IoT devices. Device actuations, such as controlling light dimming, are executed by sending PUT messages to the RESTful API with the new value. Upon receiving a correct response, the device is updated to reflect the real status. Following initialization, synchronization is achieved by receiving state updates through subscription to the MQTT service of the Home Assistant. Lastly, visualization of charts and dashboards for retrieving historical data of the devices is facilitated through an embedded browser communicating with the Grafana dashboard of the Home Assistant.

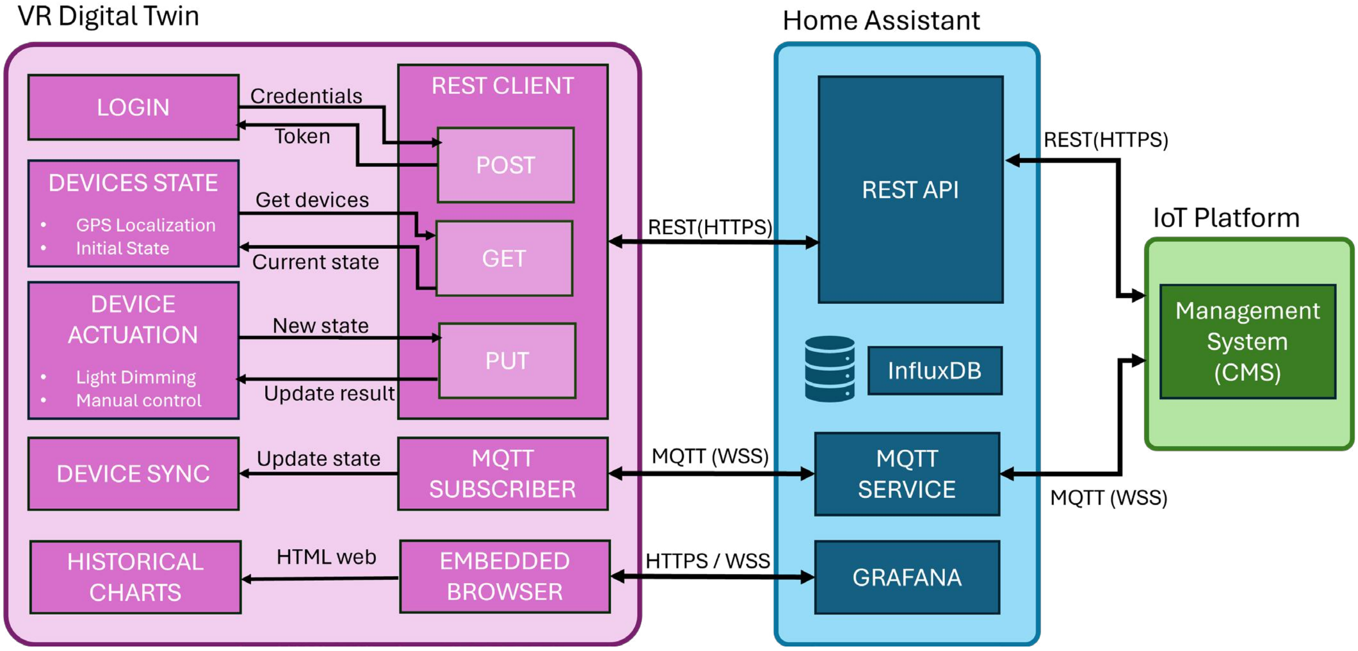


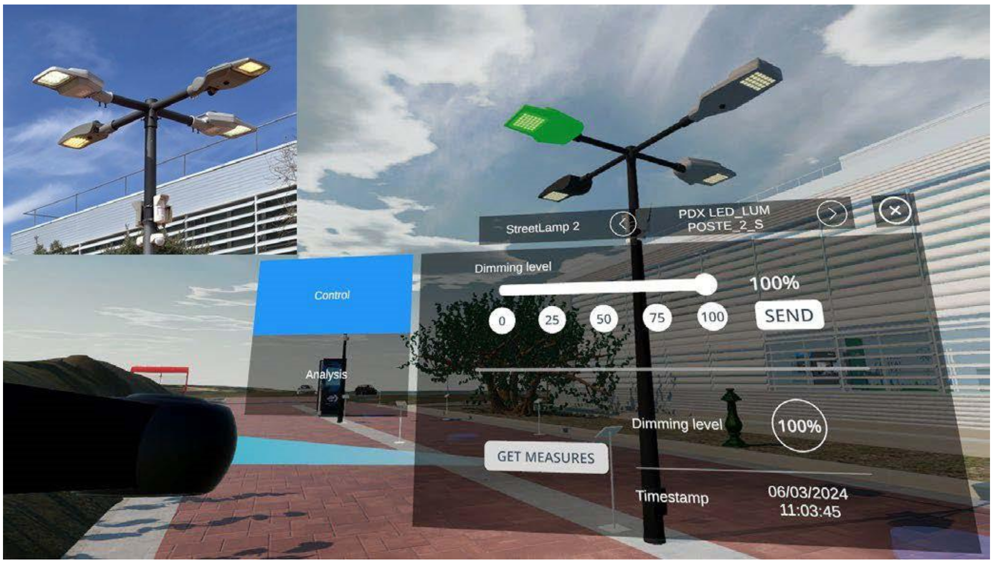
**Figure 6.** Communication channels between the IoT platform (Home Assistant) and the VR Digital Twin. Using a RESTful API, the Digital Twin can collect data and send control commands.

## 4. Implementation of Smart City Services

To demonstrate the capabilities of combining VR technology with a Digital Twin infrastructure for effectively managing the IoT devices deployed within the IoTMADLab controlled environment, the following services have been deployed.

### *4.1. Controlling and Reading Status of IoT Devices*

Users have the capability to interact with the urban elements by means of the connected IoT devices. For example, they can adjust the brightness level of individual streetlights by interacting with a slider integrated into the virtual interface. Alternatively, they can switch between different predefined values by pressing the corresponding dimming button. When a virtual streetlight is selected, the VR environment sends a REST PUT message to the IoT platform to update its current dimming value, ranging from 0% (fully off) to 100% (fully on). Upon confirmation, the VR environment is updated to reflect the new status, and a synchronization message confirms the action along with a timestamp. Correspondingly, the physical streetlight's LEDs adjust accordingly. In the event of communication failure, an error message is displayed, and the dimming value reverts to its last known valid state. This service proves invaluable for targeted maintenance tasks, such as connectivity testing, hardware diagnostics, or selective lighting control. Figure 7 illustrates a user controlling a streetlight's dimming rate, with the real-world lamp's status displayed in the inset of the upper left corner.

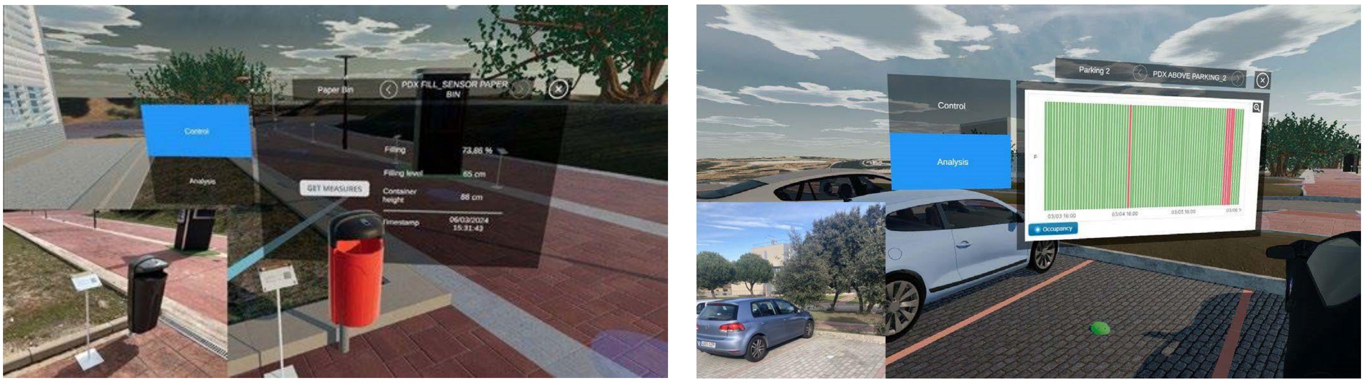


Dimming level 100%

Timestamp 06/03/2024 11:03:45

**Figure 7.** Interacting with the dimming controller to switch on the streetlight directly from the VR environment. The corresponding streetlight in the campus receives the command and automatically updates its status. This interaction highlights the potential of the VR-based Digital Twin in controlling/testing real world services (e.g., public lighting) in a complex urban environment.

Users may also consult the status of the urban elements by accessing real time data from connected IoT sensors. Figure 8 shows the interaction with a garbage bin (left) and a parking lot (right), with the real-world status displayed in the inset of the lower left corner. When accessing the IoT waste sensor, the filling level (in both percentage and volumetric amount) can be checked. For the parking lot, the DT tool shows the real status by locating a vehicle in the position and shows the occupancy patterns during the last days.



Filling 73,86 %

Filling level 65 cm

Container height 88 cm

Timestamp 06/03/2024 15:31:43

(**a**) (**b**)

**Figure 8.** Consulting status of urban elements: (**a**) garbage bin; (**b**) parking lot. The data visualization layer provides users with insights of the time-series data recorded by deployed IoT devices, helping them to enhance their operational understanding of the real environment as well as providing factual information for decision-making tasks.

### *4.2. Visual Analysis of Sensor Data*

The VR application has the capability to access data records from the IoT platform regarding specific variables of interest: e.g., dimming values, parking occupancy, garbage bin filling level, wind speed, soil moisture, or building energy consumption. These data are then visualized using interactive charts within the Grafana IoT platform (Home Assistant). Presented as line charts to depict time series data, each dot represents an average value for a given time range, aiding in trend analysis. Users can interact with the chart similarly to a regular browser, adjusting time ranges, accessing additional information via tooltips, or toggling between available variables. This service facilitates monitoring of sensor operational status across different time scales, identifying regular patterns like day-night alternations, and potential anomalies such as missing data due to communication errors or irregular conditions. Figure 9 exemplifies such monitoring, displaying a temperature sensor within the building facility from night to day, including automatic adjustments based on environmental conditions.

**Figure 9.** Interaction with a line chart showing the average of the temperature values recorded by an environmental sensor during a week. The interactive chart helps identify trends and possible anomalies for monitoring purposes.

### 4.3. Cross-Domain Applications

To showcase the potential of the DT solution for the efficient management of cross-domain city applications, the following test concepts have been implemented:

- Lighting control depending on parking occupation: streetlight dimming level can be controlled based on parking occupancy to optimize energy usage and enhance safety in urban environments. When the IoT sensors deployed in parking lots detect occupancy, nearby streetlights may automatically adjust to higher illumination levels to enhance visibility and security for pedestrians and drivers. Conversely, when the parking lot is no longer occupied, streetlights may dim to conserve energy while maintaining safety standards. This cross-domain application (street lighting and mobility services) not only reduces energy consumption and carbon emissions but also ensures that lighting levels are tailored to actual usage, contributing to more efficient and sustainable urban infrastructure.
- Watering management triggered by presence detection: green areas irrigation systems can make use of data coming from the motion detection sensors equipped at streetlights near green spaces. When presence is detected, the irrigation system may temporarily pause or reduce water flow to prevent bothering passers-by and prevent water waste. On the contrary, when no presence is detected, the system resumes or adjusts irrigation schedules to ensure adequate moisture levels for plant growth. This cross-domain application (watering and street lighting services) not only conserves water but also minimizes runoff and contributes to the creation of greener and friendlier urban spaces.
- Garbage container status signaled by streetlights blinking: streetlights can serve as effective indicators for filling levels or fire risk of garbage container, enhancing waste management efficiency in urban areas. When a garbage container reaches a predefined filling threshold or detects an abnormal temperature value, it sends a signal triggering nearby streetlights to start blinking, signaling to waste collection teams the

need for emptying to fire risk situation. This cross-domain application (waste management and street lighting services) promotes timely waste removal, preventing overflow and littering, and improving overall cleanliness and hygiene in the city.

## 5. Conclusions

In this paper, a VR-based Digital Twin solution to optimize the management of smart city services is presented. The system, whose concept test is developed at the IoTMADLab facilities in Madrid, comprises various urban elements equipped with IoT devices, facilitating seamless integration between the physical and digital worlds, enabling real-time data collection, simulation, predictive analysis, and control functionalities. The physical world component of the Digital Twin infrastructure encompasses a diverse array of IoT-enabled urban elements, such as streetlights, parking lots, waste containers and building facilities. These devices provide critical data on various parameters such as illumination, occupancy, filling level, environmental conditions, and energy consumption.

The IoT platform, based on Home Assistant, serves as the backbone of the Digital Twin infrastructure, facilitating communication between IoT network (gateways and management systems) and the virtual world. Leveraging RESTful APIs and MQTT protocols, the platform ensures seamless data retrieval and command transmission, enabling real-time interaction between the physical and virtual worlds. Integration with InfluxDB database and visualization tools like Grafana enhances data analysis and visualization capabilities, empowering users to gain insights into IoT ecosystem trends, patterns, and anomalies.

The virtual world component offers a detailed 3D digital representation of the IoT-MADLab environment. Geographically precise and visually realistic, the virtual model enables users to navigate and interact with IoT devices seamlessly using VR headsets and controllers. Optimization techniques ensure smooth performance and immersive user experience across different hardware configurations, while OpenXR API compatibility ensures platform-agnostic support for leading VR headsets.

The smooth operation of such solutions, which integrates different technologies (IoT, DTs, and VR), require the use of an IoT architecture based on open and standard protocols. The IoTMADLab proposed a reference IoT network architecture, which serves as a practical example of how these technologies can be deployed to create cross-domain smart city services. It facilitates direct connectivity between devices from different manufacturers and municipal services, promoting interoperability and efficiency in IoT deployments. Moreover, the DT concept, integrated with VR technology, offers intuitive interaction and visualization for monitoring and controlling urban assets.

The novelty and value of our approach lies not in the individual technologies per se but in their integration at the city-wide level. This endeavor actually poses a multitude of complex challenges that demand innovative solutions.

Firstly, achieving seamless integration of IoT, Digital Twins, and VR across an entire urban ecosystem involves navigating a highly fragmented technological landscape. Cities are complex entities with legacy systems and infrastructure that must be retrofitted with new technologies. Ensuring compatibility between old and new systems, while also maintaining the flexibility to incorporate future advancements, requires an understanding of both the technical and socio-economic dimensions of urban environments.

Another major challenge is the creation of a digital twin or a federated group of digital twins that could accurately represent the working of each urban service. This involves not only the technical difficulty of modeling complex urban elements and behavior, but also the logistical and organizational challenges of coordinating between multiple stakeholders, including government areas, private companies, and citizens. Achieving a comprehensive and coherent digital twin that can be effectively used for simulation, visualization, and decision support in VR requires a concerted effort.

Lastly, ensuring the system's sustainability and scalability presents another layer of complexity. As cities evolve, the integrated system must be capable of adapting to changing needs and technologies without requiring complete overhauls. This necessitates forward-thinking design principles that prioritize modularity, interoperability, and energy efficiency.

The integration of IoT, Digital Twins, and VR present several advantages, of which we would like to highlight the user interface and the cross-service applications.

On one hand, accessing VR-based digital twins through Head-Mounted Display (HMD) devices, including VR headsets and AR glasses, unlocks a multitude of possibilities and benefits for urban management and planning. These interfaces offer users an immersive, first-person perspective of these digital, 3D replicas, enabling them to navigate and interact with the virtual environment in real-time. This immersive experience facilitates a deeper understanding of complex urban dynamics, enhances decision-making processes by visualizing the outcomes of various scenarios, and improves stakeholder engagement through interactive and engaging presentations of urban projects. Furthermore, the use of HMD devices in accessing digital twins aids in education and training, allowing city officials, engineers, and the general public to simulate and rehearse responses to emergencies, infrastructure developments, and urban planning strategies with unprecedented realism.

On the other hand, the implementation of city cross-service applications, such as controlling streetlights based on parking occupancy, optimizing watering management triggered by presence detection, and signaling garbage container status through streetlight blinking, demonstrates the potential of the Digital Twin infrastructure for efficient urban management. Besides, it lays the groundwork for the invention and deployment of new applications that provide added value to both the urban technicians, the service awarded companies and, of course, the citizens.

In conclusion, the Digital Twin infrastructure deployed at the IoTMADLab represents a pioneering approach to urban management and operational oversight. By bridging the physical and digital worlds through IoT-enabled devices and VR-based interfaces, the system enables real-time monitoring, analysis, and control of urban elements, paving the way for smarter, more sustainable cities of the future.

## 6. Future Work

Looking ahead, future research should focus on expanding the capabilities of DTs for managing cross-domain smart city services. By further integrating IoT, DTs, and VR tools, cities can unlock new opportunities for innovation, sustainability, and resilience. Collaboration among stakeholders, continued standardization efforts, and investment in technology infrastructure are essential for realizing the full potential of smart cities in improving the quality of life for citizens and fostering sustainable urban development.

While initial tests demonstrate the system's functionality and rapid synchronization between actions performed in the virtual environment and their real-world counterparts, further experimental validation is required. Specifically, there is a need to characterize the timing behavior, focusing on end-to-end latency and error rates. This endeavor will aid in identifying and addressing potential bottlenecks that may arise within the system involving diverse components and technologies.

To support more informed and intelligent management of the urban assets, advanced visualization functionalities are imperative. Thus, we propose incorporating dashboards embedded with visual analysis capabilities to facilitate tasks such as preemptive anomaly detection and identification of optimal lamp distribution. In this context, the virtual environment is expected to enhance charting expressiveness and implement more intuitive interaction interfaces, such as eye or finger tracking.

Future work may also include technical upgrades such as the integration of NB-IoT and LoRaWAN as IoT network communication technologies, the creation of automated connections between the IoT platform (Home Assistant) and the IoT network management systems.

Deploying digital twins within a Virtual Reality framework, while promising, introduces challenges that extend beyond technical complexity and data integration. The economic and technical hurdles stem not only from the need for sophisticated infrastructure capable of handling real-time data but also from the absence of standardized models and frameworks. This lack of standardization hampers interoperability across diverse systems and IoT devices, affecting the efficiency and effectiveness of digital twin solutions in urban settings.

In this context, Building Information Modeling (BIM) technology emerges as a critical component in addressing some of these challenges. BIM offers a standardized approach to the design, construction, and management of buildings and infrastructure, facilitating the creation of digital representations that are detailed, consistent, and easily shareable among stakeholders. When integrated with digital twins and VR, BIM technology can significantly enhance the modeling accuracy, data management, and interactive capabilities of digital twins. For instance, the use of BIM models as the foundation for digital twins in urban planning can streamline the integration of architectural and engineering data, improving the simulation of real-world scenarios, and enabling more precise decision-making.

To overcome potential drawbacks of using HMD devices, such as user discomfort or limited field of view, continuous technological advancements are essential, like those anticipated from several leading tech companies. These new Extended Reality devices are expected to feature enhanced ergonomics, wider fields of view, and more intuitive user interfaces. These improvements will significantly mitigate user discomfort and increase adoption rates. Moreover, integrating these devices with cutting-edge AR technology will enable the overlay of digital information directly onto the physical world, enriching the user's interaction with the digital twin. This integration requires robust data processing capabilities and seamless connectivity to ensure real-time updates and interactions. By harnessing the power of next-generation HMD visors, the access to and interaction with VR-based digital twins can be made more intuitive, inclusive, and effective, marking a significant step forward in the digitalization of urban environments.

Finally, incorporating a 5G communication layer and cybersecurity measures into a VR-based digital twin solution for smart cities is essential for enhancing urban infrastructure and services. The integration of 5G provides ultra-reliable, low-latency communication critical for the real-time data transmission required by digital twins and VR applications. This ensures that urban planners and citizens can interact with the digital twin of the city seamlessly, experiencing immersive simulations with minimal delay.

To protect this sophisticated ecosystem, robust cybersecurity measures must be embedded from the outset. This includes the implementation of end-to-end encryption for data in transit, secure authentication protocols for device and user verification, and regular security audits to identify and mitigate potential vulnerabilities. Advanced threat detection systems, powered by AI, should be employed to monitor network traffic for unusual patterns indicative of cyber threats, ensuring the integrity and confidentiality of the digital twin data.

**Author Contributions:** Conceptualization, G.d.C. and L.P.; methodology, G.d.C.; software, E.S. and F.L.; validation, E.S. and F.L.; formal analysis, L.P.; investigation, all.; resources, G.d.C.; data curation, E.S. and F.L.; writing—original draft preparation, G.d.C.; writing—review and editing, G.d.C., L.P. and A.S; visualization, L.P.; supervision, A.S.; project administration, A.S. and G.d.C.; funding acquisition, A.S. and G.d.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by the IoTMADLab project (a collaborative initiative between the Madrid City and Universidad Politecnica de Madrid and funded by Business Forum for Madrid) and the MOBILITIES for EU project (HORIZON EU, grant number 101139666).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

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

## References

1. Gracias, J.S.; Parnell, G.S.; Specking, E.; Pohl, E.A.; Buchanan, R. Smart Cities—A Structured Literature Review. *Smart Citie*s **2023**, *6*, 1719–1743. https://doi.org/10.3390/smartcities6040080.
2. Soe, R.F. Smart Cities: From Silos to Cross-Border Approach. *IJEPR* **2018**, *7*, 70–88. https://doi.org/10.4018/IJEPR.2018040105.
3. Jafari, M.; Kavousi-Fard, A.; Chen, T.; Karimi, M. A Review on Digital Twin Technology in Smart Grid, Transportation System and Smart City: Challenges and Future. *IEEE Access* **2023**, *11*, 17471–17484. https://doi.org/10.1109/ACCESS.2023.3241588.
4. Bellini, P.; Nesi, P.; Pantaleo, G. IoT-Enabled Smart Cities: A Review of Concepts, Frameworks and Key Technologies. *Appl. Sci.* **2022**, *12*, 1607. https://doi.org/10.3390/app12031607.
5. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for Smart Cities. *IEEE Internet Things J.* **2014**, *1*, 22–32. https://doi.org/10.1109/JIOT.2014.2306328.
6. Ramírez-Moreno, M.A.; Keshtkar, S.; Padilla-Reyes, D.A.; Ramos-López, E.; García-Martínez, M.; Hernández-Luna, M.C.; Mogro, A.E.; Mahlknecht, J.; Huertas, J.I.; Peimbert-García, R.E.; et al. Sensors for Sustainable Smart Cities: A Review. *Appl. Sci.* **2021**, *11*, 8198. https://doi.org/10.3390/app11178198.
7. Deng, T.; Zhang, K.; Shen, Z.-J. A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. *J. Manag. Sci. Eng.* **2021**, *6*, 125–134; ISSN 2096-2320. https://doi.org/10.1016/j.jmse.2021.03.003.
8. Shahat, E.; Hyun, C.T.; Yeom, C. City Digital Twin Potentials: A Review and Research Agenda. *Sustainability* **2021**, *13*, 3386. https://doi.org/10.3390/su13063386.
9. Errandonea, I.; Beltrán, S.; Arrizabalaga, S. Digital Twin for maintenance: A literature review. *Comput. Ind.* **2020**, *123*, 103316. https://doi.org/10.1016/j.compind.2020.103316.
10. Segovia, M.; Garcia-Alfaro, J. Design, Modeling and Implementation of Digital Twins. *Sensors* **2022**, *22*, 5396. https://doi.org/10.3390/s22145396.
11. Jones, D.; Snider, C.; Nassehi, A.; Yon, J.; Hicks, B. Characterising the Digital Twin: A systematic literature review. *CIRP J. Manuf. Sci. Technol.* **2020**, *29*, 36–52. https://doi.org/10.1016/j.cirpj.2020.02.002.
12. Al-Ali, A.R.; Gupta, R.; Zaman Batool, T.; Landolsi, T.; Aloul, F.; Al Nabulsi, A. Digital Twin Conceptual Model within the Context of Internet of Things. *Future Internet* **2020**, *12*, 163. https://doi.org/10.3390/fi12100163.
13. Li, X.; Liu, H.; Wang, W.; Zheng, Y.; Lv, H.; Lv, Z. Big data analysis of the internet of things in the digital twins of smart city based on deep learning. *Future Gener. Comput. Syst.* **2022**, *128*, 167–177. https://doi.org/10.1016/j.future.2021.10.006.
14. Stacchio, L.; Angeli, A.; Marfia, G. Empowering digital twins with eXtended reality collaborations. *Virtual Real. Intell. Hardw.* **2022**, *4*, 487–505. https://doi.org/10.1016/j.vrih.2022.06.004.
15. Bononi, L.; Donatiello, L.; Longo, D.; Massari, M.; Montori, F.; Stacchio, L.; Marfia, G. Digital twin collaborative platforms: Applications to humans-in-the-loop crafting of urban areas. *IEEE Consum. Electron. Mag.* **2022**, *12*, 38–46. https://doi.org/10.1109/MCE.2022.3214944.
16. Jamei, E.; Mortimer, M.; Seyedmahmoudian, M.; Horan, B.; Stojcevski, A. Investigating the Role of Virtual Reality in Planning for Sustainable Smart Cities. *Sustainability* **2017**, *9*, 2006. https://doi.org/10.3390/su9112006.
17. Albino, V.; Berardi, U.; Dangelico, R.M. Smart Cities: Definitions, Dimensions, Performance, and Initiatives. *J. Urban Technol.* **2015**, *22*, 3–21. https://doi.org/10.1080/10630732.2014.942092.
18. Pliatsios, A.; Kotis, K.; Goumopoulos, C. A systematic review on semantic interoperability in the IoE-enabled smart cities. *Internet Things* **2023**, *22*, 100754; ISSN 2542-6605. https://doi.org/10.1016/j.iot.2023.100754.
19. Chaturvedi, K.; Kolbe, T.H. Towards Establishing Cross-Platform Interoperability for Sensors in Smart Cities. *Sensors* **2019**, *19*, 562. https://doi.org/10.3390/s19030562.
20. Jara, A.J.; Serrano, M.; Gómez, A.; Fernández, D.; Molina, G.; Bocchi, Y.; Alcarria, R. Smart Cities Semantics and Data Models. In Proceedings of the International Conference on Information Technology & Systems (ICITS 2018), Libertad City, Ecuador, 10–12 January 2018; Rocha, Á., Guarda, T., Eds.; Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2018; Volume 721. https://doi.org/10.1007/978-3-319-73450-7_8.
21. Teixeira, B.; Pinto, T.; Silva, F.; Santos, G.; Praça, I.; Vale, Z. Multi-Agent Decision Support Tool to Enable Interoperability among Heterogeneous Energy Systems. *Appl. Sci.* **2018**, *8*, 328. https://doi.org/10.3390/app8030328.
22. Aydin, S.; Aydin, M.N. Semantic and Syntactic Interoperability for Agricultural Open-Data Platforms in the Context of IoT Using Crop-Specific Trait Ontologies. *Appl. Sci.* **2020**, *10*, 4460. https://doi.org/10.3390/app10134460.
23. Dimara, A.; Vasilopoulos, V.-G.; Papaioannou, A.; Angelis, S.; Kotis, K.; Anagnostopoulos, C.-N.; Krinidis, S.; Ioannidis, D.; Tzovaras, D. Self-Healing of Semantically Interoperable Smart and Prescriptive Edge Devices in IoT. *Appl. Sci.* **2022**, *12*, 11650. https://doi.org/10.3390/app122211650.

24. Jiang, S.; Jiang, L.; Han, Y.; Wu, Z.; Wang, N. OpenBIM: An Enabling Solution for Information Interoperability. *Appl. Sci.* **2019**, *9*, 5358. https://doi.org/10.3390/app9245358.
25. Mannino, A.; Dejaco, M.C.; Re Cecconi, F. Building Information Modelling and Internet of Things Integration for Facility Management—Literature Review and Future Needs. *Appl. Sci.* **2021**, *11*, 3062. https://doi.org/10.3390/app11073062.
26. Deprêtre, A.; Jacquinod, F.; and Mielniczek, A. Exploring Digital Twin Adaptation to The Urban Environment: Comparison With Cim To Avoid Silo-Based Approaches. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* **2022**, *4*, 337–344. https://doi.org/10.5194/isprs-annals-V-4-2022-337-2022.
27. Lau, B.P.L.; Marakkalage, S.H.; Zhou, Y.; Hassan, N.U.; Yuen, C.; Zhang, M.; Tan, U.X. A survey of data fusion in smart city applications. *Inf. Fusion* **2019**, *52*, 357–374. https://doi.org/10.1016/j.inffus.2019.05.004.
28. Atitallah, S.B.; Driss, M.; Boulila, W.; Ghézala, H.B. Leveraging Deep Learning and IoT big data analytics to support the smart cities development: Review and future directions. *Comput. Sci. Rev.* **2020**, *38*, 100303. https://doi.org/10.1016/j.cosrev.2020.100303.
29. Buhnova, B.; Kazickova, T.; Ge, M., Walletzky, L.; Caputo, F.; Carrubbo, L. A cross-domain landscape of ICT services in smart cities. In *Artificial Intelligence, Machine Learning, and Optimization Tools for Smart Cities*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 63–95. https://doi.org/10.1007/978-3-030-84459-2_5.
30. The Madrid IoT Laboratory. Available online: https://iotmadlab.es/en/the-laboratory/ (accessed on 7 March 2024).
31. uCIFI Data Model. Available online: https://ucifi.org/ucifi-data-model/ (accessed on 7 March 2024).
32. Datta, S.K.; Bonnet, C. A lightweight framework for efficient M2M device management in oneM2M architecture. In Proceedings of International Conference on Recent Advances in Internet of Things (RIoT), Singapore, 7–9 April 2015; pp. 1–6. https://doi.org/10.1109/RIOT.2015.7104900.
33. Jimenez, J.; Koster, M.; Tschofenig, H. IPSO smart objects. In *Position Paper for the IOT Semantic Interoperability Workshop*;OMA Specworks, Sand Diego, CA, USA 2016.
34. Shelby, Z.; Hartke, K.; Bormann, C. *The Constrained Application Protocol (CoAP) (No. rfc7252)*; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2014.
35. Bormann, C.; Hoffman, P. *Concise Binary Object Representation (cbor) (No. rfc7049)*; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2013.
36. Home Assistant. Available online: https://www.home-assistant.io/ (accessed on 6 March 2024).
37. The OpenXR Specification. Available online: https://registry.khronos.org/OpenXR/specs/1.0/html/xrspec.html (accessed on 6 March 2024).

Information Systems Frontiers
https://doi.org/10.1007/s10796-023-10409-2

# Monitoring Framework for the Performance Evaluation of an IoT Platform with Elasticsearch and Apache Kafka

**Gonzalo Calderon**[1] · **Guillermo del Campo**[1] · **Edgar Saavedra**[1] · **Asunción Santamaría**[1]

Accepted: 18 May 2023

## Abstract

IoT platforms are in charge of extracting and processing the data that come from IoT networks, generating additional value, and providing access to the user through usable interfaces. However, the ever growing number of devices, networks, services and applications within the IoT ecosystem, and the recently adopted edge/cloud architecture, increase the complexity. Therefore, IoT platforms should integrate monitoring and visualization tools to facilitate deployment, management and maintenance tasks. In this work, we present the implementation and performance evaluation of an IoT modular platform for distributed architectures that combines the use of Elastic Stack tools (Elasticsearch, Kibana and Beats) and Apache Kafka. We have developed a monitoring framework based on Beats agents that supervise the platform performance attending to different metrics; and adapted the Kibana visualization tools to provide friendly and accessible information to platform administrators and users. Finally, we have deployed and evaluated the IoT platform in four real use cases, identifying the factors that affect the performance of the different modules: Edge Node, Data Streaming, Cloud Server and Search Engine.

**Keywords** IoT platfom · Elastic Stack · Elasticsearch · Kibana · Beats · Monitoring

## 1 Introduction

Internet of Things (IoT) is a relevant and necessary technological sector for the efficient management of resources in multiple environments. The International Data Corporation (IDC) forecasts that total spending on IoT solutions will continue to grow and Huawei expects 100 billion IoT connections by 2025 (Espinoza et al., 2020). Besides, the COVID-19 pandemic has affected the social and financial needs, increasing the value of the IoT applications and impacting on the market trends (Singh et al., 2020; Nasajpour et al., 2020). The continuous development of IoT solutions and their adaptation to different sectors has led to the emergence of heterogeneous architectures, networks, devices, protocols, middlewares, data models, and applications (Kazmi et al., 2017). In this way, IoT platforms should anticipate the growth and diversity of IoT components and data, and provide enough scalability and performance to manage and process them.

Initially, IoT platforms were implemented for specific use cases based on monolithic architectures. They are efficient to handle low workloads, and are easy to develop, suffering few integration, connection and configuration problems (Gos and Zabierowski, 2020). However, billions of IoT nodes have been deployed due to the expansion of the sector and the possibilities offered by technology (Borgia, 2014). These IoT networks generate large amounts of data at different tiers, so monolithic architectures are inefficient because they do not have enough resources to process the data. Therefore, in order to achieve higher performance, distributed architectures have

---

This paper is an extended version of the conference paper: Management and Monitoring IoT Networks through an Elastic Stack-based Platform; Gonzalo Calderon, Guillermo del Campo, Edgar Saavedra, Asuncion Santamaria. In Proceedings of the 20218th International Conference on Future Internet of Things and Cloud (FiCloud), Rome, Italy, 223-25 Aug. 2021; pp. 184-191.

✉ Gonzalo Calderon
g.calderon@upm.es

Guillermo del Campo
guillermo.delcampo@upm.es

Edgar Saavedra
e.saavedra@upm.es

Asunción Santamaría
asun.santamaria@upm.es

[1] CeDInt-UPM, Universidad Politécnica de Madrid, Campus de Montegancedo, Pozuelo de Alarcón 28223, Madrid, Spain

emerged based on specialised services that are distributed across multiple servers (Cravero, 2018).

With the deployment of distributed architectures, the complexity of the IoT platform has increased, requiring mechanisms to detect and anticipate possible errors in its modules. In this work, we propose a solution that centralizes the storage, management and visualization of the systems and services of an IoT platform based on Elastic Stack (open-source). It is a distributed and replicated IoT platform that can be used with a large volume of data in real-time, obtaining high performance. Thanks to the monitoring provided by this solution, better use of resources is achieved, thus increasing the energy efficiency of this IoT platform. Additionally, it can be applied to different use cases. Finally, the proposed solution is validated in four use cases: smart building, smart lighting, smart greenhouse and smart home.

This paper is structured as follows: In Section 2, we review the properties of available IoT platforms that include monitoring and visualization tools. In Section 3, we provide an overview of the platform architecture and present its modules. In Section 4, we describe the monitoring framework, including the metric agents. In Section 5, we report the experimental results to evaluate the performance in real use cases. Finally, in Section 6, we summarize the work and introduce future improvements.

## 2 Related work

The IoT ecosystem has become more complex and heterogeneous due to the use of different layers, technologies and multiple areas of application (Guth et al., 2018). Therefore, IoT platforms, on top of extracting and processing the data that come from IoT networks -generating additional value, and providing access to the user through an usable interfaces- should integrate monitoring and visualization tools to facilitate its deployment and maintenance.

Generally, IoT platforms present a modular architecture (network, servers, services). Hence, to detect intrinsic technical and performance issues, IoT platforms should automatically collect metrics and operational data from every module. This process, which is called monitoring, allows the platform administrator to know the real time status and availability of the modules. Without these automated monitoring systems, the alternative is to continuously look at customer charts and logs, a poor practise incompatible with scalable and profitable IoT platforms.

There are some open source solutions for monitoring, such as Nagios, Zabbix or Netdata. Nagios provides monitoring and supervision of desktop and server operating systems, collecting metrics to evaluate the performance (Barth, 2008). It controls the status of different services or processes and records the occurring events. Mostly, it is used for monitoring of mail, web, application and database services (Anusas-Amornkul and Sangrat, 2017). Zabbix enables verification mechanisms to check the availability and response level of standard services without installing any software on the monitored host. It allows the monitoring of performance statistics and real-time notification mechanisms. Its main use is to examine the states of network services, servers and devices (Tader, 2010). Netdata is a monitoring agent oriented to be installed in servers, containers, and IoT devices. It provides insights using interactive web dashboards in real-time of the events that happen on the servers or on the services (López-Peña et al., 2020). The main problem of using these monitorization tools is that they require independent configuration and deployment to get the metrics of the devices, services, and servers of the IoT platform.

Furthermore, IoT platforms integrate a visualization tool to get an overview of the ecosystem and allow the platform administrator to perform maintenance quickly and efficiently. Existing open-source solutions for visualization are designed to provide powerful charts and graphs from a wide range of data sources, especially those based on time series. The specific visualization tool should be selected based on the application and the database implemented to store the data. The most widely used open source visualisation tool is Grafana (Chakraborty and Kundan, 2021). The great advantage of Grafana is that it can be connected to the most popular time-series databases, including cloud monitoring providers. It provides several dashboards to display data (Venkatramulu et al., 2021). The use of Grafana has been discarded because we would need the implementation of an external plugin to enable the connection between Grafana and Elasticsearch -the chosen search engine for this work-. As an alternative, we propose the use of Kibana, which is already prepared for Elasticsearch.

Nowadays, there are IoT platforms available in the market that include both visualization and monitoring. These platforms may be deployed local or in the cloud. Local platforms are more difficult to scale and maintain but they provide the advantage of having full control over the platform and the data stored on it (Lee et al., 2020). In contrast, with the growth of big data (Darwish et al., 2019), Cloud computing platforms are becoming more popular because they offer dynamic storage, scalability, and replication across multiple data centres.

Three of the most used, supported, and well-known cloud providers, Amazon Web Service, Google Cloud and Azure, offer their own IoT platforms. Although these platforms are well suited to a large number of application fields such as industry 4.0 (Sahay et al., 2019; Adhikaree et al., 2017; Pasha, 2016), smart homes (Jaya and Hossain, 2018; Kang et al., 2017; Zúñiga-Prieto et al., 2018) or commercial buildings (Pelle et al., 2019), their providers are not open source, limiting their adaptation of specific application requirements and singularities.

In this work, we propose an alternative, by adapting a search engine, Elastic Stack (Elasticsearch, Kibana and Beats) to develop an IoT platform. The most common use of Elastic Stack tools is Security Information and Event Management (SIEM) (Kotenko et al., 2017), though, during the last few years has been expanded to other use cases such as IoT (Bajer, 2017). With Elastic Stack tools, different IoT data hubs have already been developed to store the data in smart cities or buildings (Talaş et al., 2017; Dharur and Swaminathan, 2018). All these solutions ingest the data through Logstash, which requires reading an output file or making HTTP requests to an external service to get the data. Instead, we present a novel contribution by using Apache Kafka for data exchange between the IoT network and the database, facilitating the platform configuration and management processes.

## 3 IoT platform architecture

In this work, we tackle the challenge of implementing a distributed and replicated, open-source, IoT platform, which not only provides the essential functionalities to manage the IoT networks but also integrates new components to maintain the platform. In order to achieve this general objective, we have defined the following specific requirements in terms of functionality and services to efficiently manage and monitor IoT networks through the IoT platform:

a) *Store the measurements of the devices*: the platform should store device measurements in JSON documents. The measurements must be processed, fitted to the data model, and structured into Elasticsearch indexes for persistent storage. Elasticsearch should be scalable horizontally due to the huge amount of data it will receive and manage (Patti and Acquaviva, 2016)
b) *Create visualizations with the measurements*: once the measurements are stored in Elasticsearch, they ought to be visualized in Kibana to get a better understanding of the stored data (Peddoju and Upadhyay, 2020). For this purpose, the data should be analyzed and filtered using queries to display it in dashboards that best suit the data type.
c) *Manage and act on the devices*: the IoT platform may allow users to manage and act easily on their IoT networks through a web interface, receiving real-time information about the IoT devices (Babun et al., 2021). Security to data access has to be guaranteed through user registration is required to guarantee security to data access.
d) *Collect metrics from servers and services*: a monitoring system should be implemented to get information in real-time about the operating system and its associated hardware to assess the real-time operation and performance of the modules, services and applications (Renita and Elizabeth, 2017)
e) *Centralize log data of the platform*: the events that occur in the platform should be collected to analyse and evaluate the development and maintenance of the platform (Ahmed et al., 2020).
f) *Check the status of the services*: the IoT platform should implement a centralized system to check the status of the modules in real-time. Services ought to be periodically monitored to verify which ones are running properly.
g) *Capture inbound network traffic on the platform*: it should detect possible intruders and control the latency of the application (Vaarandi and Pihelgas, 2014). Network traffic (web access, response times and maps) must be captured and enriched by adding geo-ip coordinates with the location where the transaction is originated.

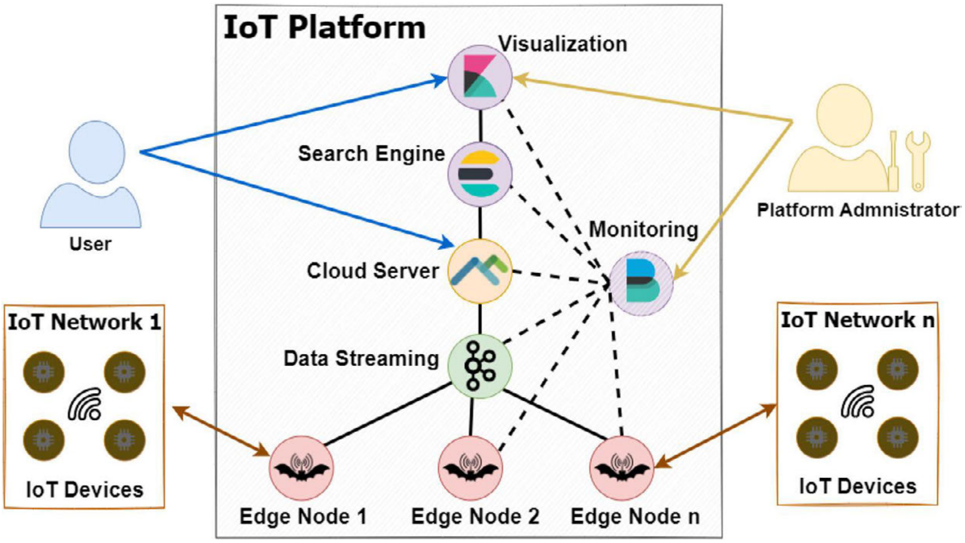
The IoT platform presented in this work, which is conceptually described in Fig. 1, is based on the traditional layer framework used for commercial and open source architectures (Guth et al., 2016), adapting it to incorporate the Elastic Stack tools.

Elastic Stack is an open source distributed system that collects data from any source in any format for searching, analyzing, and displaying it in real-time. Elastic Stack is formed by Elasticsearch, a distributed search and analytics engine responsible for indexing the data (Gormley and Tong, 2015); Beats, a metric agent that ensures collecting, aggregating, and enriching the data; and Kibana, a front-end application for managing the stack and for interactively exploring, viewing and sharing data.

The Edge Nodes concentrate communications from and to the IoT Networks, interconnecting IoT devices with the Cloud Server through the Data Streaming module. The Cloud Server allows acting on the devices through a graphical interface and sends the measurements to Elasticsearch to be stored. Beats agents, deployed in all the modules of the platform, collect metrics, logs, and network traffic statistics and send them to Elasticsearch. Finally, collected data is displayed on dashboards using the Kibana tools.

Our IoT platform integrates two new components to centralize the maintenance of the platform: a Monitoring framework based on Beats agents, which collects metrics from the different modules; and a Visualization module,

**Fig. 1** System description



which enables to overview the status and the performance of the platform.

### 3.1 IoT Devices/Networks

Different IoT devices, including energy meters, ambient sensors, light dimmers, or blind controllers either collect data or control physical assets. There are different IoT networks that can use either KNX, a wired protocol that works over Ethernet and communicates with the Edge Node through a KNXnet/IP protocol (Langels, 2008); or BatNet, a 6LoWPAN-based wireless communication protocol (Del Campo et al., 2012).

### 3.2 Edge Node

The Edge Node implements the drivers to communicate with the different IoT network technologies: KNX and BatNet. Besides, it transfers data and functionalities from the Cloud Server to the IoT Devices and vice versa. The Edge Node executes **Bat Network Manager (BatNM)**, a self-developed java web application in Eclipse Jetty with a Model View Controller (MVC), which runs in a Java Virtual Machine (JVM). BatNM communicates with the IoT networks via Constrained Application Protocol (CoAP) (Shelby et al., 2014), and streams the measurements to the Cloud Server through an Apache Kafka Producer, and receives the order of the users through a Kafka Consumer.

### 3.3 Data Streaming

The Data Streaming module includes **Apache Kafka**, an open-source distributed, partitioned and replicated stream platform that effectively process data in real-time. It includes Zookeeper, a centralised service to maintain and provide robust synchronisation within Kafka brokers, themes and partitions (Guth et al., 2010).

Apache Kafka runs to communicate the Edge Nodes and the Cloud Server, and vice versa. It guarantees security and reliability in the data transmission thanks to its customized protocol over TCP/IP. It also provides higher throughput delivering messages across multiple brokers to allow durability and data availability (Kreps et al., 2011). Kafka's architecture, based on producers and consumers, provides independence between topics and guarantees the order of the messages.

### 3.4 Cloud server

The Cloud Server module executes **Spatia**, a self-developed java application deployed in an Apache Karaf modulith runtime environment. Spatia ensures the communication between Elasticsearch and Edge Nodes by: a) creating a data model to store the measurements; and b) integrating a web user interface, based on a Representational State Transfer (RESTful) framework, to interact with the devices. Spatia sends requests by the user to the Edge Nodes through a Kafka Producer; and a Kafka Consumer receives the measurements that are organised in structured JSON documents to store them in Elasticsearch.

### 3.5 Search Engine

**Elasticsearch** is a distributed search engine that automatically distributes the data across the cluster nodes, offering horizontal scalability. Elasticsearch indexes, searches, sorts, and filters documents, instead of row or column data like relational databases. This is a different way of thinking about data and it is a very important characteristic to perform a complex full-text search efficiently in real-time.

In Elasticsearch, data is organised into indexes. Each index is composed of shards. Each shard is an instance of a Lucene index, which is a search engine in itself, indexing and managing queries on a subset of the data in an Elasticsearch cluster. There are two types of shards: primaries and replicas (copy of primaries). Replicas provide redundancy of the data in case of node failures. Every time a document is indexed, the information is replicated in multiple shards. When the cluster grows or shrinks, Elasticsearch automatically migrates the shards to re-balance the nodes. We have developed a cluster of Elasticsearch parallel nodes to distribute the workload. Elasticsearch clustering allows the IoT platform to have enough processing and storing capabilities to index the data generated by IoT networks and monitoring processes.

### 3.6 Visualization

Kibana provides the tools for the Visualization module. It is used to manage, explore, visualize, and create dashboards in real-time with the data stored in Elasticsearch. Kibana interface offers a wide variety of plugins to develop multiple visualizations:

- *Discover*: it displays the stored data organized into indexes on a time axis. In this tool, the user can make queries, apply filters, and select the period of time used as a temporal axis for the displayed values.
- *Visualize and Dashboard*: it allows users to create different types of graphs, plots, or tables adapted to the nature of the indexed data.
- *Maps*: it displays the data obtained from Packetbeat to visualize the inbound network traffic in a personalized map with multiple layers. Additionally, we have created another visualization to display the devices by their geographical coordinates, previously configured in Spatia.
- *Metrics*: it displays the data obtained from Metricbeat. In this way, the platform administrator visualizes in detail the metrics of each server or service.
- *Logs*: it displays the data obtained from Filebeat. Therefore, the platform administrator visualizes the log files of each system of the platform.
- *Uptime*: it displays the data obtained from Heartbeat. Thus, the platform administrator checks the status of services of the platform.
- *Stack Monitoring*: it shows the overall state of the cluster and its deployed services.

## 4 Monitoring framework

Monitoring framework encompasses multiple Beats agents that collect operational data from different source within the IoT platform modules. Beats are open source data shippers installed as lightweight agents on the servers or containers

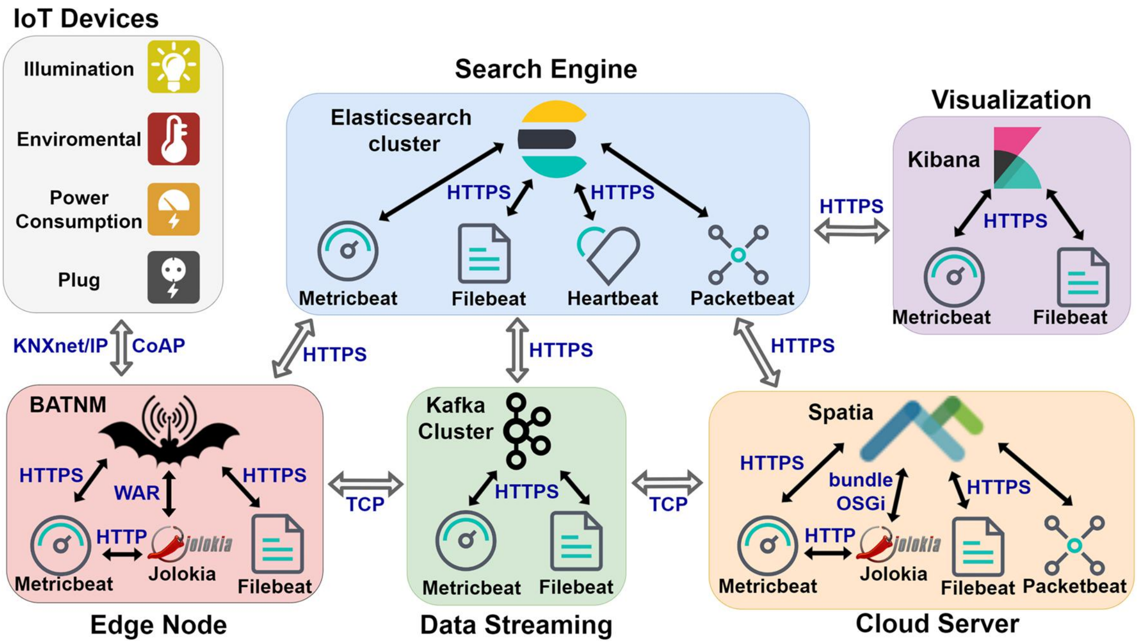


**Fig. 2** IoT Platform architecture

for collecting operational data and forward them to Elasticsearch. We have integrated the following Beats agents:

- **Metricbeat** is an agent installed on the server to periodically collect metrics and performance statistics from the operating system and the services running on the server.
- **Filebeat** is an agent installed on the server to forward and centralize log data. It monitors the log files to collect the log events and forward them to Elasticsearch.
- **Heartbeat** is an agent installed on the server to periodically check the status of the services and determine their availability. It informs the user if the deployed services are reachable.
- **Packetbeat** is a real-time network packet analyzer that monitors applications and services to evaluate their performance. It analyzes the network to obtain metrics such as latency, response times, number of accesses, errors, or transactions.

Figure 2 shows the agents implemented in the IoT platform architecture to monitor and evaluate its performance.

As described in Section 3, the IoT platform is divided into six modules: IoT Devices and Networks, Edge Nodes, Data Streaming, Cloud Server, Elasticsearch Cluster, and Visualization. Although the five modules -IoT Devices and Networks module is out of scope- are provided with Metricbeat and Filebeat specific agents, the metrics and logs of each virtual machine are collected for all the modules.

Metricbeat collects the status of each virtual machine every 5 minutes. The metricsets we have used to evaluate performance are:

- *CPU statistics*: idle, irq, user, system, iowait, softirq, cores, nice, steal, and total.
- *CPU load data*: cores.
- *Memory statistics*: swap, total, used, free, and actual.
- *Network metrics for interfaces*: in, out, errors, dropped, bytes, and packets.
- *Process execution data*: state, memory, cpu, and cmdline.
- *General process summary*: unknown, dead, total, sleeping, running, idle, stopped, and zombie.
- *Usage statistics for each CPU core*: idle, irq, user, system, iowait, softirq, cores, nice, steal, and total.
- *Disk IO metrics*: io, read, and write.

Nevertheless, we have also collected other metrics such as: *Used sockets summary*, *TCP events*, *Systemd service data*, *Filesystem metrics data*, *Fsstat metrics data*, *Uptime metrics data*, and *RAID metrics data*.

Filebeat collects the logs files created by the Unix/Linux logging service. We compile these logs:

- *Syslog*: a standardized way of producing and sending log and event information about the systems.
- *Authlog*: a registration of the activities that require authentication.

### 4.1 Edge Node metrics

Three agents have been deployed to supervise the Edge Node module: Jolokia, Metricbeat, and Filebeat.

a) **Jolokia** is an open-source project that provides a JMX-HTTP bridge as an alternative to JSR-160 connectors. It makes get requests to the JVM in which BatNM is running to know its performance. The JVM statistics collected are:
    - *HeapMemoryUsage* gets the the percentage of busy memory.
    - *Non-HeapMemoryUsage* gets the percentage of free memory resources.
    - *ThreadCount* gets how many parallel process are ran apart from the main thread.
    - *SystemCpuLoad* evaluates system utilization by percentage of CPU usage.
    - *SystemLoadAverage* gets the amount of computational work of the JVM.

b) **Metricbeat** collects and adapts the performance data and statistics obtained from Jolokia to send them to Elasticsearch.

c) **Filebeat** organizes and sends the log files generated by log4j, logging library for Java Gülcü (2003), from BatNM to Elasticsearch. A new event log is generated every time that an event occurs in the Kafka Producer or Consumer, and when the service is started or stopped.

### 4.2 Data Streaming metrics

Two Beats agents have been deployed to monitor the Data Streaming module: Metricbeat and Filebeat.

a) **Metricbeat** is configured to send metrics and statistics from the Apache Kafka cluster to Elasticsearch. Metrics are collected every ten seconds. The metricsets are:
    - *Partition*: it fetches the data for the leader partitions. Data for the replicas are not collected.
    - *Consumer Group*: it gets the cluster consumers and the offset of the messages.

   Also, we set Metricbeat to collect the metrics from the the ZooKeeper service coordinator, integrated into each broker of the Apache Kafka cluster, to Elasticsearch. The collected metrics are the results of executing the admnin keywords *Mntr, Srvr, and Cons.*

b) **Filebeat** organizes and sends the logs from Apache Kafka to centralize them in Elasticsearch. The filesets are:

- *Controller.log*: it stores the events, processes, and states of the cluster. Moreover, it monitors the znodes of Zookeeper.
- *Server.log*: it records the logs generated in the broker.
- *State-change.log*: it stores the changes in the state of the cluster.
- *Kafka-*.log*: it records cluster requests and authorization events.

## 4.3 Cloud Server metrics

Four agents have been deployed to assess the Cloud Server module: Jolokia, Metricbeat, Filebeat, and Packetbeat.

a) **Jolokia** makes get requests to the JVM in which Spatia is running to know its performance. The JVM statistics collected are the same as in the Edge Node module.

b) **Metricbeat** collects and adapts the performance data and statistics obtained from Jolokia to send them to Elasticsearch.

c) **Filebeat** organizes and sends the log files from Spatia to Elasticsearch. A new event log is generated every time that an event occurs in the Kafka Producer or Consumer. Additionally, Filebeat sends an event log when the service is started or stopped.

d) **Packetbeat** captures and monitors the network traffic of Spatia application web to ensure high levels of performance and security. This agent is configured to monitor only the network interface connected to the Internet. The monitored protocols are:

- *ICMP*, v4, and v6 traffic is supervised.
- *HTTP* requests to Spatia are controlled.
- *TLS*, SSL traffic are monitored.

## 4.4 Search Engine metrics

Three Beat agents have been deployed to manage the Search Engine module: Metricbeat, Filebeat, Packetbeat, and Heartbeat.

a) **Metricbeat** forwards the Elasticsearch Cluster performance statistics every 10 seconds. We use the following metrics:

- *Ccr*: it fetches metrics about cross-cluster replication.
- *Cluster_stats*: it fetches information about the cluster.
- *Index*: it fetches data of the indexes.
- *Node*:iIt fetches the cluster nodes information.
- *Node_stats*: it fetches the cluster nodes statistics.
- *Shard*: it fetches information about shards.

b) **Filebeat** collects the logs in the Elasticsearch Cluster. The filesets are:

- *Gc.log*: records the events that happened in the JVM where Elasticsearch runs.
- *Server.log*: it records incidents in the cluster.
- *Audit.log*: it stores the security-related events such as authentication failures and refused connections.
- *Deprecation.log*: controls the availability of the cluster.

c) **Heartbeat** monitors the services of the IoT platform, checking the availability of the services from a list of URLs. Heartbeat checks the services through the following protocols:

- *ICMP*: it checks the status of the service by sending an Echo Request. ICMP protocol messages are sent every 30 seconds to check the availability of the Edge Node, Data Streaming, Cloud Server, and Search Engine.
- *TCP*: it verifies the endpoint by sending and/or receiving a custom payload. TCP connections are established every 30 seconds to check the availability of the services: BatNM, Spatia, Apache Kafka, Zookeeper, Metricbeat, Filebeat, Heartbeat, Packetbeat, Elasticsearch, and Kibana.
- *HTTP*: it confirms that the host returns the expected response. HTTP connections are established every 30 seconds to check the services responses of BatNM and Spatia.

## 4.5 Visualization metrics

Three Beat agents have been deployed to manage the Visualization module: Metricbeat, Filebeat, and Packetbeat.

a) **Metricbeat** sends metrics and statistics of the Kibana interface to store them in the cluster. Metrics are collected every ten seconds. The metricsets are:

- *Stats*: it collects the stats metricset.
- *State*: it collects the state metricset.

b) **Filebeat** forwards the logs generated in the Kibana web interface to store them in the cluster.

c) **Packetbeat** monitors network traffic to ensure high levels of performance and security. This agent is configured to monitor only the network interface connecting to the Internet. The monitored protocols are: *ICMP, HTTP, and TLS*.

## 5 Results and discussion

In order to evaluate the performance of the IoT platform, we have implemented it into four real use cases located at the Campus de Montegancedo from Universidad Politécnica de Madrid: Smart Building, Smart Lighting, Smart Greenhouse, and Smart Home. Each use case implements different IoT networks (BatNet and KNX) that integrate multiple IoT devices (sensors and actuators). Communication with the IoT devices can be done by: a) subscription to the device, in which BatNM keeps listening for a possible call from the IoT device when it detects an event; and b) GET request, where BatNM ask periodically for a specific resource of the IoT device.

1. **Smart Building**: the CeDInt-UPM building behaviour is monitored by means of 21 ambient IoT sensors (temperature, humidity, illumination and presence). The IoT sensors send measurements to subscribers every time a value exceeds a preset threshold. Building systems energy consumption is monitored by 34 three-phase power meters, which send their measurements every five minutes using a get request.
2. **Smart Lighting**: there are 69 IoT devices that control the dimming level of the streetlights based on illumination and presence detection measurements. These values are transmitted every five minutes by get request.
3. **Smart Greenhouse**: there are 10 environmental IoT sensors that send measurements by subscription, 5 power meters and 25 led lighting controllers that transmit the data every 5 minutes by get request.
4. **Smart Home**: there are 4 blind controllers, 10 light dimmers, 4 plugs, and 5 ambient devices. All of them are configured to measure every five minutes and send data through get request.

### 5.1 Platform deployment

We have deployed the IoT platform on an HP ProLiant DL380 Gen9 Intel Xeon E5-2620V4 16GB server at the CeDInt-UPM data center. All modules are mounted on virtual machines running Debian OS. Table 1 presents the resources of each module.

- **Edge Node**: BatNM runs in a JVM hosted on a virtual machine that has 4 GB of RAM memory, 2 CPUs, and 50 GB of storage.
- **Data Streaming**: it is composed by an Apache Kafka cluster of three brokers (version 2.3.0). The resources of each broker are: 80 GB to store the messages produced during the established retention policy of 5 days, 4 GB of RAM memory and 2 CPUs to be used dynamically depending on the workload.
- **Cloud Server**: Spatia is implemented in a JVM hosted on a virtual machine with 4 GB of RAM memory, 2 CPUs, and 200 GB of storage.
- **Elasticsearch Cluster**: to improve the performance of the cluster, we configure 12 Elasticsearch nodes (version 7.10.0), which are divided into the following roles:
  - *1 Coordinating node*: it acts as orchestrator that balances traffic across the cluster by deciding which node should make each specific task. As it is the only node in the cluster in which the data stream does not flow before being indexed, it also hosts Kibana. It runs with with 5 GB of RAM memory, 2 CPUs, and 100 GB of storage.
  - *5 multi-purpose nodes*: they simultaneously play different roles: master-eligible (tracking and mapping), data (storage), ingest (pre-processing), machine learning (running ML jobs), and transform (adaptation). We have selected an odd number of master-eligible nodes to avoid split-brain issues. These nodes are deployed with 6 GB of RAM memory, 2 CPUs, and 100 GB of storage.
  - *6 data nodes*: these nodes act as a data warehouse, that is, they only have the function of storing the information in the cluster. They run with with 4 GB of RAM memory, 2 CPUs, and 500 GB of storage.

### 5.2 Performance evaluation

To evaluate the performance of the IoT platform in real-world applications, we have monitored its performance by storing the data coming from the IoT devices of the 4 use cases for

**Table 1** IoT platform resources

| Module | Service [Version] | RAM [Heap][a] (GB) | Storage (GB) | CPU[2] |
|---|---|---|---|---|
| **Edge Node** | BatNM | 4 | 50 | 2 |
| **Data Streaming** | 3 Apache Kafka [2.3.0] | 4 | 80 | 2 |
| **Cloud server** | Spatia | 4 | 200 | 2 |
| **Elasticsearch Cluster** | 1 Coordinating [7.10.0] | 5 [3] | 100 | 2 |
| | 5 multi-purpose [7.10.0] | 6 [3] | 70 | 2 |
| | 6 data [7.10.0] | 4 [2] | 500 | 2 |

[a] JVM max heap size

[b] Maximum CPU limit, it is fixed dynamically depending on the workload

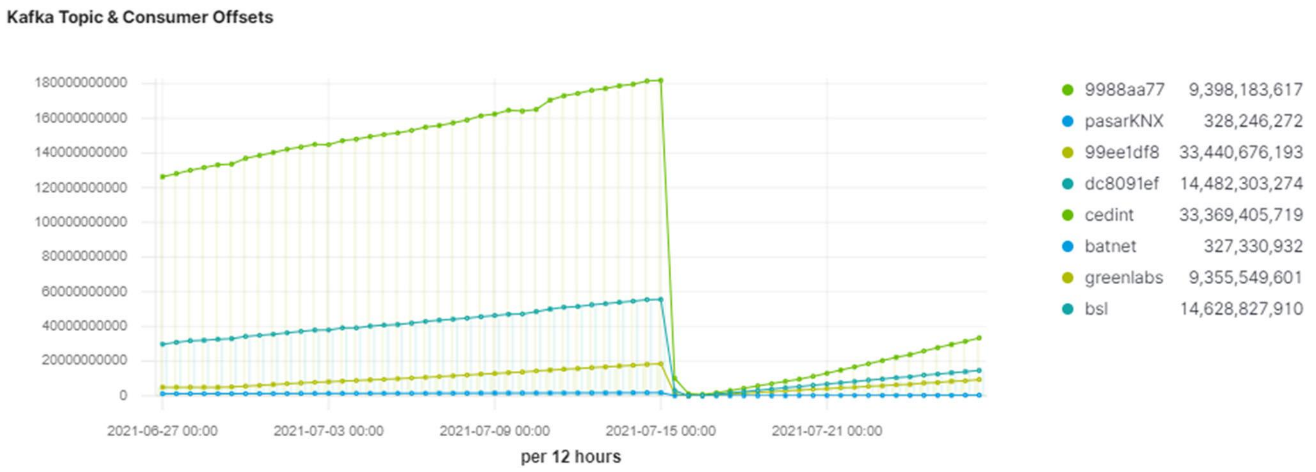


**Fig. 3** Kafka topic offset by use case

one month (27 Jun-27 Jul 2021). We have subjected the platform to different outages and variable workloads (number of IoT devices) to detect potential bottlenecks and check if the resources are well sized. Additionally, to evaluate the platform's response to power outages (*Heartbeat: ICMP, TCP, and HTTP*), we have shotdown the entire platform in the middle of the month (15th Jul 2021). Regarding the offset of Kafka Consumer, for the outage of the 15th July is configured with a "earliest" policy, starting to consume the latest messages and discarding the ones stored in buffer. As for the other three outages (1st, 8th and 23rd July), they are configured with a "latest" policy, starting to read the previous non-consumed messages (see Fig. 3).

### 5.2.1 Edge Node

1. **Smart Building**: during the evaluation period, we have modified the workload once per week, varying the number of devices in the IoT network. Figure 4 shows Kibana dashboard providing information from metrics for the performance of the Smart Building Edge Node. (detailed information can be seen in Fig. 6).
   This use case is the one with the largest number of IoT devices, therefore generates the most inbound (13.9KB/s), and outbound (48.7KB/s) traffic (*Metricbeat system: Network metrics for interfaces*). It should be noted, that the CPU Load (*Metricbeat system: CPU*

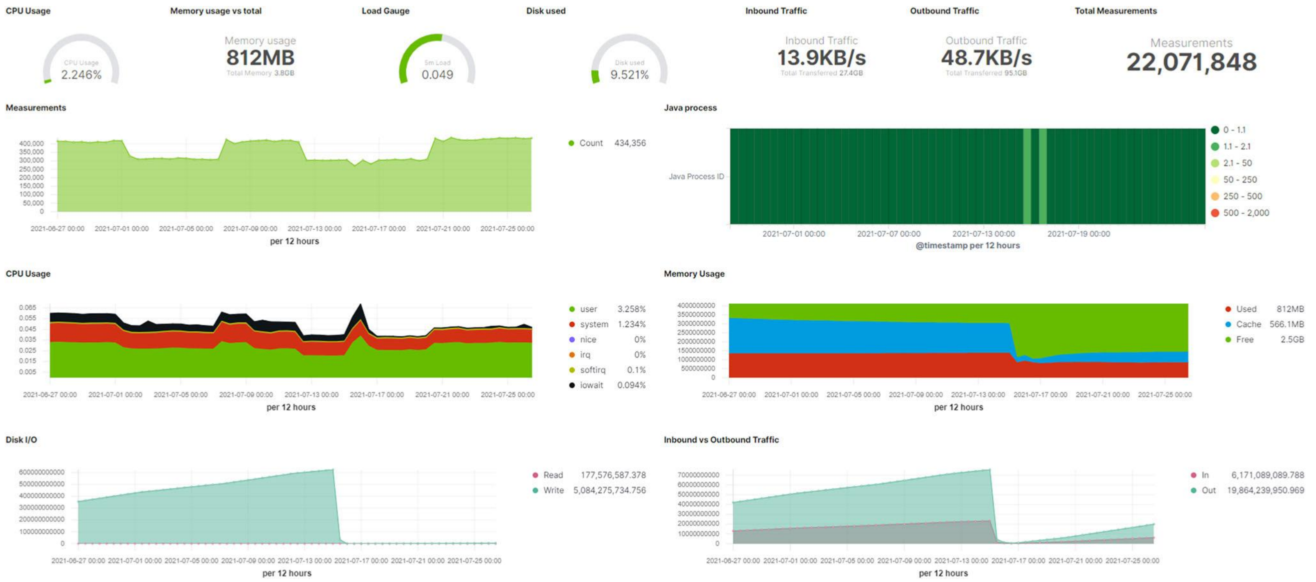
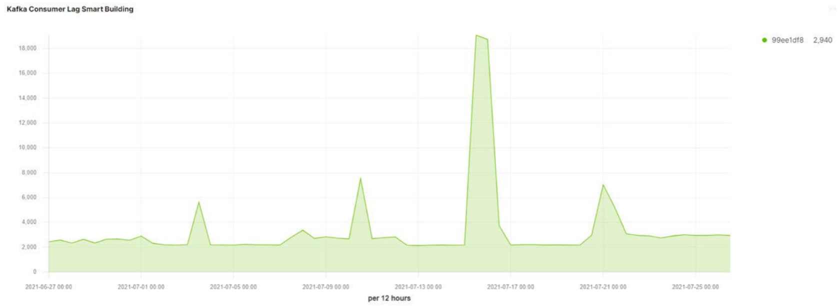


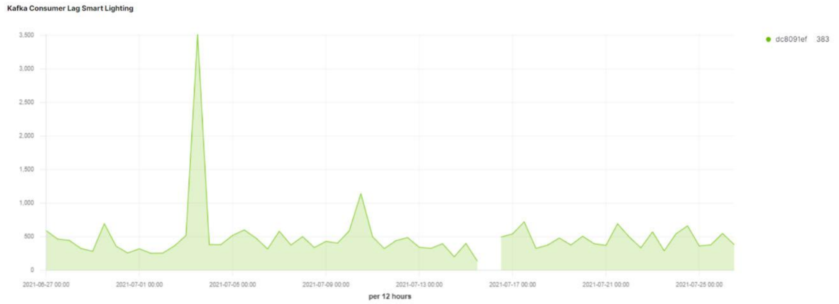
**Fig. 4** Example of the Smart Building Edge Node dashboard

*statistics, CPU load data, and Usage statistics for each CPU core; Jolokia: SystemCpuLoad*) and the System Load (*Jolokia: SystemLoadAverage*) vary according to the number of measurements. The number of simultaneous Java processes (*Metricbeat system: Process execution data and General process summary; Jolokia: ThreadCount*) is when the platform is shutdown (15th July) and the Edge Node is restarted (16th July). The coinciding processes are both BatNM, because there original process does not have time to end properly before the platform restart. Regarding memory consumption (*Metricbeat system: Memory statistics; Jolokia: Heap-*

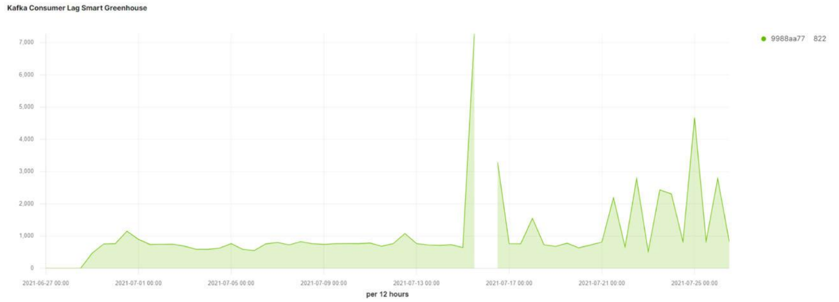
**Fig. 5** Apache Kafka Consumers Lag by use case



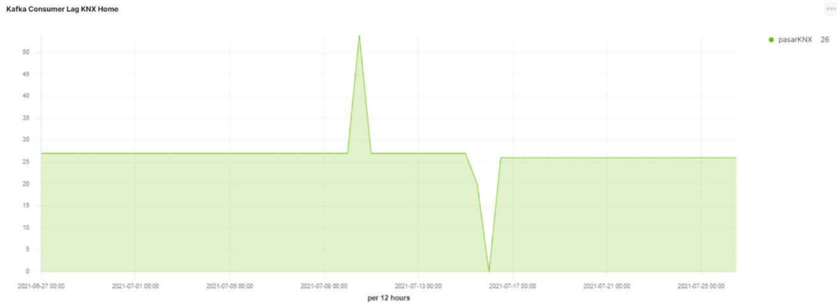
(a) Smart Building



(b) Smart Light



(c) Smart Greenhouse



(d) Smart Home

*MemoryUsage and Non-HeapMemoryUsage*), it can be seen that up to the 15th, it is approximately twice as high as in the rest of the month, especially in the cache memory. As the buffer from the Kafka Consumer is emptied, the number of disk I/O (*Metricbeat system:Disk IO metrics*) is reduced.

2. **Smart Lighting**: the number of devices has remained stable, except for the interruption in the service in BatNM for 24 hours between the 15th and the 16th. Figure 7 shows the detailed statistics of the use case.
   During this month, outgoing traffic (79.2GB) was three times higher than incoming traffic (27GB). The number of Java processes is 1 for the whole month, because during the 24h interruption there is enough time to finalize the BatNM process before the platform restart. The memory consumption remains stable and the cache memory consumes only 700MB.
3. **Smart Greenhouse**: this use case is the one that has undergone the largest variation in the number of IoT. Initially, it only has two devices, and from this day onwards, the remaining devices were added. During the last week, there were abrupt variations in the number of devices, as can be seen in Fig. 8.
   During the outage on 15th, as the IoT platform is restarted and the IoT device discovering process was reinitialized. Hence, there was a significant increase in CPU consumption, reaching the 100% of the two cores. In addition, the number of Java processes and RAM consumption increased sharply. This is caused because the discovering processes are not well configured and do not end properly.
4. **Smart Home**: it is the IoT network with the fewest number of devices. Figure 9 illustrates the performance of the Smart Home Edge Node. Most of the IoT devices are added after the 17th of July. Note that, in contrast with the other uses cases that present a higher number of IoT devices, inbound and outbound traffic on the Edge Node is very similar, because Beat agents generate a similar amount of data as the IoT network itself.

### 5.2.2 Data Streaming

In the Data Streaming module, we have analysed the delay of the Kafka Consumer group by topic (*Metricbeat Kafka: Partition and Consumer*). This lag represents the number of measurements stored in the input buffer, waiting to be sent to the Cloud Server. That is, if this value is 0, it means that the measurement sent by the Edge Node to the Cloud Server is processed immediately.

We can observe the direct relationship between the number of measurements stored in Elasticsearch and the delay of the Kafka Consumer (see Fig. 5): the Smart Building use case stores the most measurements and has the highest delay in Kafka. If the IoT devices are added gradually, the delay increases correspondingly, see Smart Lighting use case. However, if they are added abruptly, see Smart Greenhouse use case, the delay increases independently of the number of IoT devices. In the Smart Home use case, as the amount of IoT devices in considerably low, the addition of new IoT devices does not translate into a lag increase.

### 5.2.3 Cloud Server

The Cloud Server has been analysed and the metrics details are shown in Fig. 10.

The CPU consumption is higher than in the Edge Node, because Spatia receives the measurements from all the use cases (IoT networks) and has to process and send them to Elasticsearch. In addition, the CPU consumption graph shows 4 peaks that correspond with the outages. The difference in intensity between the peaks corresponds to Spatia's Kafka Consumer offset configuration.

The difference between inbound and outbound traffic (*Packetbeat: ICMP, HTTP,and TLS*) remains constant over time, except for the platform blackout on the 15th. Memory consumption increases to almost 100%, which corresponds to the increase in writing operations to disk. This is because Linux uses RAM to speed up disk operations by using memory for buffering and caching. In this way, Spatia's Kafka Consumer processes measurements faster.

### 5.2.4 Elasticsearch Cluster

In the Elasticsearch Cluster module, we have analysed three main aspects: a) the characteristics of the indexing of documents; b) the performance of each node according to its function in the cluster; c) the overall performance of the cluster. Cluster and node performance have been previously presented in (Calderon et al., 2021). Table 2 summarizes the overall of the cluster, where it can be observer that we achieve a balance between consistency and performance by configuring replication factor and shards of Elasticsearch index.

As aforementioned, in Elasticsearch, although the number of primary shards in an index is set when the index is created, the number of replica shards could be changed at any

**Table 2** Cluster and node performance

| | |
|---|---|
| Indexing Latency | 0.68 ms |
| Indexing Rate | 327.12 /s |
| Search Latency | 6.43 ms |
| Client Response Time | 2452 ms |

**Table 3** Number of segments for primary and replica shards

| | Smart Building | Smart Lighting | Smart Greenhouse | Smart Home |
|---|---|---|---|---|
| Primary shards | 23 | 17 | 13 | 10 |
| Replica shards | 22 | 15 | 16 | 10 |

**Table 5** Number of documents being indexed for primary and replica shards

| | Smart Building | Smart Lighting | Smart Greenhouse | Smart Home |
|---|---|---|---|---|
| Primary shards | 11,51 /s | 4,33 /s | 2.3 /s KB | 0.36 /s |
| Replica shards | 11,51 /s | 4,32 /s | 2,3 /s | 0.36 /s |

time. It is an important property because the shard size and the number of primary shards affects the performance. If the shard is too large, Elasticsearch takes more time to reallocate it in the cluster. However, if many shards are implemented, its maintenance is more complex. Each shard contains segments (inverted index), represented in Table 3 (*Metricbeat Elasticsearch: Ccr, Cluster_stats, Index, and Shard*). As data is written to a shard, it is published to new immutable Lucene segments on disk. Therefore, a search on the the search engine (Elasticsearch), looks into the different shards and segments. To reduce disk access time, Lucene caches disk I/O in memory to manage segments. Table 4 (*Metricbeat Elasticsearch: Node and Node_stats*) represents the number of bytes used for this purpose. The number of segments and memory heap vary accordingly to the number of IoT devices and, therefore, measurements.

Elasticsearch is schema-free, which means that documents are indexed without defining previously a structure for fields. When the indexation starts, Elasticsearch automatically detects and adds the mapping structure that best suits to the data type of the field. It is possible to index the same field in different data types. The number of indexed documents for the primary shard and the replica shard increases with the incoming data, as can be seen in Table 5.

## 6 Conclusion

In this work, we present the implementation and performance evaluation of an IoT platform that is novel in the combined use of Elastic Stack and Apache Kafka for managing and monitoring IoT networks. The IoT platform stores the measurements of the IoT devices, integrates visualization tools, allows user to manage and act on their devices, collects metrics from servers and services, centralizes log data, checks the status of the modules, and captures the network traffic.

**Table 4** Total heap memory used by Lucene

| | Smart Building | Smart Lighting | Smart Greenhouse | Smart Home |
|---|---|---|---|---|
| Lucene total | 67,1 KB | 49.5 KB | 41,4 KB | 28,5 KB |
| Terms | 32,3 KB | 23,0 KB | 20,8 KB | 14,4 KB |

We have configured the Elasticsearch Cluster to develop a distributed and replicated architecture that guarantees the availability and consistency of the stored data. On top of that, we have developed a monitoring framework based on Beats agents to facilitate the supervision of the modules operation by the platform administrator. On the other hand, we have customized the functionalities offered by the visualization tool Kibana to display the data of the platform users, IoT devices dashboards, network maps, and the data collected by Beats agents. As a result, management and maintenance tasks are much more friendly and accessible. We have tested the platform's performance in four real use cases: Smart Building, Smart Lighting, Smart Greenhouse and Smart Home. We have evaluated the load levels of each module to detect the factors that most affect the performance of the platform. In the Edge Node, the CPU load and the system load vary depending on the number of measurements.

Additionally, the number of concurrent Java processes causes an increment in the memory consumption due to the increase in the number of disk I/O. The Kafka Consumer lag directly affects to Spatia CPU consumption because Linux uses RAM to speed up disk operations by using memory for buffering and caching. To reduce this impact, we are working to apply machine learning techniques in the Edge Node so it can discard the less relevant measurements, and reduce the lag in the Kafka Consumer.

Onthe other hand, we have verified that the number of segments in a shard, memory heap usage, and index rate in Elasticsearch are proportional to the number of measurements.

In order to achieve better performance results of the IoT platform, we are working on different strategies to improve the I/O accessing times, such as adjusting the configuration of the memory heap from Elasticsearch, varying the shards size, or implementing a Hot-Warm-Cold architecture to manage the lifecycle of the indexes.

## Appendix A Performance results

This appendix contains dashboards of the Edge Nodes and the Cloud Server performance metrics for one month.

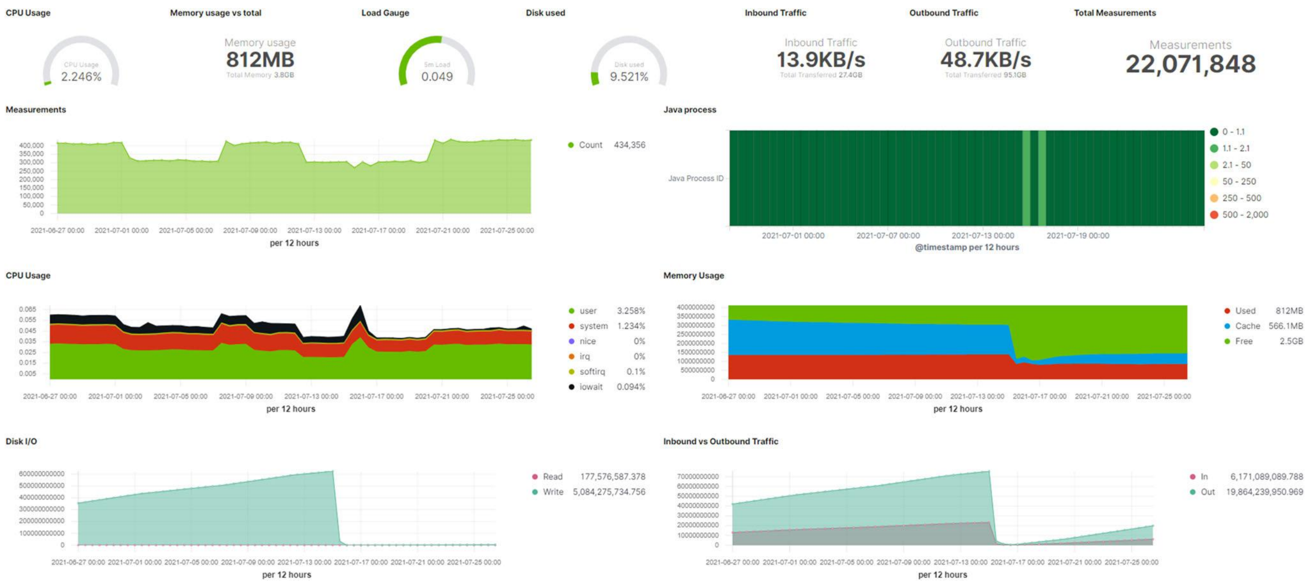


**Fig. 6** Smart Building Edge Node performance

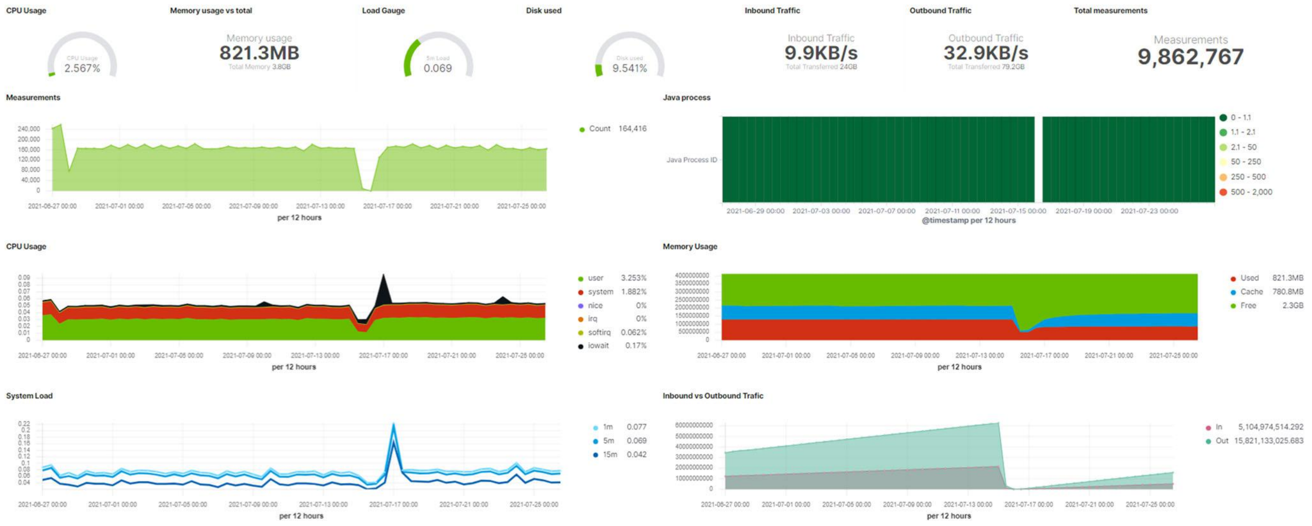


**Fig. 7** Smart Lighting Edge Node performance

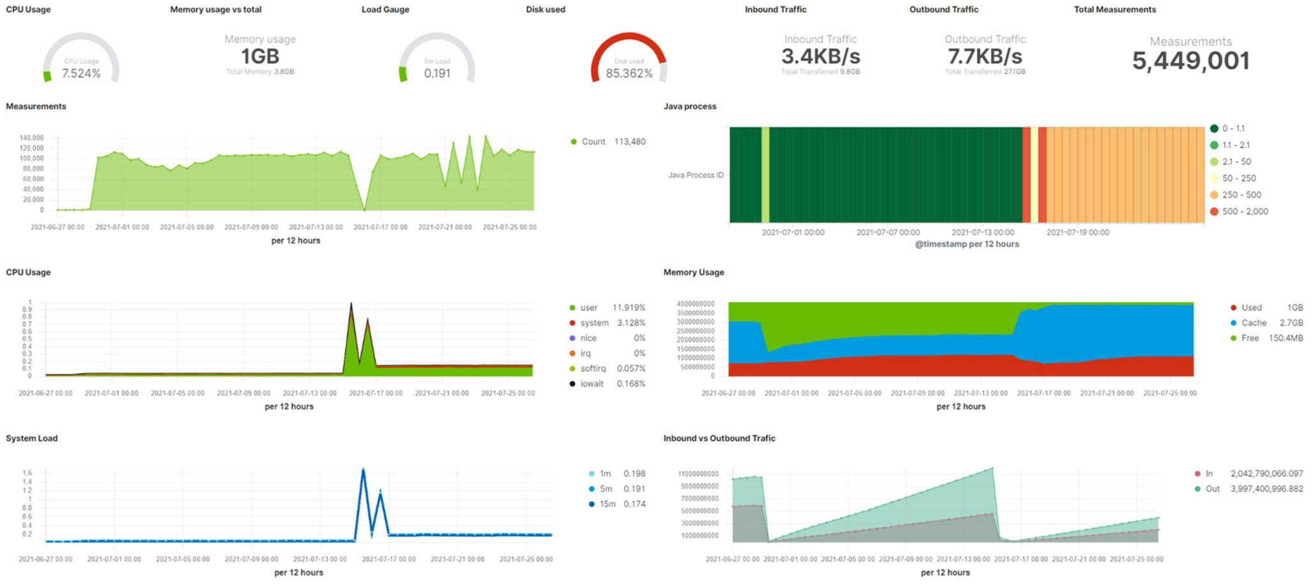


**Fig. 8** Smart Greenhouse Edge Node performance

**Fig. 9** Smart Home Edge Node performance

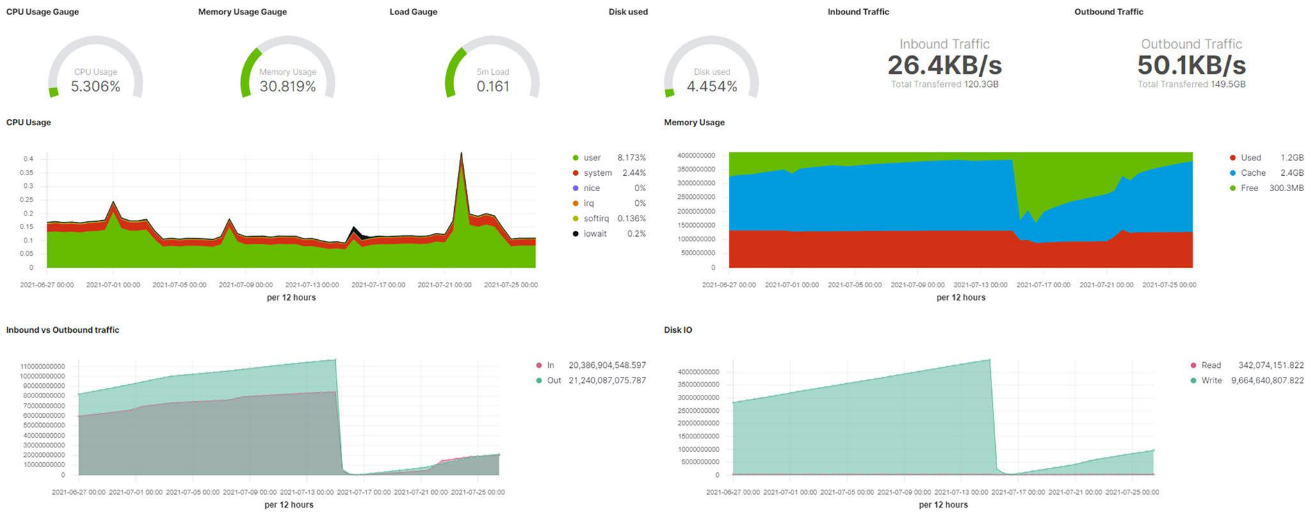


**Fig. 10** Spatia performance

**Acknowledgements** This work is part of the CHIST-ERA research project "ABIDI: Context-aware and Veracious Big Data Analytics for Industrial IoT" and was funded by the State Research Agency (AEI) from the Ministerio de Ciencia e Innovación (MICINN) of Spain, grant number PCI2019-103762.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

**Data Availability Statements** All the data were obtained from four IoT networks (Smart Building, Smart Lighting, Smart Greenhouse and Smart Home) deployed at the Centro de Domótica Integral de la Universidad Politécnica de Madrid (CeDInt-UPM).

## Declarations

**Conflict of Interest** The authors declare no conflict of interest.

## References

Adhikaree, A., Kim, T., Vagdoda, J., Ochoa, A., Hernandez, P. J., & Lee, Y. (2017). Cloud-based battery condition monitoring platform for large-scale lithium-ion battery energy storage systems using internet-of-things (IoT). In *IEEE Energy Conversion Congress and Exposition (ECCE)* (pp. 1004–1009). https://doi.org/10.1109/ECCE.2017.8095896

Ahmed, F., Jahangir, U., Rahim, H., Ali, K.,& et al. (2020). Centralized log management using elasticsearch, logstash and kibana. In *IEEE International Conference on Information Science and Communication Technology (ICISCT). IEEE, Karachi, Pakistan*, (pp. 1–7)

Anusas-Amornkul, T., & Sangrat, S. (2017). Linux server monitoring and self-healing system using nagios. In *International Conference on Mobile Web and Information Systems* (pp. 290–302). https://doi.org/10.1007/978-3-319-65515-4_24

Babun, L., Denney, K., Celik, Z. B., McDaniel, P., & Uluagac, A. S. (2021). A survey on IoT platforms: Communication, security, and privacy perspectives. *Computer Networks*, 192: 108,040

Bajer, M. (2017). Building an IoT Data Hub with Elasticsearch, Logstash and Kibana. In *IEEE 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)* (pp. 63–68). https://doi.org/10.1109/FiCloudW.2017.101

Barth, W. (2008). *Nagios: System and network monitoring*. No Starch Press

Borgia, E. (2014). The internet of things vision: Key features, applications and open issues. *Computer Communications*, *54*, 1–31. https://doi.org/10.1016/j.comcom.2014.09.008

Calderon, G., Del Campo, G., Saavedra, E., & Santamaria, A. (2021). Management and monitoring IoT networks through an elastic stack-based platform. In *IEEE 2021 8th International Conference on Future Internet of Things and Cloud (FiCloud)* (pp. 184–191). https://doi.org/10.1109/FiCloud49777.2021.00034

Chakraborty, M., & Kundan, A. P. (2021). Grafana. In *Monitoring Cloud-Native Applications* (pp. 187–240). Springer

Cravero, A. (2018). Big data architectures and the internet of things: A systematic mapping study. *IEEE Latin America Transactions*, *16*(4), 1219–1226.

Darwish, A., Hassanien, A. E., Elhoseny, M., Sangaiah, A. K., & Muhammad, K. (2019). The impact of the hybrid platform of internet of things and cloud computing on healthcare systems: opportunities, challenges, and open problems. *Journal of Ambient Intelligence and Humanized Computing*, *10*(10), 4151–4166. https://doi.org/10.1007/s12652-017-0659-1

Del Campo, G., Montoya, E., Martín, J., Gómez, I., & Santamaría, A. (2012). Batnet: a 6lowpan-based sensors and actuators network. In *International Conference on Ubiquitous Computing and Ambient*

*Intelligence* (pp. 58–65). Springer. https://doi.org/10.1007/978-3-642-35377-2_8

Dharur, S., & Swaminathan, K. (2018). Efficient Surveillance and Monitoring using the ELK Stack for IoT powered Smart Buildings. In *IEEE 2nd International Conference on Inventive Systems and Control (ICISC)* (pp. 700–705). https://doi.org/10.1109/ICISC.2018.8398888

Espinoza, H., Kling, G., McGroarty, F., O'Mahony, M., & Ziouvelou, X. (2020). Estimating the Impact of the Internet of Things on Productivity in Europe. *Heliyon*, 6(5). https://doi.org/10.1016/j.heliyon.2020.e03935

Gormley, C., & Tong, Z. (2015). *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. O'Reilly Media, Inc

Gos, K., & Zabierowski, W. (2020). The comparison of microservice and monolithic architecture. In *2020 IEEE XVIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)*. IEEE, Lviv, Ukraine, pp. 150–153

Gülcü, C. (2003). *The complete log4j manual*. QOS.ch

Guth, J., Breitenbücher, U., Falkenthal, M., Fremantle, P., Kopp, O., Leymann, F., & Reinfurt, L. (2018). A detailed analysis of IoT platform architectures: concepts, similarities, and differences. In *Internet of Everything. IEEE, Paris, France*, (pp. 81–101). Springer. https://doi.org/10.1007/978-981-10-5861-5_4

Guth, J., Breitenbücher, U., Falkenthal, M., Leymann, F., & Reinfurt, L. (2016). Comparison of IoT platform architectures: A field study based on a reference architecture. In *IEEE 2016 Cloudification of the Internet of Things (CIoT)* (pp. 1–6). https://doi.org/10.1109/CIOT.2016.7872918

Hunt, P., Konar, M., Junqueira, F. P., & Reed, B. (2010). Zookeeper: Wait-free coordination for internet-scale systems. In *USENIX annual technical conference* (vol. 8)

Jaya, N. I., & Hossain, M. F. (2018). A prototype air flow control system for home automation using mqtt over websocket in aws IoT core. In *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* (pp. 111–1116). https://doi.org/10.1109/CyberC.2018.00032

Kang, D., Park, M., Kim, H., Kim, D., Kim, S., Son, H., & Lee, S. (2017). Room temperature control and fire alarm/suppression IoT service using mqtt on aws. In *International Conference on Platform Technology and Service (PlatCon)* (pp. 1–5). https://doi.org/10.1109/PlatCon.2017.7883724

Kazmi, A., Jan, Z., Zappa, A., & Serrano, M. (2017). Overcoming the Heterogeneity in the Internet of Things for Smart Cities. In *International workshop on interoperability and open-source solutions* (pp. 20–35). https://doi.org/10.1007/978-3-319-56877-5_2

Kotenko, I., Kuleshov, A., & Ushakov, I. (2017). Aggregation of Elastic Stack Instruments for Collecting, Storing and Processing of Security Information and Events. In *IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (pp. 1–8). https://doi.org/10.1109/UIC-ATC.2017.8397627

Kreps, J., Narkhede, N., Rao, J., & et al. (2011). Kafka: A distributed messaging system for log processing. In: The 6th International Workshop on Networking Meets Databases. NetDB 2011, Athens, Greece, (vol. 11, pp. 1–7)

Langels, H.-J. (2008). Knx ip using ip networks as knx medium. In *Proceedings of the KNX Scientific Conference 2008. Siemens AG, Regensburg*

Lee, H., Mun, H., & Lee, Y. (2020). Comparing response time of home IoT devices with or withoutcloud. In *2020 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1–6). https://doi.org/10.1109/ICCE46568.2020.9043102

López-Peña, M. A., Díaz, J., Pérez, J. E., & Humanes, H. (2020). Devops for IoT systems: Fast and continuous monitoring feedback of system availability. *IEEE Internet of Things Journal*, *7*(10), 10695–10707. https://doi.org/10.1109/JIOT.2020.3012763

Nasajpour, M., Pouriyeh, S., Parizi, R. M., Dorodchi, M., Valero, M., & Arabnia, H. R. (2020). Internet of Things for Current COVID-19 and Future Pandemics: An Exploratory Study. *Journal of healthcare informatics research* (pp. 1–40). https://doi.org/10.1016/j.dsx.2020.04.041

Pasha, S. (2016). Thingspeak based sensing and monitoring system for IoT with matlab analysis. *International Journal of New Technology and Research*, *2*(6): 19–23

Patti, E. & Acquaviva, A. (2016). IoT platform for smart cities: Requirements and implementation case studies. In *IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI). IEEE, Bologna, Italy*, (pp. 1–6)

Peddoju, S. K. & Upadhyay, H. (2020). Evaluation of IoT data visualization tools and techniques. *Data visualization: Trends and challenges toward multidisciplinary perception. Springer, Singapore*, (pp. 115–139)

Pelle, I., Czentye, J., Dóka, J., & Sonkoly, B. (2019). Towards Latency Sensitive Cloud Native Applications: A Performance Study on AWS. In *IEEE 12th International Conference on Cloud Computing (CLOUD)* (pp. 272–280). https://doi.org/10.1109/CLOUD.2019.00054

Renita, J. & Elizabeth, N. E. (2017). Network's server monitoring and analysis using nagios. In *2017 IEEE International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). IEEE, Chennai, India*, (pp. 1904–1909)

Sahay, M. R., Sukumaran, M. K., Amarnath, S., & Palani, T. N. D. (2019). Environmental monitoring system using IoT and cloud service at real-time. *EasyChair Preprint*, *968*, pp. 1–8

Shelby, Z., Hartke, K., & Bormann, C. (2014). The constrained application protocol (coap). Internet Engineering Task Force (IETF), https://datatracker.ietf.org/doc/html/rfc7252

Singh, R. P., Javaid, M., Haleem, A., & Suman, R. (2020). Internet of Things (IoT) Applications to Fight Against COVID-19 Pandemic. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, *14*(4), 521–524. https://doi.org/10.1016/j.dsx.2020.04.041

Tader, P. (2010). *Server monitoring with zabbix. Linux Journal*, *2010*(195), 7.

Talaş, A., Pop, F., & Neagu, G. (2017). Elastic Stack in Action for Smart Cities: Making Sense of Big Data. In *13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)* (pp. 469–476). https://doi.org/10.1109/ICCP.2017.8117049

Vaarandi, R. & Pihelgas, M. (2014). Using security logs for collecting and reporting technical security metrics. In *IEEE Military communications conference. IEEE, Baltimore, MD, USA*, (pp. 294–299)

Venkatramulu, S., Phridviraj, M., Srinivas, C., & Rao, V. C. S. (2021). Implementation of grafana as open source visualization and query processing platform for data scientists and researchers. *Materials Today: Proceedings*. https://doi.org/10.1016/j.matpr.2021.03.364

Zúñiga-Prieto, M., Rodríguez, D., Rodríguez, J., Solano, L., Insfran, E., & Abrahão, S. (2018). IoT-adl: An adl for describing cloud IoT applications. *Journal of Computers*, *29*(6), 264–273

**Gonzalo Calderon** received the Bachelor of Engineering in Telecommunication Technologies and Services in 2017 (accredited by ABET) and the Master of Science in Telematic Services and Network Engineering in 2020, both from the Universidad Politécnica de Madrid (UPM), Madrid, Spain. He has been a researcher of the Centro de Domótica Integral (CeDInt-UPM) since 2016 at the Energy Efficiency and Internet of Things group, where he has been involved in different international projects. His research line is related to Internet of Things, databases, big data, and edge computing.

**Guillermo del Campo** received his PhD from UPM in 2017 in the Visible Light Communications field and is the head of the Energy Efficiency and IoT Research Group at CEDINT-UPM. He has participated in more than 25 R&D projects on Energy Efficiency and Smart Environments since 2009. He is author of more than 30 scientific journals and conference papers.

**Edgar Saavedra** achieved his Master in Telecommunication Engineering in 2018 at UPM. With his Master Thesis, he earned the Best Master Thesis Award by the Spanish Official Telecommunication School in 2019. He is currently (2023) a PhD. student at CEDINT-UPM in the field of Sensor Wireless Networks for Energy Efficiency in industry, homes and cities.

**Prof. Asunción Santamaría** is PhD in Telecommunications Engineering and professor at the UPM Telecommunications School since 1994. She is author or co-author of more than 30 scientific papers and books and more than 50 conference papers on wireless networks and ICT-based systems for energy efficiency. From 2005 to 2017, she has been the director of CeDInt-UPM, creating the Virtual Reality and IoT-Energy research groups.

# Alexa-based voice assistant for smart home applications

Clara Jiménez, Edgar Saavedra, Guillermo del Campo,
and Asunción Santamaría

©SHUTTERSTOCK.COM/KARSTEN NEGLIA

Today, the Internet of Things (IoT) is becoming an essential player in creating a new smart era. The communication through the Internet to IoT devices enables new applications in multiple environments, including smart buildings and cities. The IoT market is projected to grow to 75.4 billion connected devices by 2020. Within the IoT ecosystem, smart home technology is also improving at an exponential rate.

Houses are featuring more intelligent technology that automates everyday activities. Recent advances in human–device interaction (HDI) techniques simplify the ways people can manage IoT devices. Therefore, the voice interaction technique is currently at its peak, leading to a natural and straightforward coexistence of human and IoT devices, primarily in smart home solutions but also gaining importance in several smart city domains.

A virtual assistant is a software agent that grants tasks performance automation by making use of natural HDI. In the case of voice virtual assistants, users communicate their requests using their voice, and the virtual assistant processes the

*Digital Object Identifier 10.1109/MPOT.2020.3002526*
*Date of current version: 6 July 2021*

0278-6648/21©2021IEEE

> Worldwide, thousands of languages, dialects, and even accents need to be present in the systems' databases to train a reliable AI speech recognition model.

information received and answers in the same way. Voice assistants allow users to carry out multiple tasks, such as playing music from external services (e.g., Spotify or Apple Music); setting alarms or timers; or reporting the latest news, weather conditions, or real-time traffic.

Artificial intelligence (AI)-driven voice assistants, such as Amazon's Alexa, Google Assistant, Microsoft's Cortana, and Apple's Siri, have managed to define the current way in which people interact with multiple environments. According to voice assistant technology reports released in 2018, 27% of the global online population is using voice search on mobile devices, and 17% currently own a voice-controlled smart assistant.

Smart voice assistants have a relatively long history, although it was in recent years that their popularity increased enormously. One of the first voice assistants was Siri, introduced in 2010 as an iPhone application. Today, it is the assistant for the whole Apple ecosystem. The next big launch in voice assistants was Cortana, which was unveiled in 2014 as a part of Microsoft operating systems and is now available as an application. Also during 2014, Alexa was released, becoming the most-used voice assistant and capturing about 70% of the Western market share of smart speakers.

The last to appear was Google Assistant, which was launched in 2016. It is integrated with every Android phone, and it can be used in iPhones with the designated app, filling around 20% of the smart speaker share in the Western market. Every assistant has its own waking word, which are, respectively, "Hey Siri," "Hey Cortana," "Alexa," and "OK Google." Of course, they all are integrated with smart speakers (either proprietary or third party), which happens to be the most common use for voice assistants today.

Regarding speech recognition, all of these voice assistants need to improve their systems, which requires interaction time and user data. Worldwide, thousands of languages, dialects, and even accents need to be present in the systems' databases to train a reliable AI speech recognition model. Furthermore, voice assistants are able to adapt to users' voices over time, reducing any misunderstanding related to an individual's particular characteristics, such as tone or speaking rate.

Current voice assistants are able to understand different English accents with an 85% overall accuracy (and around 5% less for nonnative English speakers). On the other hand, users want a voice assistant that not only comprehends them but also speaks to them understandably; Amazon Alexa currently supports eight different languages (15 dialects). Global expansion of voice assistants will enhance the diversity of data for AI model training, ensuring proper speech recognition for every language, dialect, or accent. Technological advances in AI and market penetration should overcome any issues to avoid leaving people behind.

## Voice-assisted living in smart homes

Thanks to the recent technological improvements and the rise of user adoption, voice assistants seem the most suitable solution for handling required actions in smart homes, thus making available a natural HDI that allows a cohesive user experience across the smart home ecosystem. Nevertheless, it is predicted that voice assistants will be the cornerstone of smart home development, experiencing a 1,000% growth (from 25 million voice assistant devices used to control smart homes in 2018 to 275 million in 2023) and becoming the primary mode of interaction with hardware and software in the near future.

In this context, users would be able to verbally ask about different ambient parameters, such as the room temperature or energy consumption in a certain period of time, and even control devices like a TV or different home systems, such as lighting or heating, ventilation, and air-conditioning (see Fig. 1). However, this application turns out to be straightforward only in the case of using assistant-compatible devices. As a result, to connect third-party devices (e.g., appliances or legacy systems), it is necessary to configure both the virtual assistant and the IoT smart home platform. This will allow IoT devices to be made compatible with voice assistants to some extent, and it represents a new business opportunity for developers.

In general, the compatibility between an IoT system and a voice virtual assistant is ensured by making use of client–server architectures, such as Representational State Transfer (REST). This software architectural style allows the creation of web services that expose web resources identified by different web addresses. Thus, the server makes resources available for clients, who are able to access them using multiple techniques, e.g., HTTP methods.

In this context, while the IoT system is in charge of server responsibilities, the client side is associated with the virtual assistant. Regarding the client side, the development of new applications for every virtual assistant should be made available. Furthermore, tasks related to authentication and authorization processes need to be performed to access protected data from the IoT resources owned by every different user.

## "Alexa, who are you?"

Alexa is a voice virtual assistant, developed by Amazon, that is available on multiple Amazon devices, such as the Echo smart speakers and Fire TV, as well as devices from third-party manufacturers (e.g., Sonos smart speakers; headphones from Jabra or Bose; thermostats from Ecobee; and in-car products, such as Roav VIVA developed by Anker). According to recent surveys, this virtual assistant is broadly used to play music or the radio and provide real-time information, such as the weather, traffic, or news. These frequently used features are followed by setting alarms or timers, making to-do lists, and playing audio games. However, a growing percentage of users are using Alexa as a home automation system.

That being said, how does Alexa understand users' requests to respond correctly? For a smart speaker, such as Amazon Echo, to be capable of comprehending what users say and performing the proper tasks, AI services are needed. How this actually works is depicted in Fig. 2.

These AI services allow for the creation of a natural and enriched conversational experience by fulfilling different tasks: from recognizing person's speech to generating models fitting humans' communication methods. This is the responsibility of the Amazon cloud-base service called *Alexa Voice Service* (*AVS*). When a user has finished speaking to the Alexa virtual assistant, a recorded message is sent to AVS. AVS provides automatic speech recognition to turn the audio stream into text strings. In addition, AVS is built with natural language understanding (NLU), which interprets the result in text format and produces a suitable intent by extracting significant data that represent relevant information.

NLU, therefore, is responsible for going from just text to the real meaning of what the user wants to do, i.e., the actual intent. This intent is redirected to the corresponding Alexa built-in feature or skill so that it can be fulfilled. For example, if the user asks, "Alexa, what is the temperature in the living room?", NLU will identify that the user is requesting home-related information, drawing on the corresponding Alexa smart home feature or skill.

**In general, the compatibility between an IoT system and a voice virtual assistant is ensured by making use of client–server architectures, such as Representational State Transfer.**

Alexa skills are capabilities that developers provide to Alexa so that the device is able to perform tasks beyond its base functionalities. Thus, thedevelopment of an Alexa skill has its foundations on dialog management (DM) and response generation (RG). DM focuses on looking for the required information to generate a suitable response considering the context of the conversation. RG aims to form a sentence that makes sense with the proper information to be answered. Finally, AVS provides text-to-speech capabilities, i.e., it generates an audio response from the text answer with a synthetic voice and streams it back to the smart speaker and then the user.

## An Alexa voice-controlled smart home

Users are able to extend the applications of their Alexa device by subscribing to additional functionalities, the Alexa skills. Therefore, developers can create Alexa skills to allow customized interactions for specific purposes. In the case of a smart home skill, to implement a compatibility between Alexa and a third-party IoT solution, it is necessary to describe the suitable instructions in both the IoT platform and the virtual assistant.

Figure 3 presents a typical IoT architecture for a smart home solution, in which, for each house, multiple IoT end devices are connected to the Internet via a network hub

**FIG1** The use of a voice assistant to interact with different systems of a smart home.

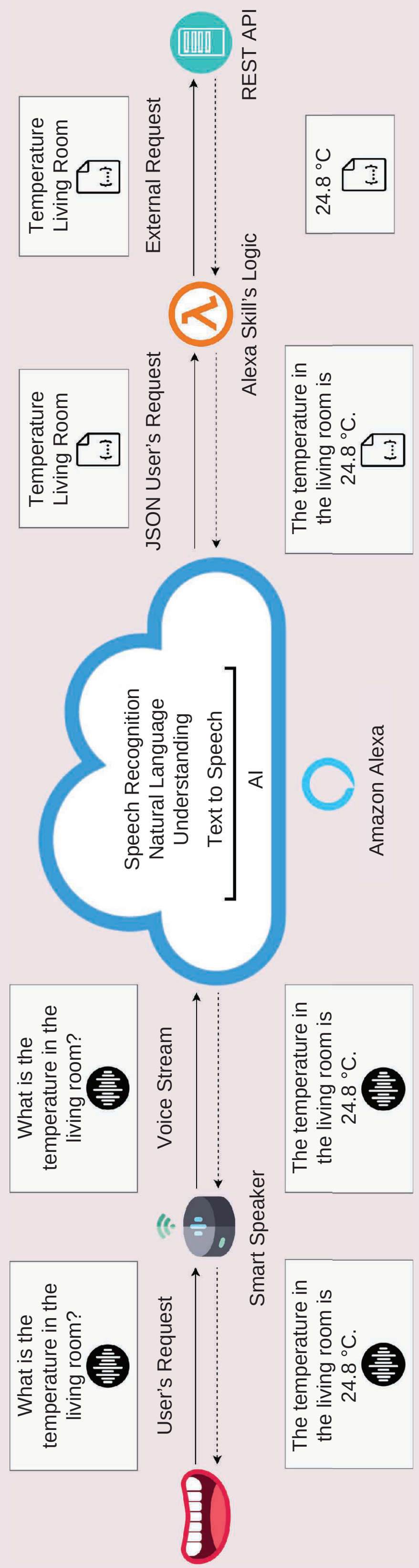


**FIG2** The architecture of the Alexa voice control solution. API: application programming interface; JSON: JavaScript Object Notation.

or gateway. A central server gathers and stores data coming from the different houses. Smart IoT applications and services using different interfaces (e.g., web or mobile) can be integrated atop the central server, as can a voice service, such as Alexa.

Data accessing and actuation services generally have to be described in the central server of the IoT platform. Usually, these services tend to be REST based to fulfill the interoperability required by virtual assistants to respond to HTTP requests, perform the corresponding tasks, and send back the proper response. The development of different applications working on top of the whole system is possible using web services provided by this central server.

For example, a web or mobile application can be developed to control IoT devices from a specific environment. Figure 4 shows a screenshot of a responsive smart home web app, where users, once logged in, can manage their devices—e.g., asking for data, turning appliances on and off, or even configuring the IoT devices. By locating the devices in specific rooms and linking artifacts to the corresponding IoT devices, it would be possible to make voice requests for carrying out specific actions. For example, if a TV artifact is associated with a smart plug installed in the living room, a voice request could be, "Alexa, turn on the living room TV." Likewise, for a smart sensor located in the bathroom, a communication could be, "Alexa, what is the humidity in the bathroom?"

On the other hand, an implementation on the virtual assistant side must be performed. In the case of Alexa, using the Alexa Developer Console, it is possible to create a new skill by describing its voice interaction model and implementing the corresponding logic. This logic can be controlled either by what is known as *Amazon Web Services* lambda functions or an external web service that can handle requests from Alexa.

### Voice interaction model

The first step in developing a skill is defining the voice interaction model.

To interact with Alexa, a user must follow this predetermined model when requesting by voice. Thanks to this model description, the user's voice requests are translated into a JavaScript Object Notation (JSON) format to be properly understood, which is an open standard for data transfer that uses human-readable text to interchange information.

The voice interaction model is formed by four components: invocation name, intents, utterances, and slots. The invocation name is used to access the skill. Regarding the smart home use case, a suitable invocation name could be "My Smart Home" and would be launched with the following command: "Alexa, open My Smart Home," or "Alexa, ask My Smart Home . . .". Intents represent the actions that users are allowed to perform, i.e., functions that the skill can fulfill.

For example, a smart home skill could provide an intent in charge of answering questions about environmental parameters in different areas of the house. Every possible way that a user might invoke this intent is defined by utterances. Examples of how users may ask Alexa to accomplish an intent include "What is the temperature in the room?" and "What is the current humidity in the kitchen?" The more sample utterances are offered in a skill, the more accurate it will be. Moreover, in these utterances, the user may include certain variables that provide relevant information. Thus, utterances of every intent can optionally have arguments called *slots*, which provide recognition for relevant values (e.g., areas of the house, ambient parameters) (see Fig. 5).

All of the intents (user-spoken requests), utterances (different expressions for each intent), and slots (relevant information included in the utterances) make up the skill's voice interaction model, which is described using a JSON format scheme (see Fig. 6).

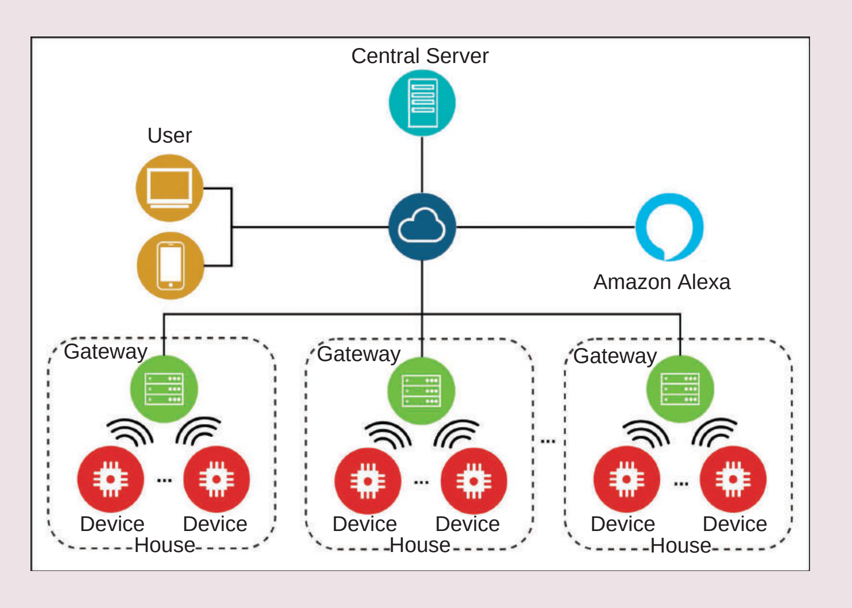


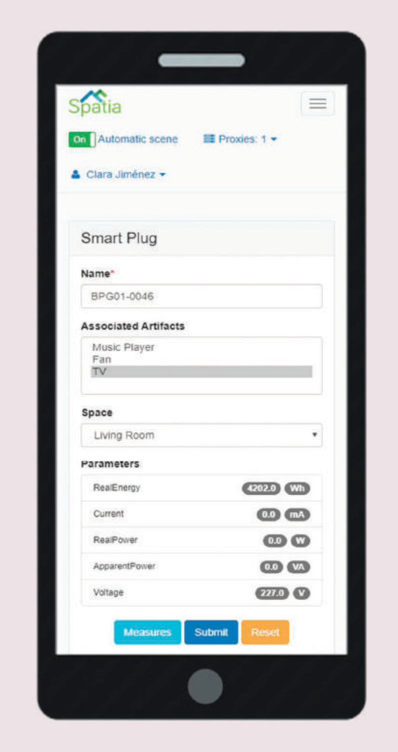
**FIG3** The IoT architecture of a smart home solution.

## *Skill logic and authentication implementation*

Once the voice interaction model is designed, the logic of the skill has to be implemented, for instance, in a lambda function. The Alexa Skills Kit provides self-service application programming interfaces and tools to build Alexa skills, enabling writing of the skill logic with the corresponding software development kits for Node.js, Python, or Java. This logic is in charge of processing the incoming requests that AVS sends after recognizing the user's appeals and responding with a customized and appropriate output. In the skill's logic, functions associated with the different possible intents requested by the user are implemented.

For example, when Alexa interprets the question "What is the temperature in the living room?", the function corresponding to the intent responsible for attending requests about environmental parameters in some area of the house is executed. This function gathers information about the temperature parameter in the living room by making the corresponding request to the IoT platform and properly returns a voice response.

Therefore, in the case of the smart home skill, it will be necessary to access external services, such as REST services exposed by an IoT platform. Using an HTTP client to retrieve information from those external services could be performed in the implemented logic functions. Those HTTP requests would be directed to the central server of the external platform, which is generally responsible for the management of every end device in any IoT solution. Once relevant information is retrieved, the voice assistant is ready to return a proper voice response to the user.



Smart Plug

Name*
BPG01-0046

Associated Artifacts
Music Player
Fan
TV

Space
Living Room

Parameters

| | | |
|---|---|---|
| RealEnergy | 4202.0 | Wh |
| Current | 0.0 | mA |
| RealPower | 0.0 | W |
| ApparentPower | 0.0 | VA |
| Voltage | 227.0 | V |

**FIG4** A screenshot of a responsive smart home web app.

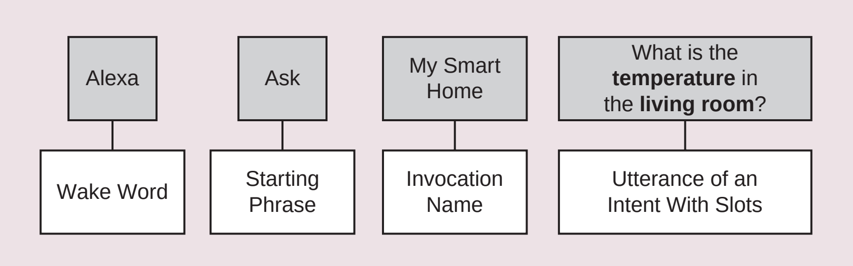


**FIG5** The components of the Alexa voice interaction model.

```
{
    "interactionModel": {
        "languageModel": {
            "invocationName": "my smart home",
            "intents": [
                {
                    "name": "ParameterIntent",
                    "slots": [
                        {
                            "name": "room",
                            "type": "AMAZON.Room"
                        },
                        {
                            "name": "parameter",
                            "type": "ParameterSlot"
                        }
                    ],
                    "samples": [
                        "what is the {parameter} in the {room}",
                        "how is {parameter} in {room}"
                        "tell me about {parameter} in {room}"
                        "{parameter} in the {room}"
                    ]
                }
                ...
            ],
            "types": [
                {
                    "name": "ParameterSlot",
                    "values": [
                        {
                            "name": {
                                "value": "illumination"
                            }
                        },
                        {
                            "name": {
                                "value": "humidity"
                            }
                        },
                        {
                            "name": {
                                "value": "temperature"
                            }
                        }
                    ]
                }
            ]
        }
    }
}
```

**FIG6** A JSON code example for the voice interaction model: The first line refers to the skill's invocation name ("my smart home"). Then, the intents the skill can fulfill are also listed, with the corresponding sample utterances, including the slots and their type.

However, how can the HTTP requests coming from each house be distinguished? That is, how will Alexa know where (i.e., in which house) the devices to which the user is referring are located? To solve this issue, an initial authentication process is necessary. Certain skills require the user registered in Amazon Alexa to be linked with a user belonging to another application or service (e.g., using a specific account for a media services provider, ordering food through delivery services, or booking transportation).

Likewise, smart home skills must connect the identity of the Alexa user with an identity in the smart home service provider's system, resulting in an account linking process. The aim is to generate a link between the Alexa user and the user that is registered in the external smart home system that provides functionality to the skill.

This authentication is performed via a web interface that displays a login page when linking the Alexa account via the Alexa mobile app. Once users provide their credentials in the login page and authenticate with the service, the skill is ready to use, allowing the management of IoT devices owned by the registered account. The external system provides Alexa an access token that is stored and identifies the user in the resource server. Thus, once the user has completed the authentication process in that system through the Alexa app, the JSON with the user's voice requests now includes this token in addition to intent and slot value information.

This form of implementation of linking accounts from Alexa and another system uses RFC6749, the OAuth 2.0 authorization framework. This standard protocol allows for limited access to user accounts via an HTTP service. It delegates the user authentication process to the external service that controls the account and authorizes third-party applications, such as Alexa, to access that user account. Thus, OAuth 2.0 specifies four roles working in the context of Alexa skills: resource

**Alexa skills are capabilities that developers provide to Alexa so that the device is able to perform tasks beyond its base functionalities.**

owner, client, resource server, and authorization server.

The resource owner, in the specific context of Alexa, is the user registered at the external system who can access the restricted data provided by the resource server. The client is the application that needs to access data at the resource server on behalf of the resource owner, i.e., the Alexa skill.

First, the client logs in as the user account owned by the resource owner to authenticate itself at the authorization server to gain access to the resource server. The resource server hosts the resources or services the user wants to access. The authorization server, which can be the same as the resource server, is in charge of authenticating the identity of the resource owner and issues an access token if convenient. This access token is needed to access restricted resources provided by the resource server, so the skill uses it to access information in the external system (see Fig. 7).

### *Visual complementary information*

Finally, to complete the smart home experience, Alexa offers the user a multimodal environment through the voice assistant devices that are equipped with a screen. These devices can complement the voice answers offered with a visual interface. The management of a device through the voice is simple and natural. However, the creation of a multimodal experience combining the vocal interface and a touchscreen; visual answers through text, images, or video; and so on offers the user a new type of management more comfortable and attractive without reducing its simplicity.

The pseudoprogramming language used to describe these visual answers is called *Alexa Presentation Language* (*APL*). It is based in JSON documents that describe the information that has to be shown and how it must be done when defining a graphical user interface.

The APL document of a skill contains design objects named *layouts* that are defined by configurable variables provided by Amazon Alexa. Thus, this document can import existing packages of APL content that have been developed by Amazon Alexa, such as properties, styles, or predefined components, that can be used in the description of the content that is going to be shown. Figure 8 shows an example of the answer for the question "What is the temperature in the living room?" of the Alexa smart home skill using both voice and visual-type responses.

### Give voice to smart cities

When their use out of homes is expanded, voice assistants can open new ways for people to live in and enjoy their cities. Increasingly, cities around the world are gathering enormous amounts of data relative to citizens, pollution, transportation, events, and so on. These data tend to be open, so anyone may take advantage of them. However, many cities provide data that are not exploited yet, either due to a lack of easy ways to use them or the fact that no straightforward utility for them has been discovered.

Voice assistants could try to exploit all of these open data, providing natural interactions with the user and playing a completely new role regarding smart cities. The most

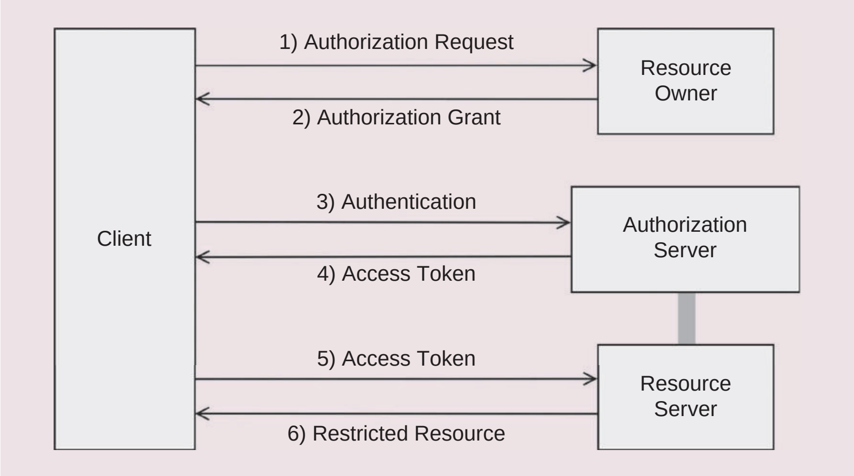
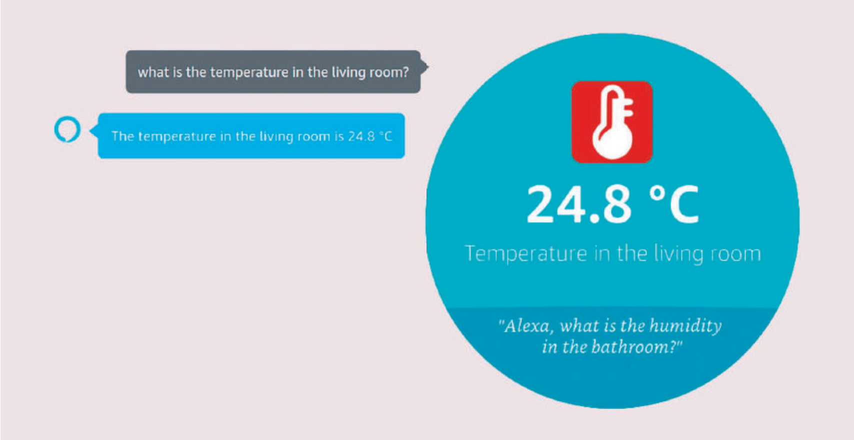


**FIG7** The workflow of the account-linking process using OAuth 2.0.



what is the temperature in the living room?

The temperature in the living room is 24.8 °C

24.8 °C

Temperature in the living room

"Alexa, what is the humidity in the bathroom?"

**FIG8** An example of the voice and visual interface combination. APL allows additional information to be shown, such as an Alexa footer, which guides the conversation with the user by advising him or her on what to ask next.

> **Voice assistants can play a very important role in breaking barriers and helping people with disabilities, especially by easing access to technology.**

straightforward application is a change in how people interact with information, shifting from navigation through many screens to voice conversations. Mobility will be one of the key fields where voice assistants can make a difference. There are large amounts of data about transit, traffic, routes, and so on. Therefore, a voice assistant can be aware of its real-time location.

An example application could be someone walking down the street, heading a bus stop—not sure he or she is on time. The user might ask the assistant, “Am I on time to get my bus?” Then, the assistant itself may recommend taking a new route or speeding up the pace to arrive to the destination as soon as possible. It could even propose other transport alternatives, such as a taxi or car-sharing service.

## Benefits and challenges

Thanks to voice assistants, people can discover a set of benefits never experienced before. Faster, easier ways of doing common daily things might be achieved while reducing human effort.

However, a few challenges come along with this great convenience, such as privacy, security, and accessibility. Privacy is a topic that greatly concerns users. In the beginning, many people used to think that voice assistants were recording their conversations continuously. However, the reality is that they listen to you only after they are invoked with the wake word. Once listening, these recordings are stored, yet only a small number of them are reviewed by a human. Still, companies now allow users to opt out of these studies or delete their recordings.

Regarding security risks, it is just a matter of precaution. For example, a secure Wi-Fi network can be created with the use of a router from a reputable manufacturer or even by configuring a separate network for the home automation system. Other simple methods include updating passwords and keeping device software up to date.

Another important challenge concerns accessibility for people with disabilities or elderly indivduals. Voice assistants can play a very important role in breaking barriers and helping people with disabilities, especially by easing access to technology. However, to enhance the home assisted living experience, the voice assistant should be able to take the initiative, i.e., by not only answering requests but also communicating with users as a result of home and external information.

## Conclusion

Voice assistants have developed a completely new way of communicating with technology and with the world around us. Shifting from sight and touch to voice and ear is creating a completely new paradigm in our connected digital society. A novel way of designing and developing digital tools has come along with it, as have new developers.

Researchers and engineers must take all of these new instruments and assist society in rising technologically. Woefully, ethical consequences or opportunities are often overlooked when a new, disruptive technology comes into play. This time, considering the genuine change voice technologies are creating, engineers must not only directly exploit voice assistants in the legacy, technological manner but also in the way society can be enhanced.

## Read more about it

- “Voice Search: A deep-dive into the consumer uptake of the voice assistant technology,” GlobalWebIndex, Insight Rep., 2018. [Online]. Available: https://www.insidemarketing.it/wp-content/uploads/2019/01/Voice-Search-global-web-indexreport.pdf
- S. Lucero, “IoT platforms: Enabling the Internet of Things,” IHS Technology, Mar. 2016.
- “Smart homes: Strategic opportunities, business models & competitive landscape 2019-2024,” Juniper Research Ltd, 2019.
- “Alexa Skills Kit documentation,” Amazon.com. https://developer.amazon.com/docs/ask-overviews/build-skills-with-the-alexa-skills-kit.html
- Alexa Developers Channel home page. [Online]. Available: https://www.youtube.com/channel/UCbx0SPpWT6yB7_yY_ik7pmg

## About the authors

***Clara Jiménez*** (c.jimenez@upm.es) earned her master’s degree in telecommunications engineering in 2018 from Universidad Politécnica de Madrid (UPM), Madrid, Spain. She currently works at CeDInt-UPM, Madrid, Spain, where she is pursuing a Ph.D. degree in cross-platform-independent services applied to accessibility and noninvasive wellness monitoring.

***Edgar Saavedra*** (e.saavedra@ieee.org) earned his master’s degree in telecommunications engineering in 2018 from Universidad Politécnica de Madrid, Madrid, Spain. He is currently a Ph.D. student at CeDInt-UPM, Madrid, Spain, in the field of sensor wireless networks for energy efficiency in industry, homes, and cities.

***Guillermo del Campo*** (gcampo@cedint.upm.es) earned his Ph.D. degree from Universidad Politécnica de Madrid, Madrid, Spain, in 2017 in the visible light communications field. He is the head of the Energy Efficiency and Smart Environments research group at CeDInt-UPM, Madrid, Spain.

***Asunción Santamaría*** (asun@cedint.upm.es) earned her Ph.D. degree in telecommunications engineering. She has been a professor at the Universidad Politécnica de Madrid Telecommunications School, Madrid, Spain, since 1994.

# Digital Twins for Street Lighting: Challenges for a Virtual Reality solution based on Internet-of-Things Devices and Photometry Rendering

Guillermo del Campo
*CEDINT-UPM*
*Universidad Politécnica de Madrid*
Madrid, Spain
guillermo.delcampo@upm.es

Luca Piovano
*CEDINT-UPM*
*Universidad Politécnica de Madrid*
Madrid, Spain
luca.piovano@upm.es

Francisco Pedro Luque Oostrom
*CEDINT-UPM*
*Universidad Politécnica de Madrid*
Madrid, Spain
fp.luque@upm.es

Edgar Saavedra
*CEDINT-UPM*
*Universidad Politécnica de Madrid*
Madrid, Spain
e.saavedra@upm.es

Georges Zissis
*LAPLACE*
*Université Toulouse III Paul Sabatier*
Toulouse, France
georges.zissis@laplace.univ-tlse.fr

Asunción Santamaría
*CEDINT-UPM*
*Universidad Politécnica de Madrid*
Madrid, Spain
asun.santamaría@upm.es

*Abstract*— **Public lighting systems are an essential infrastructure that allows to shape the functional use of the urban spaces. With the arrival of the Internet of Things (IoT), the street lighting infrastructure is considered the backbone of smart cities services, providing connectivity and access to the power grid. Therefore, improving its management is essential towards digitalization of cities. To this end, a combination of Digital Twin (DT), IoT and Virtual Reality (VR) technologies appears to be an optimal tool to enhance the street lighting management. A Digital Twin differentiates from models and simulations because it enables mutual interaction between the physical and digital world in real-time. Sensors and IoT technologies handle the transfer of information to update the virtual model according to the physical counterpart's live changes and vice-versa. The goal is to provide technicians, stakeholders, and managers with an interactive tool to foster a data-driven understanding of the outdoor lighting system that improves their decision-making. The introduction of VR is particularly promising since it allows a realistic, controlled, and intuitive reproduction of real environments at scale. However, several challenges arise when practically building such DT. Based on CEDINT-UPM experience – which is currently working on a DT solution for monitoring the performance of the outdoor lighting installations at a university campus –, we would like to highlight our experience and the lessons learnt during its implementation. The focus of this work is on the communication aspects with IoT systems and the rendering of realistic lighting effects. To this end, our objective is three-fold: 1) to highlight the main challenges faced towards its development and deployment; 2) to introduce the technical solutions to overcome such challenges; and 3) to outline the desirable features aiming at expanding its functionalities.**

***Keywords— Smart Lighting, Street Lighting, Virtual Reality, Digital Twin, Internet of Things***

## I. INTRODUCTION

Public lighting systems are an essential infrastructure that allows to shape the functional use of urban spaces. With the arrival of the Internet of Things (IoT), the street lighting infrastructure is considered as the backbone of smart cities services for its capillary presence and for providing connectivity and access to the power grid [1]. Its correct and efficient management is therefore essential to furnish different types of services to citizens and it is one of the cornerstones towards the digitalization of a city. The integration of Digital Twin (DT) and Virtual Reality (VR) technologies is expected to have a positive impact on the management of the street lighting infrastructure. The concept of a Digital Twin has been recently introduced by [2] and then subsequently refined in works like [3,4]. In its very essence, this is a virtual representation of a physical object or system that mirrors its real-world counterpart in terms of its structure, behavior, and context. To this end, a DT is composed by three main components: the physical entity (or a subset of its processes) plus its physical environment; the virtual model of the entity/process; and the data connection between the two sides. This last component enables the bi-directional exchange of information between the physical part and the virtual model, allowing to twin their relationships. A Digital Twin differentiates from models and simulations because it enables mutual interaction between the physical and digital world in real-time. Sensors and IoT technologies handle the transfer of information to update the virtual model according to the physical counterpart's live changes and vice-versa. In this way, the use of digital twins in street lighting systems introduces substantial opportunities for enhancing operational efficiency, cost-effectiveness, environmental sustainability, and proactive urban management. The goal is to provide technicians, stakeholders, and managers with an interactive tool to foster a data-driven understanding of the outdoor lighting system that improves their decision-making. For instance, they allow city planners to test various scenarios (e.g. analyzing the impact of different lighting schedules or bulb types on energy consumption and light pollution; configuring the required amount of light based on the hour of the day, people presence, and weather conditions, which requires a high degree of interoperability with other digital systems) without disrupting real-world operations. Thanks to live data feeds, anomalies can be detected in real-time, enabling rapid response and minimizing downtime. Similarly, Digital Twins may also facilitate predictive maintenance because they can support the identification of potential issues before they occur, reducing repair costs and enhancing the lifespan of physical assets.

In this context, the introduction of VR is particularly promising since it enriches many of the features of a DT by providing an immersive, realistic, controlled 3D environment to interact with. This immersive visualization significantly aids in comprehending complex systems and spatial

---

This work has been partially supported by the Transnet project funded by the Interreg SUDOE call of the European Commission, grant number SO4E / P1 / F0986).

relationships, thus enabling better design and operational decisions. Furthermore, VR allows for real-time, interactive simulations, facilitating scenario testing and training without risks associated with physical testing. It also enhances communication and collaboration among different stakeholders, as they can virtually navigate and experience the digital twin from any perspective. Additionally, the first-person point-of-view is particularly suitable to perform qualitative and subjective analysis of alternative scenarios.

However, creating digital twins of a city street lighting system is not straightforward and it presents different challenges. Firstly, it requires a tight collaboration between both the back- (e.g. the sensors collecting real data, the communication networks, and the IoT platform) and front-end (e.g. the 3D models, the interactive VR scenarios, and the visualization devices) subsystems. On the other hand, the involvement of heterogeneous components and technologies introduces additional difficulties in terms of synchronization, maintenance, communication, scalability, compatibility, and updating of both hardware and software components, among others. Furthermore, it is necessary to find a proper tradeoff between the graphical realism of the 3D models and the computational resources of end-devices, so that the VR experience may run as smooth as possible without losing precision in the visual feedback and expressiveness in the information to visualize (e.g. sensors data, volumetric lights).

CEDINT-UPM is currently working on a DT solution for monitoring the performance of the outdoor lighting installations at a university campus [5]. Among its objectives, it is worth mentioning the study of energy efficiency solutions, the experimentation with IoT communication and network protocols, and the comparison of lighting conditions with illumination devices by several manufacturers. Although at a pilot scale, the results obtained through the DT may form the foundations to set up a more comprehensive strategy at urban scale (e.g. parks, neighborhoods, pedestrian/restricted traffic zones). Based on this experience and the lessons learnt during its implementation, the objective of this work is three-fold: 1) to highlight the main challenges faced towards its development and deployment; 2) to introduce the technical solutions to overcome such challenges; and 3) to outline the desirable features aiming at expanding its functionalities.

## II. STATE OF THE ART

DT VR-based solutions for urban, outdoor lighting solutions are increasingly receiving more attention, as shown, for instance, in [6], where the authors summarize the state-of-the-art of the use of digital twins for supporting lighting analysis in the urban/outdoor context (up to 2022). Research works combining the use of digital twins and virtual reality tools for lighting can be divided into three main categories: optimization of lighting design, analysis of lighting perception in virtual environments and DT VR-based tools that focus on lighting aspects.

The use of virtual reality tools for lighting design allows cost and time reductions while enabling the simulation of multiple configurations and features. In [7], the authors discuss the application of VR technology in various types of smart lighting designs, for interiors and outdoor objects, comparing the computer simulations with reality. In [8], a new methodology to use Unreal Engine as a tool for lighting applications is introduced, comparing the results with those obtained from DIALux evo software. Similarly, in [9], the authors analyze the efficacy of immersive VR simulation for landscape lighting design, including user validation. VR is also integrated for BIM-based lighting design feedback, simulating daylight and artificial lights in buildings and visualizing the VR environment using head mounted displays [10].

On the other hand, the integration of digital twin tools with VR environments facilitates the interaction and provides new features in different fields. In [11], the authors present an AR application that uses Microsoft HoloLens to visualize the Digital Twin data of a CNC milling machine, improving the efficiency of the processes with a more natural human machine interface. In [12], the authors present a digital-twin-based VR-based tool to improve the design of the luminaires and decorations in the interior zone of the tunnels. In [13], the authors describe the use of a DT that combines artificial intelligence and VR for the management of a residential facility. Finally, in [14], the authors present a tool to visualize information from a city's street lighting system Microsoft HoloLens. The application, developed in Unity Pro, obtains data from both the real streetlights and a simulation model.

The use of VR environments for lighting requires a strong correlation with reality on how the light (comfort, glare, contrast) is perceived. In [15], the authors investigate the role of subjective assessments in lighting research using virtual reality, highlighting that there are limitations to overcome. In [16], the authors show the results of experimental analysis about urban lighting quality perception using VR tools such as eye-tracking. In [17], the authors present an experiment on lighting perception comparing a VR environment in HTC Vive Pro HMD and the real world. The results show similarities in well-lit scenes while differences appear in high contrast scenes. Similarly, in [18], the authors examine different daylight configurations and their impacts on users' satisfaction comparing a physical office and its digital twin, where there is correlation except in glare perception and lighting comfort at workstations.

Existing research works validate the use of VR for DT tools, especially for lighting design applications. However, to the best of our knowledge, there are no previous works that take into account the specific requirements for current and future street lighting applications: 1) integration of IoT devices (to allow connection between real and virtual worlds); and 2) processing capabilities of HMDs (to find a trade-off between graphical quality and tool performance).

## III. CHALLENGES

There are several challenges that must be solved to promote the use of DT VR-based tools for street lighting applications: increase of VR headset comfort, improvement of VR natural interaction, virtual recreation of some lighting features, close the gap between light designers and technology providers. In this paper we focus in two aspects that are essential for the adoption of DT VR-based tools for street lighting design and management: 1) the communication protocols enabling the bi-directional interchange of information between the real and digital worlds; and 2) the expedients to improve the visual rendering of the photometric behavior of the outdoor light devices (especially those LED-based) without negatively affecting the graphical quality of the DT when using commercial VR headsets.

### A. *Interaction with IoT systems*

The DT should provide bi-directional communication between the physical and digital worlds, updating the status of virtual reality components, but also enabling the control of physical systems from the DT tool. Considering the heterogeneity of IoT networks in terms of technologies, platforms and protocols, the interconnection should be done by means of Application Programming Interfaces (APIs) to the IoT platform.

In the past, proprietary solutions used to have more impact as there was less competition in the smart street lighting field. Hence, provided that one brand released a good functioning product, that manufacturer would eventually be able to deploy the rest of their infrastructure and solution on upper layers – be they: control platform, visualization, analytics, and other typical services. However, the current Smart City and Internet of Things paradigm has evolved, so there are multiple brands and manufacturers offering connected/smart streetlamps (e.g. Signify, Itron, Schreder, to name a few producers). As cities acquire street lighting features in different lots, they plead for standard connectivity solutions. This is due to mainly two factors: 1) the ease of interoperability with the rest of the IoT connected elements in the area; and 2) the lack of dependency on a single vendor, resulting in benefits such as ease of reparability, greater availability of spare pieces and technicians, and more market offers for future deployments, among the others.

#### 1) *IoT connectivity*

There are several IoT connectivity solutions, with different features depending on the final goal and application [19]. In the lower layers, the most suited technologies for street lighting applications are LoRa, NB-IoT, and 6LoWPAN.

LoRa employs a low power, wide-area network (LPWAN) protocol designed for long-range communication at a low bit rate, leveraging an efficient remote management and real-time street lighting control when higher bitrates are not needed [20].

NB-IoT (Narrow Band-Internet of Things) is another prominent technology that is having a significant presence and market, as the largest telecommunications companies promote it. This technology uses (sometimes) existing cellular infrastructure to assure wide area coverage and direct cloud connectivity. One of its key benefits is the existence of a working network and (more or less) international coverage and compatibility.

6LoWPAN (IPv6 over Low-Power Wireless Personal Area Network) is a network protocol defined by the IETF that allows for the transmission of IPv6 packets over low-power wireless networks. This protocol is useful in scenarios where systems with limited power or processing capabilities need to be connected to a wider network, or if the use case requires direct access to every end node, both ways; a factor that the other two cannot provide.

#### 2) *Application interface*

On the upper layers, the most common protocols are MQTT (Message Queuing Telemetry Transport) and CoAP (Constrained Application Protocol). Both are communication protocols designed for the IoT, but MQTT is a publish-subscribe-based messaging protocol whilst CoAP inherits a more client-server, request-like approach. However, one does not necessarily exclude the other. Achieving the proper balance between power and computational resource demands, as well as meeting the constrained real-time requirements of a DT, is possible when both protocols are employed.

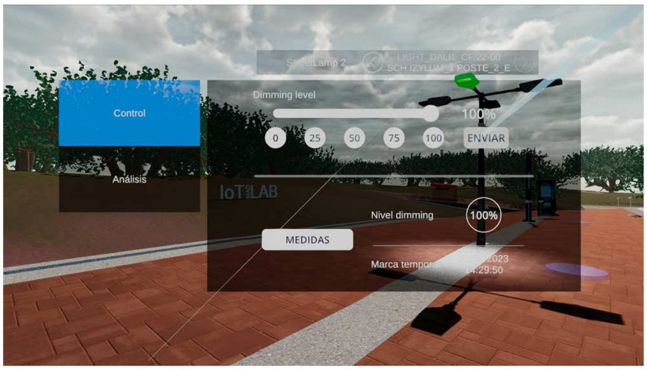


Fig. 1 – Screenshot illustrating the interface to control the dimming level of streetlights form the VR environment of the DT at University Campus de Montegancedo (Pozuelo de Alarcón, Madrid – Spain). All communication occurs in real-time to retrieve/update the status of the selected streetlamp as well as to show data and statistics to visualize (currently through the RESTful protocol only).

Using MQTT allows the VR system to be linked to real-world changes thanks to a subscription mechanism through a socket connection open between them. This way, as soon as any change or event is detected by any sensing element in the real world (such as temperature sensors or traffic counters), this event is instantly pushed to the VR system. There is no need to monitor the state sensors periodically, minimizing communications and processing resources, and it also possible to avoid the delay that any sampling period would cause. In fact, this sampling period ends up being a trade-off among resources availability, number of sensing entities, and permitted VR delay. On the other hand, RESTful protocol is useful to make the VR interactions become real (Fig. 1). Such interactions are specifically committed by the user, being either a control command (e.g. a PUT command for switching on a streetlight), or for information request (e.g. GET command for asking for the status of a sensor or device).

### B. *Graphic rendering*

A precise and realistic graphical rendering of lighting is a crucial aspect for assessing its potential effects on human and wildlife activities, such as road traffic safety, pedestrian safety, and reducing light pollution. The subjective response is also a critical factor in evaluating lighting solutions. Current cutting-edge software can simulate spatial average luminance from light sources, excelling particularly in interior spaces, where architectural components create a more predictable lighting environment. However, accurate simulation of outdoor lighting effects remains a challenging task. Artificial light reproduction involves mirroring the behavior of diverse light sources like streetlights, building façade lighting, or architectural lighting installations. This precision demands calculation of the photometric properties of different light sources, including their intensity, color temperature, and directionality, both individually and in combination. Outdoor environments present additional complexity due to natural illumination conditions derived from sunlight, atmospheric scattering, and environmental geometry, among other factors. It is essential to consider how natural and artificial sources interact to accurately render light distribution.

#### 1) *Commercial software*

Characterizing the photometric properties of all light sources and materials present in the scene is a primary challenge. Development of intricate mathematical models that can capture distinct, dynamic lighting effects, such as sunlight, atmospheric scattering, and object shadows, becomes fundamental. These models are solved by specialized software considering constraints like numerical precision, the number of equations or terms to consider, and expected output/response time (e.g. real-time applications). The results are then transformed into suitable visual representations through sophisticated rendering techniques to achieve overall, realistic illumination effects. Notable examples of such software suites include Lumion [21], V-Ray [22], Relux [23], and DIALux [24], each with unique capabilities and features. Each of these software packages has demonstrated their proficiency in specific domains, contributing significantly to the advancement and versatility of design and visualization processes. One of the primary benefits of these tools is their ability to accurately simulate different lighting scenarios, allowing designers to optimize lighting conditions before implementation. This experimentation can lead to significant cost savings and improved aesthetic and functional results. Additionally, these tools typically feature extensive libraries of light fixtures and materials, enabling high detail in renderings. However, this specialized software requires substantial computing resources, potentially limiting some users. They also have steep learning curves and mastering them may require time and resources. Furthermore, while this software provides impressive visualization capabilities, they are still simulations, and real-world results might not always align with the rendered output due to variables like weather conditions and material properties that are challenging to configure in these environments.

### 2) *Graphical engines*

Graphical engines emerge as beneficial tools when superior real-time visualization, specific hardware compatibility, or customized visualizations concerning interaction, deployment, or data sources become a necessity [25]. Unity 3D [26] and Unreal Engine [27] are two leading game development platforms currently revolutionizing the gaming industry, providing robust frameworks for crafting immersive, visually stunning, and interactive experiences. Both offer a comprehensive set of features for creating top-tier games, ranging from 2D mobile games to AAA console titles. They also serve as excellent tools for research applications due to their robust features and extensive customization options, enabling researchers to simulate a broad spectrum of environments and scenarios. Both engines support real-time rendering and interaction, which are crucial for studies involving human-computer interaction and VR. They also boast large, active communities and comprehensive documentation, offering an abundance of resources and support for researchers and practitioners.

Both engines aim to deliver high-quality, realistic visuals while maintaining real-time performance, that is two of the most critical aspects impacting the visual quality and immersive experience of VR application development. Unity 3D introduces two distinct, scriptable rendering pipelines, namely the Universal Render Pipeline (URP) and the High-Definition Render Pipeline (HDRP). Each pipeline possesses unique lighting models and attributes that significantly shape the visual rendering of virtual environments. The URP is engineered for optimized performance across diverse hardware, maintaining high levels of graphical fidelity. It accommodates various lighting functionalities, including Global Illumination, Light Probes, and a Mixed Lighting mode. URP is performance-driven, making it an ideal choice for applications targeting platforms such as mobile and VR where resources may be constrained. It offers a scalable solution, catering to varying visual fidelity levels depending on the platform's capabilities. Furthermore, URP's simplified lighting setup provides ease of use, particularly for beginners or smaller-scale projects. However, URP falls short of some advanced lighting features available in HDRP, such as volumetric lighting or path-tracing. While performant, the visual quality offered by URP may not match HDRP, particularly in highly realistic scenes. On the other hand, HDRP, tailored for high-end platforms like PC and consoles, aims to deliver high-definition visuals. It accommodates a broad spectrum of advanced lighting features, including Physically Based Rendering (PBR), real-time ray tracing, volumetric lighting, and screen space reflections. HDRP encompasses numerous advanced lighting techniques, such as volumetric fog and area lights, enabling visually impressive scenes. Since it is designed for high-end games, it employs a physically based lighting model, augmenting the realism of materials and surfaces under different lighting conditions. Furthermore, HDRP's custom render allows developers to integrate their rendering effects, offering a high degree of control and customization. On the contrary, HDRP's advanced features can be resource-demanding, making it less suitable for lower-end platforms. Additionally, HDRP possesses a steeper learning curve due to its comprehensive range of features and settings.

With an approach like Unity's HDRP pipeline, Unreal Engine exhibits exceptional prowess in its visual capabilities. Concerning its lighting functionalities, it offers a robust suite of features, including sophisticated lighting, shading, and particle systems. This enables developers to create highly immersive and visually stunning environments. Its lighting system, leveraging physically-based rendering and real-time ray tracing, facilitates highly accurate rendering of light and shadows, thus enhancing the realism of illumination scenarios. Moreover, the engine supports advanced illumination features such as dynamic global illumination and volumetric fog, providing a nuanced depth and realism to virtual environments. However, these robust capabilities come with certain challenges. The computational demands associated with its advanced lighting features can render Unreal Engine resource-intensive, potentially limiting its usability on less powerful hardware, such as the VR headsets. Furthermore, the engine's lighting system is complex, presenting a steep learning curve that could be a barrier for new developers. Moreover, striking an optimal balance between performance and visual quality often necessitates a deep understanding of the engine's features and a meticulous approach to optimization.

### 3) *Light emission profiles*

Enhancing light emission and distribution in Unity3D can be significantly bolstered using illumination profiles. A leading standard in this context is provided by the Illuminating Engineering Society (IES), widely recognized and employed within the architecture and lighting design sectors [28]. The IES standard is used to encapsulate and disseminate photometric data, which is a quantifiable interpretation of the spatial light intensity from a real-world light source. This data is derived from laboratory examinations of actual physical light fixtures, allowing for a precise reproduction of the light

Fig. 2 – A virtual representation of a streetlight. Only a lamp is currently switched on. Above: with surface shader; Below: with street light cookie.

Fig. 3 – Similar to Fig. 2, with all the four streetlights turned on to see the combining lighting effects at sunset (above) and during daylight (below).

source's distribution. The utilization of IES files within 3D rendering scenarios is primarily advantageous due to the enhanced realism and accuracy they contribute to scene recreation. Light, a complex and multifaceted phenomenon, is often challenging to simulate with precision. IES files enable capturing the intricate characteristics of a specific light source's emission and illumination, thus offering a more comprehensive and realistic depiction. Currently, Unity3D HDRP natively supports the importation of IES files while URP does not. For URP, a possible alternative is to employ specific shaders that can be referenced as additional properties of object materials, both for surfaces and components lighting and for lamp footprint (cookie). While the application of IES files within Unity3D HDRP brings a heightened level of realism and immersion within virtual scene lighting, it augments the complexity of the project due to the necessity for additional assets and an increased burden on the rendering pipeline. This can create performance challenges, especially when a scene incorporates multiple IES profiles. Furthermore, identifying the appropriate IES file to meet specific needs can be a time-consuming process, as each file is tailored to a particular type of light source.

For the balance between achieving realism and maintaining real-time performance, as well as the type of application being implemented, the Digital Twin developed at CEDINT-UPM has been implemented in Unity3D URP (see Fig. 2 and Fig. 3)

### 4) *3D Modelling*

Another fundamental aspect for the development of a visually/aesthetically/photometrically realistic digital twin is the high-fidelity modelling of the outdoor environment. This involves the meticulous recreation of the physical space in the digital domain, with precision not only in the macroscopic elements such as object positions and dimensions, but also in the finer details including material properties, surface textures, and color characteristics, among others. Concrete issues to consider include the accurate representation of the geometrical layout, where even minor inaccuracies can lead to substantial discrepancies in the resultant lighting simulation. Additionally, the reflection and absorption properties of materials used in the environment have a significant impact on light behavior and must be accurately considered. These elements play a central role in determining the aesthetics and photometric performance of a lighting design. Without such high-fidelity modeling, the digital twin may fail to accurately predict real-world lighting conditions, leading to discrepancies between the virtual model and the physical reality.

### 5) *Human perception*

Finally, it is worth noting that the creation of a truly realistic digital twin must not only consider the technical aspects described above, but also encapsulate the nuanced human perception of light and aesthetic evaluations. The human visual system's response to light is a complex interplay of physiological and psychological factors. Understanding and incorporating how different lighting conditions are perceived by human observers into the digital twin is essential for the design of aesthetically pleasing and functionally effective lighting scenarios. However, this subjective evaluation of aesthetics varies significantly among individuals, making the task of integrating it into a universally applicable digital model challenging. Consequently, the development of a visually, aesthetically, or photometrically realistic digital twin for lighting analysis demands a multi-disciplinary approach, bridging the gap between rigorous scientific modeling and the subjective realm of human perception.

## IV. CONCLUSIONS

Our experience in developing a Digital Twin for street light solutions suggests that it is important to establish different trade-offs in the implementation of such virtual replica. The goals and the operational context are some of the main drivers to consider deploying a tool being as accurate and versatile as possible in assisting stakeholders' decisions on the qualitative analysis of an outdoor lighting scenario. This paper focuses on two specific technical aspects that can provide greater operability and adoption to a VR-based DT: IoT communication protocols and simulation photorealism.

The smart street lighting ecosystem presents a heterogeneous landscape of IoT technologies. However, in order to cover the features of the digital twin, the connection between the physical and virtual world should be bi-directional and both event and request based. For this end, MQTT (publish-subscribe-based) and CoAP (client-server, request- approach) should be combined.

The integration of VR engines, specialized lighting software, and IES profiles significantly advances photometric rendering in outdoor lighting applications. This blend creates a dynamic 3D platform for photorealistic light simulations, aiding in design customization. IES profiles bolster authenticity of lighting effects, while VR environments facilitate early issue detection and resolution, conserving resources and time. However, challenges persist. Real-time VR lighting rendering poses substantial computational demands, potentially hampering performance and requiring robust hardware. A potential solution lies in pre-computing lighting configurations, improving VR resource efficiency but limiting design versatility. Accurate IES profile implementation in VR necessitates complex calibration and extensive knowledge, which can be time-consuming and error-prone. Lastly, the steep learning curve associated with these technologies underscores the need for a multidisciplinary approach to expedite development and maximize potential.

## REFERENCES

[1] T.H. Kim, C. Ramos and S. Mohammed, “Smart city and IoT”, Future Generation Computer Systems, 76, pp. 159-162, 2017.

[2] M. Grieves and J. Vickers, “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems”, Transdisciplinary perspectives on complex systems: New findings and approaches, pp. 85-113, 2017.

[3] F. Tao, and M. Zhang. “Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing”. IEEE Access, vol. 5, pp. 20418-20427, 2017.

[4] F. Jiang, L. Ma, T. Broyd, and K. Chen. “Digital twin and its implementations in the civil engineering sector”. Automation in Construction, vol. 130, 103838, 2021.

[5] L. Piovano, G. Calderon, G. D. Campo, E. Saavedra, F. Luque and A. Santamaría, "Towards a Digital Twin for Smart Street Lighting systems Using a Virtual Reality interface," 2021 Joint Conference - 11th International Conference on Energy Efficiency in Domestic Appliances and Lighting & 17th International Symposium on the Science and Technology of Lighting (EEDAL/LS:17), Toulouse, France, 2022, pp. 1-6.

[6] M.U. Hassan, S. Angelaki, C.V.L. Alfaro, P. Major, A. Styve, S.A.A. Alaliyat, ... & R. da Silva Torres, “Digital Twins for Lighting Analysis: Literature Review, Challenges, and Research Opportunities”. In 36th International ECMS Conference on Modelling and Simulation, ECMS, vol. 36(1), pp. 226-235, 2022.

[7] R. Krupiński, “Virtual Reality System and Scientific Visualisation for Smart Designing and Evaluating of Lighting”, Energies, vol. 13, 5518, 2020. https://doi.org/10.3390/en13205518.

[8] M. Scorpio, R. Laffi, A. Teimoorzadeh, G. Ciampi, M. Masullo, S. Sibilio, “A calibration methodology for light sources aimed at using immersive virtual reality game engine as a tool for lighting design in buildings”, Journal of Building Engineering, vol. 48, 103998, ISSN 2352-7102, 2022. https://doi.org/10.1016/j.jobe.2022.103998.

[9] J.-H. Lee and Y. Lee, “The effectiveness of virtual reality simulation on the qualitative analysis of lighting design”. Journal of Digital Landscape Architecture, vol. 6, pp. 195–202, 2021.

[10] W. Natephra, A. Motamedi, T. Fukuda et al., “Integrating building information modeling and virtual reality development engines for building indoor lighting design”, Vis. in Eng., vol. 5(19), 2017. https://doi.org/10.1186/s40327-017-0058-x.

[11] Z. Zhu, C. Liu, X. Xu, “Visualisation of the Digital Twin data in manufacturing by using Augmented Reality”, Procedia CIRP, vol. 81, pp. 898-903, ISSN 2212-8271, 2019. https://doi.org/10.1016/j.procir.2019.03.223.

[12] Y. Shen, J. Ling, X. Li, H. Li, S. Feng, H. Zhu, “Holistic digital-twin-based framework to improve tunnel lighting environment: From methodology to application”, Building and Environment, vol. 224, 109562, ISSN 0360-1323, 2022. https://doi.org/10.1016/j.buildenv.2022.109562.

[13] E. M. Elfarri, A. Rasheed and O. San, "Artificial Intelligence-Driven Digital Twin of a Modern House Demonstrated in Virtual Reality," in IEEE Access, vol. 11, pp. 35035-35058, 2023, doi: 10.1109/ACCESS.2023.3265191.

[14] W. Piper, H. Sun and J. Jiang, "Digital Twins for Smart Cities: Case Study and Visualisation via Mixed Reality," 2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall), London, United Kingdom, 2022, pp. 1-5.

[15] M. Scorpio, D. Carleo, M. Gargiulo, P.C. Navarro, Y. Spanodimitriou, P. Sabet. M. Masullo, G. Ciampi, “A Review of Subjective Assessments in Virtual Reality for Lighting Research”, Sustainability, vol. 15, 7491, 2023. https://doi.org/10.3390/su15097491

[16] M. Masullo, F. Cioffi, J. Li, L. Maffei, G. Ciampi, S. Sibilio, M. Scorpio, “Urban Park Lighting Quality Perception: An Immersive Virtual Reality Experiment”, Sustainability, vol. 15, 2069, 2023. https://doi.org/10.3390/su15032069.

[17] S. Rockcastle, M. Danell, E. Calabrese, G. Sollom-Brotherton, A. Mahic, K. Van Den Wymelenberg, and R. Davis, “Comparing perceptions of a dimmable led lighting system between areal space and a virtual reality display”. Lighting Research & Technology, vol. 53(8), pp. 701-725, 2021.

[18] R. Jafarifiroozabadi, P. MacNaughton and A. Osnaga, "Investigating Lighting Quality in Office Workstations: A Combined Approach Utilizing Virtual Reality and Physical Workstations," 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Christchurch, New Zealand, pp. 85-87, 2022, doi: 10.1109/VRW55335.2022.00029.

[19] G. del Campo, I. Gomez, G. Cañada, L. Piovano, A. Santamaria, “Guidelines and criteria for selecting the optimal low-power wide-area network technology”, LPWAN Technologies for IoT and M2M Applications, Academic Press, pp 281-305, 2020, https://doi.org/10.1016/B978-0-12-818880-4.00014-4

[20] E. Saavedra, L. Mascaraque, G. Calderon, G. del Campo, A. Santamaria, “A Universal Testbed for IoT Wireless Technologies: Abstracting Latency, Error Rate and Stability from the IoT Protocol and Hardware Platform” Sensors, vol 22, 4159, 2022. https://doi.org/10.3390/s22114159

[21] https://lumion.com/ (Last access: 30 May, 2023)

[22] https://www.chaos.com/all-products#vray (Last access: 30 May, 2023)

[23] https://reluxnet.relux.com (Last access: 30 May, 2023)

[24] https://www.dialux.com (Last access: 30 May, 2023)

[25] C. Morse, “Gaming Engines: Unity, Unreal, and Interactive 3D Spaces”, Technology | Architecture + Design, vol. 5(2), pp. 246-249, 2021. https://doi.org/10.1080/24751448.2021.1967068

[26] https://unity.com (Last access: 30 May, 2023)

[27] https://www.unrealengine.com (Last access: 30 May, 2023).

[28] Illuminating Engineering Society. “ANSI/IES LM-63-19, Approved Method: IES Standard File Format for the Electronic Transfer of Photometric Data and Related Information”. New York: Illuminating Engineering Society, 2019

# Towards a Digital Twin for Smart Street Lighting systems Using a Virtual Reality interface

Luca Piovano
*CEDINT-UPM*
*Universidad Politécnica de Madrid*
Madrid, Spain
lpiovano@cedint.upm.es

Guillermo del Campo
*CEDINT-UPM*
*Universidad Politécnica de Madrid*
Madrid, Spain
gcampo@cedint.upm.es

Francisco Luque
*CEDINT-UPM*
*Universidad Politécnica de Madrid*
Madrid, Spain
fluque@cedint.upm.es

Gonzalo Calderon
*CEDINT-UPM*
*Universidad Politécnica de Madrid*
Madrid, Spain
gcalderon@cedint.upm.es

Edgar Saavedra
*CEDINT-UPM*
*Universidad Politécnica de Madrid*
Madrid, Spain
e.saavedra@upm.es

Asunción Santamaría
*CEDINT-UPM*
*Universidad Politécnica de Madrid*
Madrid, Spain
asun@cedint.upm.es

***Abstract*—The smart management of the public lighting infrastructure is essential to reduce energy consumption without negatively affecting the quality of life of citizens. The corresponding decision-making process needs tools and approaches to steer stakeholders' strategies based on clear evidence and data. By introducing Virtual Reality in the context of the Digital Twin framework, it is possible to mirror a system in real-time and at scale through a realistic and controlled reproduction of the real environment and its functionalities. Interactivity, immersion, and visualization techniques are expected to bring more understandability and replicability in the management and design of street lighting policies. In this paper, a Digital Twin solution for monitoring the performance of the outdoor lighting installations at a university campus is proposed. Through a Virtual Reality replica of the campus, current lighting management services are introduced to showcase the advantages of the approach.**

***Keywords—Smart Lighting, Street Lighting, Virtual Reality, Digital Twin, Internet of Things***

## I. INTRODUCTION

Public lighting systems, being an essential asset to shape the appearance and the functional use of public spaces of cities, account for 13-14% of electricity consumption worldwide [1], and are responsible of night sky pollution, which affects both biodiversity and human sleep cycles [2]. The recent use of Light Emitting Diode (LED) lamps has decreased the electricity demand (from a high 19%), but also introduced a greater control over several of its features (e.g. reducing energy consumption, maintaining comfortable conditions, ensuring safety to drivers and vulnerable road users, and being respectful with the environment) [3,4]. To achieve these expected advantages, its smarter management may be addressed from a holistic approach based on multiple Information and Communication Technologies (ICT). In this scenario, the adoption of the Digital Twin (DT) paradigm may represent a further step towards a more comprehensive and evidence-based decision-making process [5]. The aim of a Digital Twin is to have a virtual copy of a real/physical system mirroring in real-time a (sub)set of its features of interest. To this end, a DT is mainly composed by five components: a physical asset, its corresponding virtual (or digital) representation, a connection / communication infrastructure, data streams, and a service/application layer [6,7].

Among the DT enabling technologies in the street lighting domain, there are several implementations of Internet of Things (IoT) systems (e.g. see [8] for a review of some cases of study). Smart city platforms providing access to several data sources have been proposed as well to tackle public lighting issues (e.g. a description of a possible architecture to deploy a digital twin at a city scale is presented in [9]). Artificial Intelligence strategies for massive data analysis are also explored to drive processes and decision-making activities based on solid evidence (e.g. a recent European project aiming at facilitating the implementation of the digital twins throughout the phases of Solid-State Lighting product design as well as boosting their applications upstream, up to digital twins of lighting systems of large infrastructures is described in [10]).

The virtual representation of the real, outdoor infrastructure is critical for the deployment of a Digital Twin solution since it provides the simulation / testing environment. However, it seems that there is a gap between the *theoretical* aspects (i.e. the methods used to understand the operational context, to design future scenarios, and to estimate their possible effects) and their *representation* in practice (that is, how to visually present, inspect and analyze the different aspects of a simulation/solution). Theoretical aspects could consist of mathematical models or the raw, underlying data, while practical representation consists of charts, physical mock-ups, and web-based interfaces. The introduction of Virtual Reality (VR) as a tool to propose, simulate and assess different solutions for the smart management and monitoring of the performances of outdoor lighting installations (e.g. adapting dimming values according to the environmental conditions) is particularly promising to bring several benefits to stakeholders. VR allows a realistic and controlled reproduction of real environments at scale. Intuitive interactions with 3D objects, extended possibilities to put in the right context the information at disposal and the capability for simulating scenarios make it possible for users to approach the management of complex systems under a distinct perspective. Additionally, VR allows users to experience a high degree of immersion (i.e. the capacity to isolate from the stimuli of the surrounding environment and feel the presence in the virtual world as it was the real one) [11]. In this sense, the Virtual Reality approach is expected to bring more reproducibility, understandability, and interpretability to the underlying theoretical and modelling steps. VR is not an absolute novelty in the domain of outdoor lighting systems. It has especially employed in the process of assessing the suitability of a solution from users' perspective, where the focus has been on reproducing the photometric behavior of the

---

This work has been a joint effort of the SuperVR project funded by the Ministerio de Ciencia e Innovación (MICINN) of Spain, grant number TEC2017-83769-R; and the Tr@nsnet project funded by the Interreg SUDOE call of the European Commission, grant number SO4E/P1/F0986).

light distribution to replicate the human perceptions and experiences in real spaces [12,13]. However, to authors' knowledge, the adoption of a VR-based solution to support the operational control (e.g. supervision, check for failures, maintenance) over the lighting components and/or the system as a whole, to frame the general decision-making process for its management, is still lacking.

The objective of this paper is to present the practical implementation of a Digital Twin solution for monitoring the performance (e.g. detecting breakdowns or reducing overall energy consumption) of the outdoor lighting installations at a university campus. Emphasis is put on the immersive and interactive VR tool creating a virtual replica of the real environment and the main functional features of the streetlights. This virtual environment offers intuitive interaction to perform typical monitoring tasks, through both Head Mounted Displays (HMDs) and its controllers as well as a voice interface (e.g. commercial voice assistant Alexa).

The rest of the paper is organized as follows: Section 2 describes the main components of the DT, at both physical and virtual level, the technologies used for its deployment and the interconnections between the different elements of the DT itself; Section 3 introduces the services currently implemented for monitoring the lighting installation; and Section 4 presents the future research lines to expand the DT capabilities.

## II. SYSTEM DESCRIPTION AND DT MAIN COMPONENTS

The building blocks composing the Digital Twin infrastructure for its management and operational control are shown in Fig. 1. The installation of an efficient street lighting system based on LED lamps has been deployed at Campus of Montegancedo (Universidad Politécnica de Madrid, Spain), covering the pedestrian areas and a parking lot. The integration of IoT sensors and actuators in the streetlights provides the data to feed the virtual/digital world; the virtual world replicates, simulates and helps users to predict the behaviors of the physical elements; the communication network and the IoT platform establish a bridge between the real and virtual components; the service layer provides the added value of the DT, by enabling, for instance, monitoring and control capabilities, by means of virtual interfaces based on HMDs and voice assistant.

### *A. The physical world*

There are 69 LED streetlights, each one equipped with a smart lighting device (Bat Street Light – BSL). BSL include light dimming control, ambient illumination monitoring and presence detection [14]. The BSLs are wirelessly connected in a mesh network using 6LoWPAN (BatNet), which guarantees communication to the Internet using the IPv6 protocol.

The streetlights can work either autonomously or manually. The autonomous working mode is based on a finite-state machine (FSM) with a smart algorithm built on hysteresis and dynamic thresholds. Streetlights measure the luminosity at their surroundings to decide when to trigger their luminaire. They are illuminating with the minimum legal amount of light allowed until they detect a (human) presence. In such a case, the streetlight detecting such event switches to its maximum luminance level and, moreover, it propagates this information to its streetlight neighbors, so they may switch their lights on as well. This way, it enhances pedestrian visual comfort while optimizing energy consumption. On the other hand, the manual working mode allows streetlights to respond to remote control commands.

### *B. The IoT platform*

Data from the smart lighting devices are stored in real-time through an IoT platform that combines Elastic Stack tools (Elasticsearch, Kibana, and Beats) and Apache Kafka [15]. Elasticsearch is a search engine that provides a distributed architecture and data replication to index the data; Beats

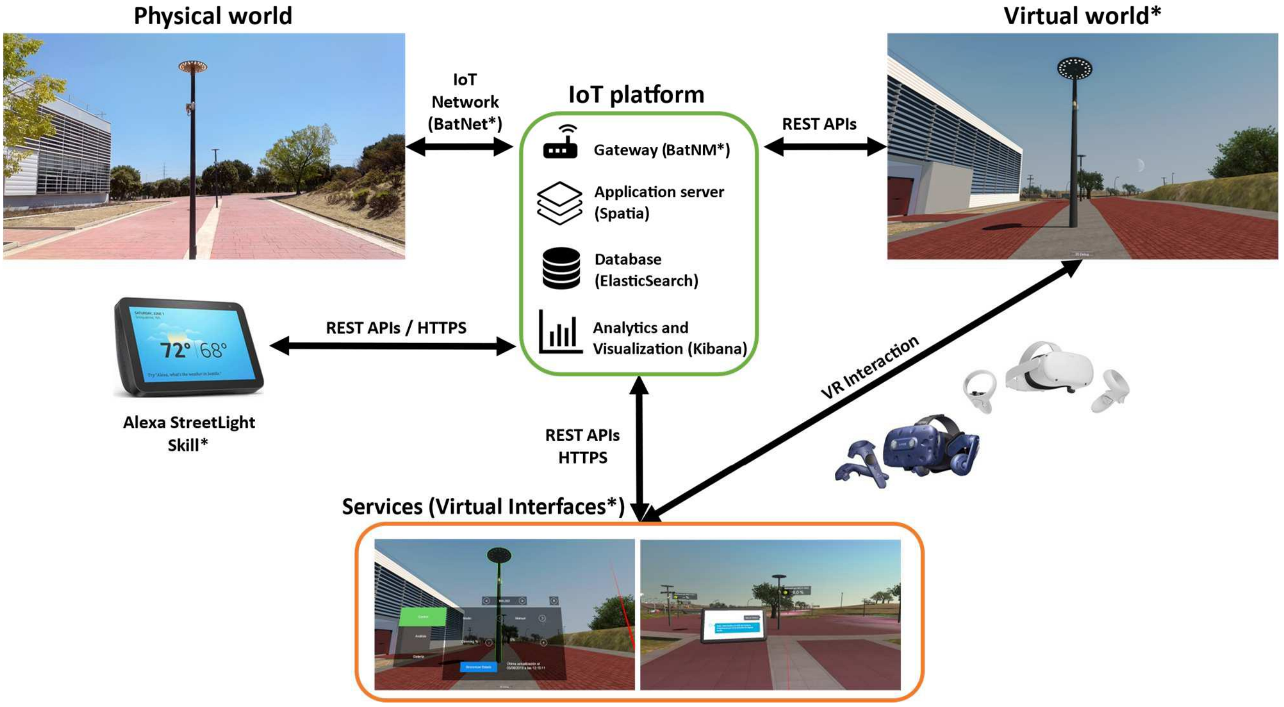


Fig. 1. Main building blocks for the Digital Twin implemented for the outdoor lighting system at Campus of Montegancedo (Madrid, Spain). Black arrowed lines highlight the communication channels, and they are labeled with the employed protocols or means. Solutions marked with an asterisk (*) refer to a CEDINT-UPM technological product.

monitors and gets operational data from the systems and services of the platform; Kibana provides a web interface to explore, visualize, and display the data stored in Elasticsearch; and Apache Kafka follows a publisher/consumer approach to data exchange between the platform and the IoT network.

The IoT platform manages the IoT devices of the BatNet network (i.e. the BSLs attached to the streetlights). The Gateway (BatNM) makes GET requests periodically (by default, every 5 minutes), to the streetlights to collect the values from the devices (dimming level, ambient illumination, presence detection and current FSM state). Such measures are organized as JSON documents and sent to the application server (Spatia) and from there, they are dispatched to the Elasticsearch database where they are conveniently stored. Additionally, through a web application integrated within Spatia, it is possible to interact with the IoT devices to check their status and current information. Finally, the Kibana suite provides a customizable interface to perform advanced, visual analysis of the collected data (e.g. creation of analytic dashboards to visualize time-series measurements at different temporal granularity; definition and visualization of changes in relevant Key Performance Indicators).

### *C. The Virtual world*

The Virtual Reality environment, implemented with the Unity 3D engine, relies on a 3D digital model of the Campus where the smart lighting solution is currently deployed (Fig. 2). The virtual model is accurate from a metric point of view and relies on GPS coordinates for the positioning of the streetlights (and related sensors) in the virtual scene. Besides, it is also visually realistic since the models have been textured using close-up photos of the real materials. To provide a smooth, immersive experience, it has been modeled, texturized, and optimized according to the different hardware requirements and limitations of the most used commercial HMDs. For instance, a low poly model with lower resolution textures is used when running the application in full portable HMD devices, because these headsets, working without a link cable connection to a PC, usually have less computing and graphical power. Additionally, different optimization techniques have been implemented to prevent possible frame drops that can affect the quality of the experience as well as VR-motion discomfort and sickness to users. It includes object culling (i.e. only the objects in the users' cone of view are rendered); meshes with different Levels of Detail (i.e. far objects do not require the same details as the closer ones); reduction of visual effects that require heavy computation without affecting the global visual quality (e.g. shadow or transparent materials). The current solution has been implemented for two commercial headsets, namely the HTC Vive Pro and the Oculus Quest 2.

### *D. The Virtual interfaces*

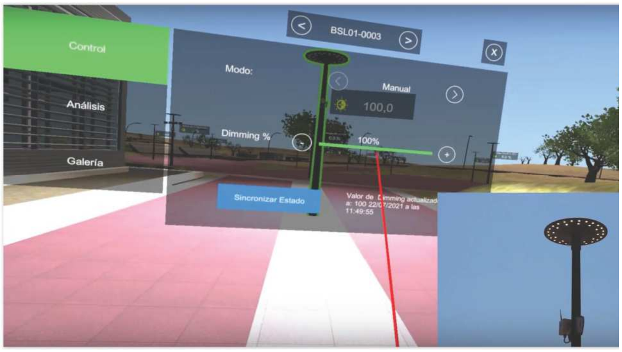
Navigation in the virtual scene is achieved through a first-person perspective for the most immersive experience. HMD users can move freely inside the calibrated physical space and the virtual camera view is adjusted according to the HMD rotational and positional tracking. To be able to virtually travel at a campus-scale distances, users can teleport themselves to different locations by using VR controllers. Users may also choose to change their point of view of the global smart lighting solution through some pre-defined aerial perspectives. The VR controllers can also enable the interaction with any of the operational devices inside the virtual scenario (i.e. the streetlights). When hovering the pointer towards any of these, the corresponding model is highlighted in yellow and can be activated by pulling the main trigger of the controller. An active device is represented by a green highlight border and shows a 3D user interface in front of the camera describing its status and offering control and analysis options.

Besides hand controllers, the virtual environment also counts with the implementation of a more natural human-computer interface. It is possible to interact with the Amazon's virtual assistant Alexa by using specific working utterances. A skill to control and manage the smart streetlight solution (together with other smart device networks) has been implemented as described in [16]. To enable interaction with Alexa, a 3D model of the voice assistant is shown in the virtual scenario. The users must press the trigger of the VR controller to enable it and then, spell the command (e.g. "Alexa, tell me the luminance value of streetlight number 3"). The VR application takes advantage of the microphone installed in the HMD to record voice orders and sent them to the skill over the Internet. The assistant's responses are conveyed back to the users through both the HMD's headphones (audio feedback) and in text format displayed on the assistant 3D model (visual feedback).

### *E. Communication channels*

In order to enable external applications and services make requests of the data stored in the Elasticsearch database, the IoT platform integrates an Application Programming Interface based on a Representational State Transfer framework

Fig. 2. The Campus of Montegancedo (Madrid, Spain) where the smart lighting solution have been deployed: real vs. digital version (above and below, respectively)



Valor de Dimming actualizad
a: 100 22/07/2021 a las
11:49:55

Fig. 3. Interacting with the dimming controller. The user is switching on the streetlight from the VR environment. The corresponding streetlight in the campus receives the command and automatically updates its status.

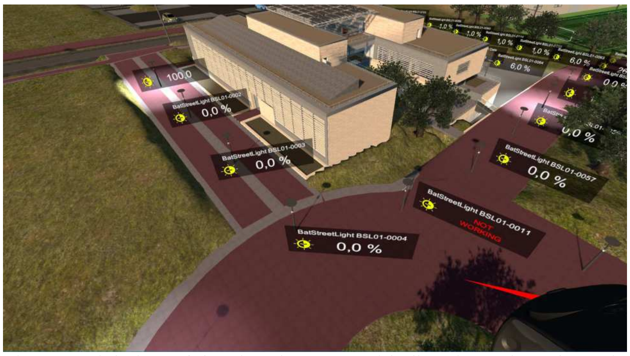


Fig. 4. Aerial view of the virtual campus with dimming values shown in a panel close to each streetlight. In this way, a visual summary of the current situations is shown to the user.

(RESTful API). These interfaces provide a bi-directional communication channel, so that they not only expose the information of interest from the real world to its virtual counterpart but accept that the digital world may directly act on the real devices as well. Data can be requested in their raw format or using some statistical aggregation (e.g. the average of the dimming values in a specific time range). On the other hand, Kibana puts at the analyst's disposal a visualization framework, whose charts and/or dashboards can be exported via Inline Frames (IFrames) and embedded in external web applications.

To synchronize the physical and virtual worlds, the VR environment makes use of this communication layer. A first data exchange occurs during the booting up phase of the VR environment, when it opens a secure authenticated connection with the IoT platform. Apart from retrieving the JSON list of streetlights to populate the digital scenario, it also gathers the relevant information for their configuration such as the device IDs, the operational status, and the GPS coordinates for the positioning of the corresponding 3D models inside the scene. This scheme enables to scale up the application to any number of devices and possible changes in the locations of the streetlights / sensors. The VR application may be configured to query on-demand the last known sensor values and the level of dimming for each streetlight. The latter is used to compute the light intensity around each streetlight and the LEDs of the luminaire. The visual appearance of the corresponding 3D models is then updated in accordance with the expected value of light contribution for each streetlight. The Unity 3D engine is the responsible to render such visual effect as much accurately as possible to enhance the realism of the representation of the actual illumination.

## III. IMPLEMENTATION OF MANAGEMENT CASE STUDIES

To showcase the potential of the VR approach coupled with a DT infrastructure for the efficient management of an outdoor lighting system, the following services have been implemented.

### A. *Controlling the smart sensors and actuators of the streetlights*

Users can remotely change the level of dimming of a given streetlight by interacting with the corresponding slider embedded in the virtual interface. Whenever a virtual streetlight is selected, the VR environment sends a message to the IoT platform to update its current value. The dimming value is expressed in a range from 0% (totally switched off) to 100% (completely switched on). Once the desired value has been selected and confirmed, a message is sent to the IoT platform to be reflected in the real world. If the action has been accomplished successfully (i.e. an acknowledge response from the platform is received), the VR environment is updated to reflect the new status while a synchronization text is presented to the user confirming the value introduced and the timestamp of the event. At once, the LEDs of the corresponding streetlight change their dimming status accordingly. In case of a communication failure, an error message is displayed to acknowledge the user and the dimming value falls-back to the last known valid state. This service is particularly useful when there is a need to perform specific maintenance operations, for instance: to test the connectivity of the smart devices, to check possible streetlight hardware failure, or to switch on/off only specific sectors of the lighting installation. Fig. 3 shows an example of such an interaction. The user is controlling the dimming rate of a streetlight by interacting with the slide-bar. The inset in the bottom-right corner shows the status of the corresponding lamp. The status of the virtual and real objects is then synchronized, and the corresponding value is stored in the data platform. In this case, the working mode of the streetlight has been set to "manual" to allow this direct interaction. In Fig. 4, an aerial view of the campus shows some of the streetlights belonging to the smart lighting system. Beside each digital model, an informative panel presents the dimming status of the lamp. It is possible to observe that most of the lamps are switched off (dimming value set to 0%), because it is daytime. However, some lamps are turned on, such as the first lamp to the left being lit at its maximum level. On the other hand, there is a streetlight showing a "Not Working" warning indicating a failure in retrieving such a measure. This information would help the system manager investigate the reasons of this malfunction (e.g. connection timeout, the sensor was not able to record or sent the information or the lamp has broken).

### B. *Visual analysis of time series*

The VR application may retrieve from the IoT platform the data records about specific variables of interest (currently dimming or illumination values only) and visualize them in a Kibana-based, interactive chart for their visual inspection. Being time series, data are presented as line charts: each dot is a convenient average value for a given time range, to highlight trends and de-clutter the graph. Users can interact with the graph in the same way as in a regular browser, for instance by selecting different time ranges, asking for extra information of specific time points in a tooltip, or switching between the available variables. This service is used to monitor the operational status of sensors at different time scales to find

Fig. 5. Interaction with a line chart showing the average of the luminosity values recorded by a streetlight during half a day

regular patterns (e.g. day- and night-time alternation) as well as possible anomalies (e.g. missing values due to communication errors or irregular operational conditions). Fig. 5 shows an example of such interaction. The selected streetlight was turned on during the night at a minimum lighting value, to guarantee visibility without consuming too much energy. Towards the sunrise (at 07:00 h), the light was automatically switched off because of the illumination environmental conditions. During the day, the pole was finally turned on at its maximum power, possibly to test the operational conditions of the lamp. On the other hand, Fig. 6 shows the average of the luminosity values recorded by a streetlight in the period between December 28, 2021 and January 10, 2022. It is possible to see the regular pattern between the day- and night-time luminosity levels, respectively the peaks and valleys of the line chart. However, some irregularity can be detected in the days immediately after December 31, where it seems that any data have not been recorded during approximately a week, since two consecutive valleys are connected to each other. This chart highlights a possible problem either on the sensor or the network side that should be further investigated to prevent future service breakdowns.

## IV. CONCLUSIONS AND FUTURE WORK

In this paper, the implementation of a Digital Twin to optimize the management of a smart, outdoor lighting system is presented. The main architectural components of the DT are described, with an emphasis on the development of a Virtual Reality environment as the main, bi-directional interface between the real and virtual worlds. The paper showcases practical examples in which the VR environment is particularly suitable to understand the operational context of the smart lighting management.

Although preliminary tests show a proper work of the system and a quick synchronization between actions taken in the virtual world on its physical counterpart (or vice versa), further experimental validation is needed to characterize the timing behavior, especially in terms of end-to-end latency and error rate. This characterization will help to identify and solve the possible bottlenecks that may appear in the system involving different components and technologies.

On the other hand, as aforementioned, the use of DT systems for smart street lighting systems opens a range of new features and uses. One straightforward application may be the use of the presence detection data to identify the most common pathways followed by campus users. To this end, the

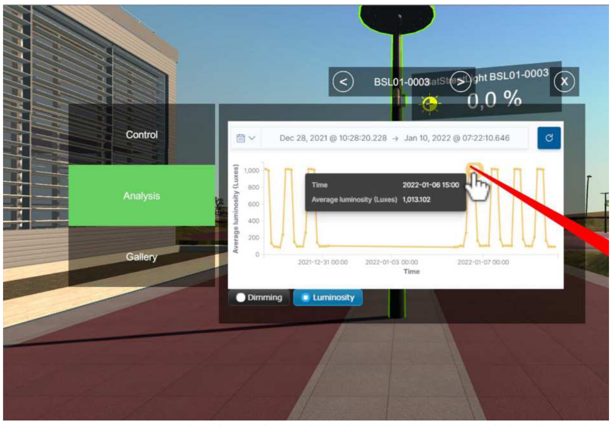


Fig. 6. Interaction with a line chart showing the average of the dimming values recorded by a streetlight during almost two weeks

VR application may recreate the occurrence of these pathways by superimposing the trajectories as semi-transparent, segmented lines to visually appreciate trends and deviations. Besides, such representation may help identify possible locations where to install new streetlights in order to cover a wider area of interest and increase at the same time the feeling of safety in campus users. In order to improve the accuracy of pedestrian detection and avoid false positives, such as those caused by tree branch sway or animals, extra information may be acquired by processing video recordings of artificial vision cameras.

Within the Tr@nsnet project, CEDINT-UPM is deploying several bio-acoustic sensors and camera traps to monitor and analyze how campus biodiversity is affected by the activity of its users (students, staff, public and private transport) and, of course, lighting. The data obtained by integrating such devices into the streetlight DT system could help to identify if the switching of the streetlights, their dimming level or even the light temperature color, may contribute to modify wildlife behavior, including birds, small mammals, insects, and especially, bats.

Advanced visualization functionalities are required to support a more informed and smarter management of the whole lighting system. Therefore, dashboards embedding visual analysis capabilities to explore relationships between variables will be included to support tasks such as preventive detection of operational anomalies and identification of hot spots for a more efficient spatial distribution of the lamps. In this sense, the virtual environment is expected to improve the actual charting expressiveness and implement more natural and faster interaction interfaces (e.g. eye or finger tracking).

Finally, new graphical techniques should be considered to improve the visual rendering of the photometric behavior of the light in outdoor environments, in order to have more precise feedback on the intended combined effects of the light distribution, including the use of LED temperature color.

## REFERENCES

[1] G. Zissis, P. Dupuis, L. Canale, and N. Pigenet. "Smart lighting systems for smart cities". In Holistic Approach for Decision Making Towards Designing Smart Cities, Lazaroiu, G. C., Roscia, M., & Dancu, V. S. editors, Springer, Cham, 2021, p. 75-92

[2] J. Bennie, J.P. Duffy, T.W. Davies, M.E. Correa-Cano, K.J. Gaston. "Global Trends in Exposure to Light Pollution in Natural Terrestrial Ecosystems". Remote Sens. 2015, vol. 7, pp. 2715-2730. https://doi.org/10.3390/rs70302715

[3] P. Chamoso, A. González-Briones, S. Rodríguez, and J.M. Corchado. "Tendencies of technologies and platforms in smart cities: a state-of-the-art review". Wireless Communications and Mobile Computing, 2018.

[4] K. Brock, E. den Ouden, K. van der Klauw, K. Podoynitsyna, and F. Langerak, "Light the way for smart cities: Lessons from Philips Lighting". Technological Forecasting and Social Change, 2019, v. 142, pp. 194-209.

[5] M. Grieves. "Digital Twin: manufacturing excellence through virtual factory replication: A whitepaper", 2014, p. 1-7.

[6] F. Tao, and M. Zhang. "Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing". IEEE Access, 2017, vol. 5, pp. 20418-20427.

[7] F. Jiang, L. Ma, T. Broyd, and K. Chen. "Digital twin and its implementations in the civil engineering sector". Automation in Construction, 2021, v. 130, 103838.

[8] S. Talari, M. Shafie-Khah, P. Siano, V. Loia, A. Tommasetti, and J.P. Catalão. "A review of smart cities based on the internet of things concept". Energies, 2017, 10(4), 421.

[9] S. Ivanov, K. Nikolskaya, G. Radchenko, L. Sokolinsky, and M. Zymbler. "Digital Twin of City: Concept Overview". In 2020 Global Smart Industry Conference (GloSIC), IEEE, November 2020, pp. 178-186)

[10] G. Martin, A. Poppe, S. Schöps, E. Kraker, C. Marty, W. Soer, and J.Yu. "AI-TWILIGHT: AI-digital TWIn for LIGHTing–a new European project". In 2021 27th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC), IEEE, September 2021, pp. 1-6

[11] J. Jerald. The VR book: Human-centered design for virtual reality. Morgan & Claypool, 2015

[12] M. Scorpio, R. Laffi, R., M. Masullo, G. Ciampi, A. Rosato, L. Maffei, L. and S. Sibilio. "Virtual Reality for Smart Urban Lighting Design: Review, Applications and Opportunities". Energies, 2020, 13(15), 3809

[13] R. Krupiński. "Virtual reality system and scientific visualisation for smart designing and evaluating of lighting". Energies, 2020, 13(20), 5518

[14] S. Calatrava, G. Cañada, C. Jiménez, E. Saavedra, G. del Campo and A. Santamaría. "Smart Street Lightning System for Eco-Cities Based on 6LoWPAN IoT Networks". Poster presented at: TR@NSENER Workshop on: Scenarios and perspectives for the Energy Efficiency in the SUDOE Region for 2050, 2018, Toulouse, France.

[15] G. Calderon, G. del Campo, E. Saavedra, and A. Santamaria. "Management and Monitoring IoT Networks through an Elastic Stack-based Platform". In 2021 8th International Conference on Future Internet of Things and Cloud (FiCloud), IEEE, August 2021, pp. 184-191. https://doi.org/10.1109/FiCloud49777.2021.00034

[16] C. Jimenez, E. Saavedra, G. del Campo, and A. Santamaria. "Alexa-based voice assistant for smart home applications". IEEE Potentials, 2021, 40(4), pp. 31-38. https://doi.org/10.1109/MPOT.2020.3002526.

2021 8th International Conference on Future Internet of Things and Cloud (FiCloud)

2021 8th International Conference on Future Internet of Things and Cloud (FiCloud) | 978-1-6654-2574-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/FICLOUD49777.2021.00034

# Management and Monitoring IoT Networks through an Elastic Stack-based Platform

Gonzalo Calderon, Guillermo del Campo, Edgar Saavedra, Asuncion Santamaria
*CeDInt-UPM*
*Universidad Politécnica de Madrid*
Pozuelo de Alarcón, Spain
{gcalderon, gcampo, esaavedra, asun}@cedint.upm.es

***Abstract*—With the increase of IoT deployments and their complexity, both management and maintenance are becoming challenging tasks. With the aim of easing the detection and anticipation of potential issues, we propose an IoT platform combining Elastic Stack tools (Elasticsearch, Kibana and Beats) and Apache Kafka. The platform, based on a distributed architecture and data replication, provides scalability and performance to process, store, and visualize data in real-time. Besides, it allows communication between users and IoT devices, and integrates different metric agents to monitor performance and consistency. Deployment in three different use cases and experimental evaluation shows the suitability of our approach for IoT heterogeneous applications and services.**

***Index Terms*—IoT platform, Elastic Stack, Elasticsearch, Kibana, Beats**

## I. INTRODUCTION

Nowadays, IoT is a relevant and necessary technological sector for the efficient management of resources in multiple environments. Despite the concern for security and privacy [1], more and more people, companies, factories or cities are betting on deploying IoT networks of IoT devices to improve the management of their resources, save costs or increase their productivity [2]. In fact, the International Data Corporation (IDC) forecasts that total spending on IoT solutions will continue to grow and Huawei expects 100 billion IoT connections by 2025 [3]. Besides, the COVID-19 pandemic has impacted on the social and financial needs, increasing the value of the IoT applications and impacting on the market trends [4], [5].

The continuous development of IoT solutions and their adaptation to different sectors has led to the emergence of heterogeneous architectures, networks, devices, protocols, middlewares, data models, and applications [6]. IoT networks generate large amounts of data at different levels, so platforms are required to process, store, and manage them efficiently. In this way, platforms should anticipate the growth and diversity of IoT components and data, and provide enough scalability and performance to manage and process them.

In this work, the implementation goes a step further, not only providing the essential functionalities to manage the IoT networks, but also integrating modules to centralize the maintenance of the platform. The management and maintenance of distributed architectures require complex mechanisms to detect and anticipate potential errors in its modules. To carry out this work, we propose a solution that centralizes the storage, management, and visualization to monitor the systems and services of the platform. Thanks to this contribution, we monitor the status and the performance of the platform in real-time.

This work tackles the challenge of implementing a distributed and replicated IoT platform using Elastic Stack tools, providing fully configurability to be adapted to different environments and use cases. The platform processes and sends the data from the IoT networks through Apache Kafka before storing them in Elasticsearch. Availability and consistency of the stored data is ensured by the distribution and replication of specialized services across multiple servers that form a cluster.

This paper is structured as follows: In section 2 we review the differences between our proposed platform from the others that are available in the literature. Section 3 presents an introduction to Elastic Stack. Section 4 gives an overview of the platform, its funtional and non-functional requirements. In Section 5, we describe the implementation of the IoT platform with a detailed explanation of their modules. In Section 6, we report the experimental results to evaluate the performance in real use cases and future improvements. Section 7 includes the conclusions.

## II. RELATED WORK

Traditionally, IoT platforms have been deployed for specific uses and hosted on monolithic and vertically integrated systems [7]. However, this management approach is inefficient and slow because if the workload is not distributed across multiple services, the platform does not have enough resources to process all data. Additionally, in order to detect technical and performance issues, an IoT platform should collect metrics and operational data from the modules that compose it. This process, which is called monitoring, allows the platform administrator to know the status of the networks and the availability of services and servers in real-time. There are some open-source solutions for monitoring, such as Nagios, which offers the possibility of monitoring hardware and software [8], or Zabbix, which is mainly used to supervise communication networks [9].

978-1-6654-2574-2/21/$31.00 ©2021 IEEE
DOI 10.1109/FiCloud49777.2021.00034

An open-source alternative tool is Elastic Stack (Elasticsearch, Kibana and Beats), which provides the possibility of unifying managing, monitoring, visualising, and storing functionalities [10]. The most common use of Elastic Stack tools is Security Information and Event Management (SIEM) [11], though, during the last few years has been expanded to other use cases such as IoT [12], [13]. With Elastic Stack tools, different IoT data hubs have already been developed to store the data in smart cities or buildings [14], [15], [16]. These solutions ingest the data through Logstash, which requires reading an output file or making HTTP requests to an external service to get the data. Instead, we present a novel contribution by using Apache Kafka for data exchange between the IoT network and the database, facilitating the platform configuration and management processes.

## III. ELASTIC STACK

Elastic Stack is a set of open-source tools developed by the Elastic company that unifies the tools to store, monitor, analyse, and visualize in real-time the data generated by servers, systems, or applications. The tools of Elastic stack that are used in the designed IoT platform are:

### A. *Elasticsearch*

It is a real-time distributed search and analytics engine for all types of data. Elasticsearch organizes the information in distributed documents that contain fields structured in key-value pairs. These documents are indexed in real-time in a data structure called index, without defining previously a structure for fields. Being a distributed search engine, it balances the load across nodes providing high availability and scalability: as the data and query volume grows, it automatically distributes the data across the available nodes.

### B. *Beats*

It is a collection of agents that collects operational data from different sources and forward them to Elasticsearch for its analysis and display. There are seven different Beats agents, but in this project, we only use four of them:

*1) Filebeat:* it is designed to read any text file from a system or application. We use Filebeat to monitor and collect log events.

*2) Heartbeat:* it is a lightweight agent for monitoring uptime by pinging a list of services. We use Heartbeat to check services' availability.

*3) Metricbeat:* it collects metrics from the operating system and applications. We use Metricbeat to collect metrics and statistics from services and servers.

*4) Packetbeat:* it is a real-time network packet analyser that monitors applications and services to evaluate their performance. We use Packetbeat to analyse the network performance by means of metrics such as latency, response times, accesses, errors, or transactions.

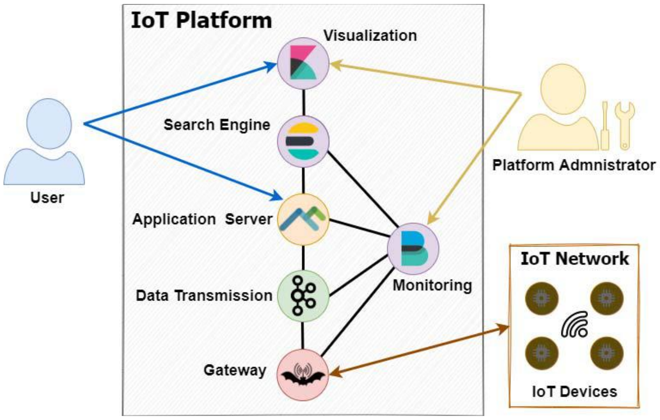


Fig. 1: System description

### C. *Kibana*

It is an upper layer of Elasticsearch intended to manage, explore, visualize, and display the data in real-time. Kibana includes a wide variety of plugins and tools that allow users to search, view, and analyse the indexed data.

## IV. SYSTEM REQUIREMENTS

The purpose of our IoT platform, which is conceptually described in Fig. 1, is to store and manage in real-time data coming from IoT devices and networks. A Gateway collects the measurements from the IoT Devices and sends them to the Application Server through the Data Transmission Module. The Application Server allows acting on the devices through a graphical interface and sends the measurements to Elasticsearch to be stored. Beats agents, deployed in all the modules of the platform, collect metrics, logs, and network traffic statistics and send them to Elasticsearch. Finally, this collected data is displayed on dashboards using the Kibana tools.

### A. *Functional requirements*

The IoT platform developed with Elastic Stack should meet the following requirements in terms of functionality and services to manage and monitor IoT networks.

*1) Store the measurements of the devices:* the platform should store the measurements obtained from the IoT devices in Elasticsearch in a structured way.

*2) Manage and act on the devices:* the platform should allow users to have limited and controlled access to the search engine to visualize the measurements, and act on the devices through a web interface.

*3) Create visualizations with the measurements:* the platform should integrate a module to create visualizations and dashboards based on the stored measurements.

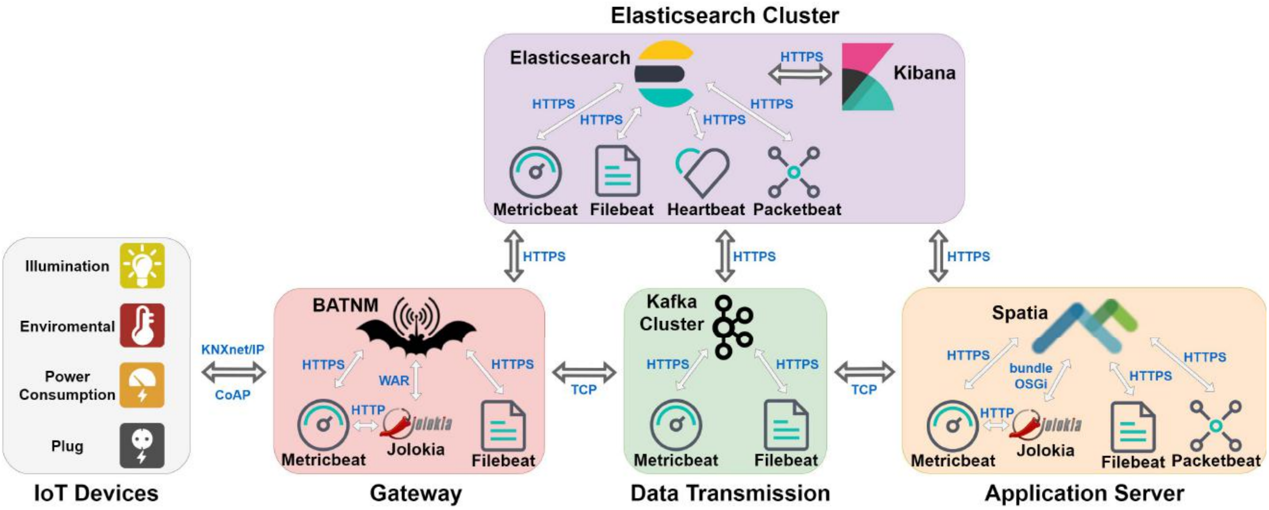


Fig. 2: IoT Platform Architecture

*4) Collect metrics from servers and services:* a monitoring system should be implemented to assess the operation and performance of platform, enabling the administrator to adjust the resources.

*5) Centralize log data of the platform:* the collection and storage of the logs should be implemented to facilitate the development and the maintenance of the platform.

*6) Check the status of the services:* the visual checking of the IoT platform should be implemented, providing quick and efficient verification of the status of the entire platform in real-time.

*7) Capture inbound network traffic on the platform:* the platform should implement a network monitoring system to supervise traffic (latency, response times) and protect against undesired access (it is a public and accessible through a web application).

## V. IoT PLATFORM IMPLEMENTATION

Fig 2 shows the IoT platform architecture, which is divided into five modules: IoT Devices and Networks, Gateway, Data Transmission, Application Server, and Elasticsearch Cluster.

Although the five modules are provided with Metricbeat and Filebeat specific agents, there are some common metrics and logs that are collected in all the modules.

Metricbeat common modules:

- *System metrics*: they collect every 5 minutes to monitor the status of each server.
- *Beat metrics*: they monitor every 10 seconds the Beats agents deployed on the system.

Filebeat common modules:

- *System logs*: they collect the syslog and authlog files created by the Unix/Linux logging service.
- *Beat logs*: they collect logs of any Beat agent deployed on the system.

### A. IoT Devices/Networks

Data from different devices, including energy meters, ambient sensors, light dimmers, or blind controllers are collected using both wired and wireless networks. The platform integrates KNX, a wired technology that works over Ethernet and communicates with the gateway through a KNXnet/IP protocol [17]; and BatNet, a 6LoWPAN-based wireless communication technology [18].

### B. Gateway

The Gateway Module manages IoT networks through specific drivers to communicate with different IoT technologies: KNX and BatNet. Besides, it transfers data and functionalities from the Application Server to the IoT Devices and vice versa. It comprises the following components: Bat Network Manager (BatNM), Jolokia, Metricbeat, and Filebeat.

*1) BatNM:* it is a self-developed web application in Eclipse Jetty with a Model View Controller (MVC), which runs in a Java Virtual Machine (JVM). The communication between BatNM and IoT Devices is done by Constrained Application Protocol (CoAP) [19].

*2) Jolokia:* it is an open-source project that provides a JMX-HTTP bridge as an alternative to JSR-160 connectors [20]. It makes get requests to the JVM in which BatNM is running to know its performance.

*3) Metricbeat:* it collects and adapts the performance data and statistics obtained from Jolokia to send them to Elasticsearch.

*4) Filebeat:* it organizes and sends the log files generated by log4j, logging library for Java [21], from BatNM to Elasticsearch. A new event log is generated every time that an event occurs in the Producer or Consumer of Kafka.

### C. Data Transmission

The Data Transmission Module is designed to provide the communication between the Gateway and the Application Server, and vice versa. The modules implemented are: Apache Kafka Cluster, Metricbeat, and Filebeat.

*1) Apache Kafka:* it is an open-source distributed, partitioned and replicated stream platform that effectively process data in real-time [22]. Apache Kafka follows a publisher/consumer approach, in which a producer publishes data in a message queue, and the consumer subscribes to the streams of records. The data published in the queue is stored in a persistent way, providing high throughput, low latency and fault-tolerance.

*2) Metricbeat:* it is configured to send every 10 seconds partition and consumer group metrics from the Apache Kafka cluster to Elasticsearch.

*3) Filebeat:* it organizes and sends the log files from Apache Kafka to centralize them in Elasticsearch.

### D. Application Server

The Application Server Module ensures the communication between Elasticsearch and the Gateway Module by: a) creating a data model to store the measurements; and b) integrating a web user interface based on a Representational State Transfer (RESTful) framework that allows the communication between the users and the application. The modules of the implemented application server are Spatia, Jolokia, Metricbeat, Filebeat, and Packetbeat.

*1) Spatia:* it is a self-developed Java application accessible through a public IP that allows users to manage and act easily on their IoT devices and networks. User registration is required to guarantee security to data access. After the devices are discovery and linking, users are able to access the current values, view and download historical measurements, and interact with their IoT devices. Spatia has been deployed in an Apache Karaf container enabling the installation of a new application at any time.

*2) Jolokia:* it makes get requests to the JVM in which Spatia is running to know its performance.

*3) Metricbeat:* it collects and adapts the performance data and statistics obtained from Jolokia to send them to Elasticsearch.

*4) Filebeat:* it organizes and sends the log files from Spatia to Elasticsearch. A new event log is generated every time that an event occurs in the Producer or Consumer of Kafka.

*5) Packetbeat:* it captures and monitors the network traffic of Spatia application web to ensure high levels of performance and security.

### E. Elasticsearch Cluster

The Elasticsearch Cluster Module is composed by six elements: Elasticsearch, Metricbeat, Filebeat, Packetbeat, Heartbeat, and Kibana.

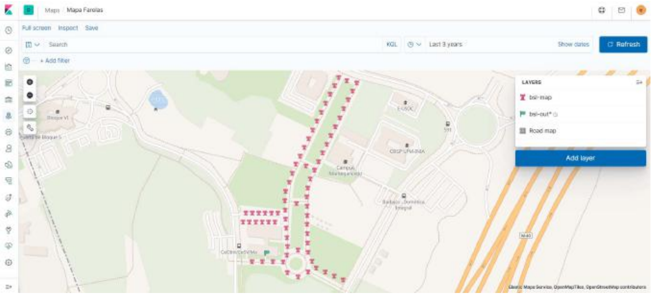


Fig. 3: Map visualization for Smart Street Lighting use case

*1) Elasticsearch:* we have developed a cluster of Elasticsearch parallel nodes to distribute the workload, offering horizontal scalability. Elasticsearch clustering allows the IoT platform to have enough processing and storing capabilities to index large volumes of data generated by the IoT devices and Beats agents. To ensure the availability and the consistency of the data, we configure an adaptive index mapping: if the data comes from a measurement, the platform replicates it in another cluster node; if the data comes from a Beats agent, there is not replication because losing them is not critical for the system. In this way, a balance between consistency and performance in the cluster is achieved.

*2) Metricbeat:* two Metricbeat agents have been added to the platform to monitor the cluster:

- *Elasticsearch metrics*: it monitors the state of the cluster every 10 seconds.
- *Kibana metrics*: it collects the statistics from Kibana every 10 seconds.

*3) Filebeat:* two Filebeat agents have been added to collect the cluster log files:

- *Elasticsearch logs*: it sends the logs file from Elasticsearch.
- *Kibana logs*: it sends the logs file from Kibana.

*4) Packetbeat:* it monitors the network traffic of Kibana user interface to ensure high levels of cluster performance and security.

*5) Heartbeat:* it checks the availability of the IoT platform services every 30 seconds, requesting their status and sending it to Elasticsearch.

*6) Kibana:* we have configured the Kibana interface to display different types of data: platform users, sensor measurements and Beats metrics:

- *Discover*: it displays the stored data organized into indexes on a time axis. In this tool, the user can make queries, apply filters, and select the period of time used as a temporal axis for the displayed values.
- *Visualize and Dashboard*: it allows users to create different types of graphs, plots, or tables adapted to the nature of the indexed data.
- *Maps*: it displays the data obtained from Packetbeat to visualize the inbound network traffic in a personalized

(a) Coordinating node (b) Multi-purpose nodes (c) Data nodes

Fig. 4: Elasticsearch nodes performance

map with multiple layers. Additionally, we have created another visualization to display the devices by their geographical coordinates, previously configured in Spatia. As an example, Fig. 3 shows the location of the IoT devices installed at the Smart Street Lighting use case.

- *Metrics*: it displays the data obtained from Metricbeat. In this way, the platform administrator can visualize in detail the metrics of each server or service.
- *Logs*: it displays the data obtained from Filebeat. In this manner, the platform administrator can visualize the log files of each system of the platform.
- *Uptime*: it displays the data obtained from Heartbeat. Thus, the platform administrator can check the status of services of the platform.
- *Stack Monitoring*: it shows the overall state of the cluster and its deployed services.

## VI. EXPERIMENTAL RESULTS

### A. *Real use cases*

After the design and development phases, we have deployed the IoT platform in three different real use cases, located at the Campus de Montancedo from Universidad Politécnica de Madrid. The platform has been under production from June 2020, increasing the reliability of the implementation.

*1) Smart Building:* the objective of this use case is to monitor the energy behaviour of the CeDInt-UPM building. It comprises three different IoT networks with 21 ambient sensor devices, 34 smart meters, 4 blind controllers, 10 light dimmers, and a weather station.

*2) Smart Lighting:* the goal of this use case is to optimize the outdoor smart street lighting system. There are 69 light dimming devices that, apart from controlling light level, collect luminosity, presence detection, and alarms notifications.

*3) Smart Greenhouse:* the objective of this use case is to decrease energy consumption in the facilities of the Centro de Biotecnología y Genómica de Plantas (CBGP). There are 10 ambient sensor devices, 23 power meters, and 25 LED dimmers.

### B. *Experimental Setup*

In order to have enough resources to index large volumes of data and manage the measurements, metrics, and operational data, we have designed an Elasticsearch cluster of 12 nodes. With the aim of improving the performance of the cluster, we configure the nodes to play specific roles: coordination, multi-purpose, and data storage. The analysis of this configuration is presented in Fig. 4 where we show the performance of the cluster depending on the type of node.

- *1 Coordinating node*: it acts as orchestrator that balances traffic across the cluster by deciding which node should make each specific task. As it is the only node in the cluster in which the data stream does not flow before being indexed, it also hosts Kibana. This node should have enough RAM memory to make queries in Kibana smoothly. We assume that 6 GB dedicating 3 GB to the JVM heap is adequate. Fig. 4(a) represents that the use of JVM Heap fluctuates between 2 GB and 900 MB, caused by the process of updating the last value coming from the IoT devices. However, it does not need much CPU, barely reaching the 4% of 2 dedicated CPUs because it only processes data when accessing Kibana. It only has incoming and outgoing (I/O) traffic occasionally: when users or administrators performs a search on the platform using Kibana. It does not need too much storage but it has been deployed with 100 GB to install the Beats agents on it.

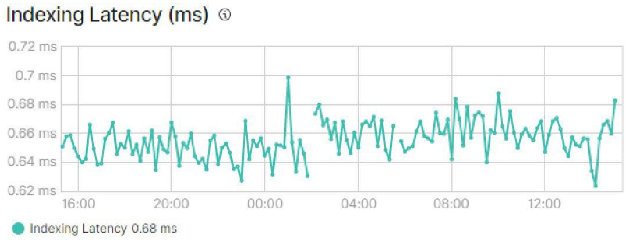


Fig. 5: Indexing Latency

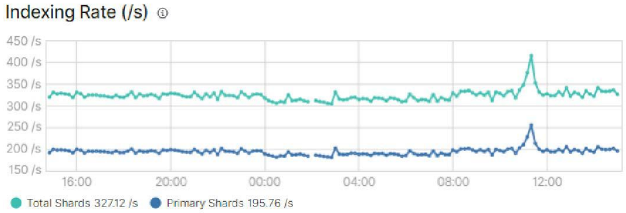


Fig. 6: Indexing Rate

- *5 multi-purpose nodes*: they simultaneously play different roles: master-eligible (tracking and mapping), data (storage), ingest (pre-processing), machine learning (running ML jobs), and transform (adaptation). We have selected an odd number of master-eligible nodes to avoid split-brain issues. These tasks need a lot of CPU so 2 fixed CPUs have been dedicated to these nodes. Fig. 4(b) shows that the 2 CPUs dedicated to these nodes are working at full capacity, close to 90%. Due to their great workload, each node has been deployed with 6 GB of RAM memory dedicating 3 GB to the JVM heap. It can be observed how the use of the JVM Heap is almost constant, close to 2.3 GB. Their I/O traffic is low except when they have to re-allocate a shard in the cluster. The storage of each node has been reduced to 70 GB to avoid the overloading of the nodes and slowing down its operation.
- *6 data nodes*: these nodes act as a data warehouse, that is, they only have the function of storing the information in the cluster. For this reason, they do not require so much RAM and 4 GB has been assigned, dedicating 2 GB to the JVM heap. As it can be seen in Fig. 4(c), this RAM is enough because it uses more or less 50% of its capacity. On the other hand, the use of the 2 dedicated CPUs is significant because they use it to process and store the data. It fluctuates between 60% and 95% and sometimes reaches the peak of 100%. Data nodes are the ones with the most I/O traffic since they are responsible for indexing the data. Likewise, they need more storage, so each node is deployed with 500 GB.

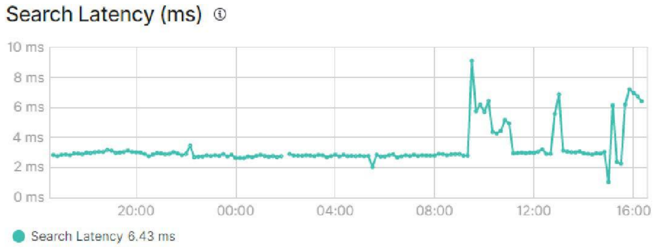


Fig. 7: Search Latency

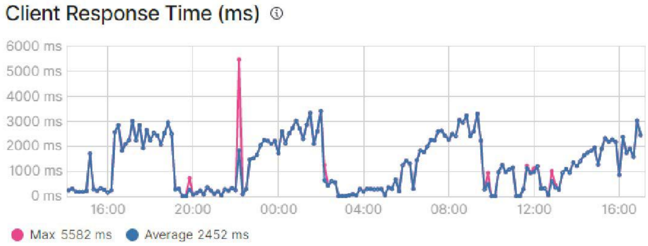


Fig. 8: Client Response Time

The Apache Kafka cluster integrated in the Data Transmission Module has 3 brokers. The resources of each broker is configured based on the requirements: 80 GB to store the messages produced during the established retention policy of 5 days, 4 GB of RAM memory and 2 CPUs to be used dynamically depending on the workload.

Spatia and BatNM use differents JVMs within virtual servers. Each server has been deployed with 3 GB of RAM memory, 50 GB of storage and 2 CPUs.

### C. *Performance evaluation*

To validate the cluster performance, we have measured different parameters related to data access operations.

- *Index latency*: it is the time taken by the cluster for indexing a single document in a shard. In our cluster, the average index latency is 0.68 ms (see Fig. 5), low enough to guarantee a quick access to information.
- *Indexing rate*: it is the number of documents being indexed per second. In the platform, it is approximately 200 documents/second of the primary shards (see Fig. 6). The indexing rate for total shards is less than double because just the data coming from IoT devices are replicated.
- *Searching latency*: it is the time taken by the cluster for searching in a shard. In our cluster, the average searching latency is 6.43 ms, ensuring quickness in searches made by users (see Fig. 7).
- *Client response time*: it is the response time for client requests to the Kibana instance. In our platform, the average response time is 2452 ms (see Fig. 8). Response time of 2.5 seconds ensures that the platform allows the user to search, manage, and act on its devices smoothly.

TABLE I: Documents indexed

| | |
|---|---|
| Measurements | 1.216.686.996 |
| Filebeat | 1.147.835.808 |
| Metricbeat | 795.206.138 |
| Packetbeat | 380.818.643 |
| Heartbeat | 42.792.261 |
| **Total documents** | **3.583.339.846** |
| **Total data** | **1 TB** |

TABLE II: Elastic Stack tools

| | |
|---|---|
| Elasticsearch nodes | 12 |
| Kibana instances | 1 |
| Filebeat agents | 20 |
| Metricbeat agents | 20 |
| Packetbeat agents | 2 |
| Heartbeat agents | 1 |

TABLE III: Performance evaluation

| | |
|---|---|
| **Indexing Latency** | 0.68 ms |
| **Indexing Rate** | 327.12 /s |
| **Search Latency** | 6.43 ms |
| **Client Response Time** | 2452 ms |

### *D. Discussion*

The workload of the platform is practically the same every day, but it can suffer variations within the day hours. Table I presents the total documents stored by the IoT platform. The Elasticsearch cluster indexes more than 3 billion documents that are structured in 250 indexes and require more than 1 TB of storage.

Table II summarizes the Elastic Stack tools configured in the IoT platform: 12 Elasticsearch nodes, 43 Beat agents, and an instance of Kibana. The Elasticsearch cluster has been storing data since September 2018 while Beat agent metrics were introduced in June 2020. It can be observed how, though Beat metrics are relatively recent, they are major contributor to the amount of indexed documents. Therefore, we could improve performance and reduce storage by lowering the frequency of Beat agents metrics.

As we have seen in the performance experimental evaluation, summarized in Table III, we achieve a balance between consistency and performance by configuring replication factor and shards of Elasticsearch index. In this way, users and platform administrators can manage, visualize, and act on the IoT network quickly and smoothly. In order to achieve better performance results, we are working on different strategies, such as configuring an internal setup of Elasticsearch nodes or adjusting specific features of the system, e.g. JVM or the kernel.

The platform is deployed in the CeDInt-UPM data centre, however, we are migrating it to the cloud using Docker. Running the platform in Docker containers will facilitate its distribution to deploy it anywhere using the cloud. When the platform is fully migrated to the cloud, we will evaluate its performance and compare it with the local implementation.

## VII. CONCLUSIONS

In this work, we present the development and deployment of an Elastic Stack based IoT platform for managing and monitoring IoT devices and networks. The platform, which is novel at the combined used of Elastic Stack and Apache Kafka for IoT applications, has been tested in three different real use cases.

The IoT platform provides scalability and performance to process, store and manage large volumes of data in real-time. These data comes from IoT networks, log events, metrics, statistics, services' availability, and network traffic to manage, maintain, and evaluate the platform. We centralize and visualize this collected data using the open-source tools provided by Elastic Stack.

We have configured the Elasticsearch cluster to develop a distributed and replicated architecture that guarantees the availability and consistency of the stored data. We have developed an infrastructure based on Beats agents to facilitate the supervision of the modules operation by the platform administrator. The functionalities offered by Kibana have been customized to display the data of the platform users, IoT devices dashboards, and maps, and the data collected by Beats agents. As a result, management and maintenance tasks are much more friendly and accessible.

## ACKNOWLEDGMENT

This work is part of the CHIST-ERA research project "ABIDI: Context-aware and Veracious Big Data Analytics for Industrial IoT" and was funded by the State Research Agency (AEI) from the Ministerio de Ciencia e Innovación (MICINN) of Spain, grant number PCI2019-103762.

This work has been selected as a 2020 EMEA Elastic Search Award for the innovation of applying Elasticsearch to manage an IoT platform.

## REFERENCES

[1] J. C. Talwana and H. J. Hua, "Smart World of Internet of Things (IoT) and its Security Concerns," in *IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Dec. 2016, pp. 240–245.

[2] F. Dahlqvist, M. Patel, A. Rajko, and J. Shulman, "Growing Opportunities in the Internet of Things," *McKinsey*, Jul. 2019.

[3] H. Espinoza, G. Kling, F. McGroarty, M. O'Mahony, and X. Ziouvelou, "Estimating the Impact of the Internet of Things on Productivity in Europe," *Heliyon*, vol. 6, no. 5, May. 2020.

[4] R. P. Singh, M. Javaid, A. Haleem, and R. Suman, "Internet of Things (IoT) Applications to Fight Against COVID-19 Pandemic," *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, vol. 14, no. 4, pp. 521–524, Jul. 2020.

[5] M. Nasajpour, S. Pouriyeh, R. M. Parizi, M. Dorodchi, M. Valero, and H. R. Arabnia, "Internet of Things for Current COVID-19 and Future Pandemics: An Exploratory Study," *Journal of healthcare informatics research*, pp. 1–40, Nov. 2020.

[6] A. Kazmi, Z. Jan, A. Zappa, and M. Serrano, "Overcoming the Heterogeneity in the Internet of Things for Smart Cities," in *International workshop on interoperability and open-source solutions*, Apr. 2017, pp. 20–35.

[7] T. Suganuma, T. Oide, S. Kitagami, K. Sugawara, and N. Shiratori, “Multiagent-based Flexible Edge Computing Architecture for IoT,” *IEEE Network*, vol. 32, no. 1, pp. 16–23, Jan. 2018.

[8] W. Barth, *Nagios: System and Network Monitoring*. No Starch Press, 2008.

[9] R. Olups, *Zabbix Network Monitoring*. Packt Publishing Ltd, 2016.

[10] S. Chhajed, *Learning ELK stack*. Packt Publishing Ltd, 2015.

[11] I. Kotenko, A. Kuleshov, and I. Ushakov, “Aggregation of Elastic Stack Instruments for Collecting, Storing and Processing of Security Information and Events,” in *IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug. 2017, pp. 1–8.

[12] D. Laffey, “Paid Search: The Innovation that Changed the Web,” *Business Horizons*, vol. 50, no. 3, pp. 211–218, May. 2007.

[13] G. J. Myatt, *Making Sense of Data: a Practical Guide to Exploratory Data Analysis and Data Mining*. John Wiley & Sons, 2007.

[14] M. Bajer, “Building an IoT Data Hub with Elasticsearch, Logstash and Kibana,” in *IEEE 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, Aug. 2017, pp. 63–68.

[15] A. Talaş, F. Pop, and G. Neagu, “Elastic Stack in Action for Smart Cities: Making Sense of Big Data,” in *13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, Sep. 2017, pp. 469–476.

[16] S. Dharur and K. Swaminathan, “Efficient Surveillance and Monitoring using the ELK Stack for IoT powered Smart Buildings,” in *IEEE 2nd International Conference on Inventive Systems and Control (ICISC)*, Jan. 2018, pp. 700–705.

[17] H.-J. Langels, “Knx ip using ip networks as knx medium,” in *Proceedings of the KNX Scientific Conference 2008*, 2008.

[18] G. Del Campo, E. Montoya, J. Martín, I. Gómez, and A. Santamaría, “Batnet: a 6lowpan-based sensors and actuators network,” in *International Conference on Ubiquitous Computing and Ambient Intelligence*. Springer, 2012, pp. 58–65.

[19] C. Bormann, A. P. Castellani, and Z. Shelby, “CoAP: An Application Protocol for Billions of Tiny Internet Nodes,” *IEEE Internet Computing*, vol. 16, no. 2, pp. 62–67, Mar. 2012.

[20] “Jolokia jmx on capsaicin,” https://jolokia.org/index.html, accessed: 2021-03-10.

[21] C. Gülcü, *The complete log4j manual*. QOS. ch, 2003.

[22] B. R. Hiraman *et al.*, “A Study of Apache Kafka in Big Data Stream Processing,” in *IEEE International Conference on Information, Communication, Engineering and Technology (ICICET)*, Aug. 2018, pp. 1–3.

# A Novel, Self-Powered, Non-Intrusive, Sigfox-Enabled Smart Meter for Challenging Scenarios

Edgar Saavedra
*CeDInt-UPM*
*Universidad Politecnica de Madrid*
Madrid, Spain
e.saavedra@upm.es

Guillermo del Campo
*CeDInt-UPM*
*Universidad Politecnica de Madrid*
Madrid, Spain
gcampo@cedint.upm.es

Asuncion Santamaria
*CeDInt-UPM*
*Universidad Politecnica de Madrid*
Madrid, Spain
asun.santamaria@upm.es

***Abstract*— In this work, a wireless autonomous power smart metering device is presented. This device is able to measure current at mains electrical lines, exploiting the magnetic field inducted around them as energy harvesting. Therefore, it does not need a dedicated power supply or battery replacement. The electromotive force inducted shall be properly handled to make it suitable to be stored and retrieved from an energy storage element. Since very low power is harvested from the mains magnetic field, a good treatment for the incoming power, as well as a minimal power consumption system are vital. In order to make the system autonomous, wireless communication reliability is a crucial factor. Sigfox is the selected technology as it offers almost out-of-the-box functioning and wide coverage. Proven results based on a constant 15-minute metering period obtaining infinite lifetime are shown, providing an optimal solution for challenging scenarios such as rural areas or developing countries.**

***Keywords*— Energy Harvesting, Smart Meter, Sigfox, Self-Powering, Autonomous Device, Energy Saving, Internet of Things**

## I. INTRODUCTION

Worldwide electricity consumption is constantly growing: the global increase from 2017 to 2018 was 3.5%, and it has more than doubled from 1990 [1]. Although more trust is put on renewable energies, fuel-fossil-dependant energy generation is still a crucial contributor to global pollution. Whereas USA and EU have been reducing carbon emissions since the 2000s —both for energy and goods production—, emerging economies such as China and India present an increasing $CO_2$ contribution trend [2]. To overcome this global issue, there are different measures such as the use of renewable energies, smart energy consumption analysis to reduce usage, ambient-aware systems, smart grids or on-demand response policies. Among them, and focusing on emerging and developing countries, smart metering seems to be the most suitable approach. Apart from being the less cost-demanding, it may help to learn about inhabitants' energy habits, hence possibly reducing consumption and targeting on energy drains.

Regarding smart meters market, there are several options working in different ways. Some smart meters are those installed by electric companies to remotely acquire consumption data, and therefore bill accordingly. Others are powered directly by the mains line with a transformer, such as those encountered in [3, 4]. These sorts of devices intrude electrical installations, meaning that a specialist is needed for installation and the building power supply might be interrupted temporally. On the other hand, they have the advantage of also metering voltage; thus, reckoning power factor. However, in many cases —especially in developing countries or rural areas— electrical board panels face different challenges, not allowing the installation of intrusive smart meters. Furthermore, inexpensive solutions may be a crucial factor in decision for smart meters installation.

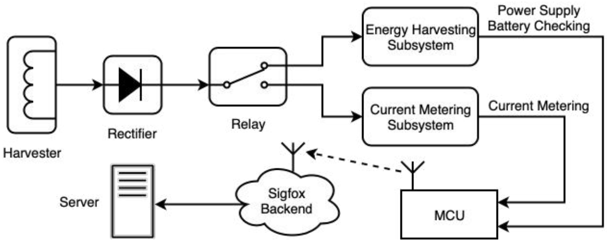


Figure 1: General block diagram for the system

This paper presents a novel device that is able to measure current at 230V single-phase mains lines, with no intrusion in the current electrical installations, which may be crucial to be deployed in emerging countries or rural areas. Neither wired data connection nor power lines are required to make the system work.

## II. SYSTEM DESIGN

The device proposed in this work is meant to measure current at mains electrical lines periodically. The general block for the system can be plotted out as in Figure 1. Measure range has been set to 0A-25A since it seems to be a fair-minded span for most metering scenarios. Moreover, it allows encoding each current measurement with just one byte, providing 0.1A precision —since one byte can represent integers from 0 to 255. This means an overall relative precision of 0.4%. Energy is harvested with a clamp inductor, i.e. a current transformer, which generates an AC current. This current passes through a rectifier where is converted to DC before attacking one of the two following circuits: a) the energy harvesting one, or b) the current metering one. These two circuits are switched depending on the working state.

If the system is in sleep/stand-by mode, the energy harvesting block is working, supplying power to the microcontroller (MCU) and storing the remaining energy in a battery. Otherwise, the MCU is in charge of reckoning the current value in the mains wire by using the current metering block. Values are stored in the MCU's RAM until a sending event is triggered, when data are parsed and transmitted to a backend that performs a request to a server, where data are stored and prepared for visualisation and analysis.

Sigfox has been the selected communication technology. Although it presents some drawbacks in terms of packet size (12-byte messages), bandwidth (100bps) or messages per day (140), it is the most suitable one concerning the smart metering application in different scenarios: rural environments, crowded buildings, developing countries, etc. Sigfox works out-of-the-box with its own network, providing end-to-end encryption and ultra-narrow band modulation [5].

978-1-7281-6158-7/20/$31.00 ©2020 IEEE

Taking this into account, in order to reduce power consumption, it has been decided to acquire current measures every 15 minutes, store them locally and send them every three hours. This sending period answers to the trade-off between Sigfox 12-byte message size and the 1-byte current measurement encoding.

### *A. Energy Harvesting*

Energy harvesting has been a trending topic during the last years, becoming a feasible solution when very-low-power electronic devices were achievable —from the 2010s. Providing energy harvesting capability to wireless devices enables them to continuously acquire energy, therefore eliminating the concern of their lifetime being dependant on the energy storage system capacity [6]. Every energy harvesting device is compounded, at least, of three main blocks: the harvester itself, the signal conditioning block and the storage element.

The device proposed in this work is power-supplied by means of magnetic induction energy harvesting (MIEH). When magnets teeter through a coil, they create a variable magnetic field. This magnetic field happens to generate an electromotive force (EMF) into the coil. This phenomenon is described in Faraday's Law of Induction [7, 8], which explains the EMF created when a time-variant magnetic field is present. As Faraday's Law is reciprocal, every conductor carrying certain amount of current will create a surrounding magnetic field. Be this magnetic field time-variant, it may be able to be exploited. Thus, some amount of electrical energy might be drawn.

This device takes advantage of the magnetic field happened around a conductor carrying electricity —which will be time-variant as mains electricity is AC— to produce an EMF in a coil. This is the principle used in most current probes for current reckoning, notwithstanding with the fact that in this work it is also proven to be the power supply for the system. This approach to MIEH has been little used and there is not much literature about it [9, 10]. The device proposed in this work and [10] use a similar configuration for the main blocks of the system, but different wireless and switching technologies.

### *B. Harvester*

The harvester itself is a clamp inductor —a current transformer— with a current ratio of 1500:1. This means that the current outgoing the inductor is 1500 times lower —the secondary coil— than that being carried in the mains wire — the primary coil. The inductor has been characterised, including measurements for short-circuit current in the secondary coil, open-circuit voltage at the secondary coil and maximum dissipated power. The maximum current that could be used to charge the battery when a current of 25.6A is in the mains wire happens to be 1.13mA, although the short-circuit current can get as high as 17.07mA.

### *C. Microcontroller*

The MCU must comply with different requisites that can be summarised in: i) very low stand-by power consumption, ii) Sigfox-compliant, iii) fast and wide firmware development.

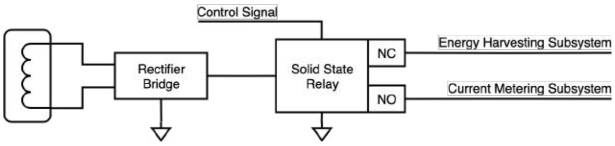


Figure 2: Conceptual schematic of the switching and conditioning circuit

Having this in mind, Pycom LoPy4 is the chosen hardware platform[1]. This ESP32-based board runs MicroPython and has a Sigfox transceiver built-in. It provides a deep-sleep consumption of just 25μA. Moreover, the integrated 12-bit ADC provides 4096 measure points, meaning a theoretical overall precision of 6.1mA within the 0A-25A measurable range —enough for this purpose and above the encoding precision of 100mA set by the encoding format.

### *D. Switching and Conditioning Subsystem*

Since the same clamp is acting as harvester and current probe, a proper conditioning stage is needed to get the signal ready for either energy harvesting or current measurement (see Figure 2). Considering that this power meter is intended for measuring only active power, and that electronic circuitry works on DC, the incoming AC signal is rectified from the very beginning so as to simplify electronic design and procedures. Working with DC allows using a single pole switch —since the other pole is ground—, whilst AC would require dual pole switches. Thus, a full-wave rectifier is implemented: CBRHDSH1-40L[2]. Although every component added to the system will come with certain amount of power losses along, those resulting from an efficient, full-wave rectifier can be considered negligible.

Once the signal is rectified, it must go either to the energy harvesting circuit or the current measurement one. For this switching to be performed, a solid-state relay is chosen: LBA710[3]. Reduced energy consumption in operation and stand-by is crucial, and a relay guarantees a non-consuming default state (normally open, NO). Furthermore, being a solid-state relay means less energy consumption and faster switching than a magnetoelectric one.

### *E. Energy Harvesting Subsystem*

This block is intended to be as efficient and simple as possible in order to maximise energy harvesting capabilities (see Figure 3). The main element of this block is a power management IC intended for low-power energy harvesting applications: BQ25504[4]. The BQ25504 is a boost converter that manages the charge of a battery as main storage element and a capacitor as first-stage, instantaneous storage element.

This IC has a restricted input voltage of up to 3V, but the incoming signal may be higher. Therefore, it is necessary to limit input voltage. Due to efficiency reasons, the buck converter TPS62122[5] is selected, providing an output of 2.8V.

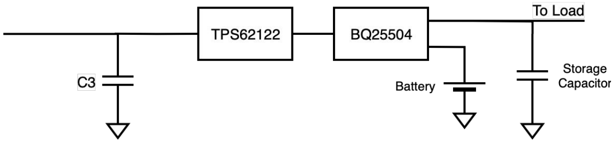


Figure 3: Conceptual schematic of the energy harvesting circuit

---

[1] https://docs.pycom.io/datasheets/development/lopy4
[2] https://www.mouser.com/datasheet/2/68/cbrhdsh1-40l-32670.pdf
[3] http://www.ixysic.com/home/pdfs.nsf/www/LBA710.pdf
[4] http://www.ti.com/lit/ds/symlink/bq25504.pdf
[5] http://www.ti.com/lit/ds/symlink/tps62122.pdf
[1-5] Accessed April 2020

The BQ25504 has been set for an output working range of 3.5V-4.2V, being these values those corresponding to battery (3.4V-4.2V) and MCU (3.5V-5.5V) specifications. These thresholds are set by means of configuration resistors following datasheet instructions.

### *F. Current Metering Subsystem*

This block is in charge of reckoning the current being carried by the mains wire. A measurement resistor is needed for converting the proportional current signal of the probe into a voltage signal for the ADC to be read (see Figure 4). Two series resistors are implemented —i.e. a voltage divider—, sampling the voltage in the second resistor of the branch.

This signal is full-wave rectified, yet a capacitor is still to be set in this block. The ripple is desired to be as negligible as possible, but the littler the ripple is, the longer the stabilisation time becomes. A long stabilisation time is not wished as more time would be spent in measuring instead of harvesting; hence more energy would be wasted. Be that as it may, a perfectly constant signal is not needed as the signal will be sampled 1100 times, then averaged, so the ripple will be overcome.

It is worth noting that, for voltages under 50mV (0.98A), the ADC did not work properly and measurements were very inaccurate, often returning negligible values. To overcome this issue, an offset was set to the signal read by the ADC: a pretended reference was introduced and soared up from GND. This was made with a voltage divider, using the 3.3V coming out from LoPy4's GPIO pin when the relay is toggled. With a capacitor value of 2.2µF, a convergence time of 200ms is obtained. The ripple gets to be 15.5mV in the worst-case scenario. The mean value for the signal plus the ripple (940.7mV) remains lower than the ADC limit (1.1V).

A calibration process has been performed by reading the values returned by the ADC when several known currents were sampled. For this purpose, 48 different current values were sampled (average value of 10000 samples for each current value). This leads to obtain the quadratic approximation (1) for the current by means of ADC readings with an $R^2 = 1$, which means a greater order would not necessarily enhance formula's performance. This formula is then programmed in the MCU's firmware to reckon the current in the mains wire.

$$f(x) = 1.083 \cdot 10^{-4}\, x^2 + 0.147\, x - 9.314 \qquad (1)$$

### *G. Software*

The MCU's firmware is written in MicroPython, intended to be as straightforward and minimal as possible to make processes fast and stable. The behaviour is event-driven, based on a 15-minute timer to take measurements. Its working principle may be summed up as follows:

1. When the device wakes up from sleep mode, it checks the charge of the battery ($VBAT_{OK}$ signal). If it is above the required threshold, it continues running. Otherwise, error is handled, then it returns to sleep.
2. If battery health is good, the metering process begins. The relay is toggled, then stabilisation time is waited. Next, 1100 samples of the signal are collected, averaged and converted to a current value. If the current reading is out of boundaries, an error message is encoded.
3. The new measurement is buffered and, if buffer gets full, the Sigfox sending process is performed to begin anew.

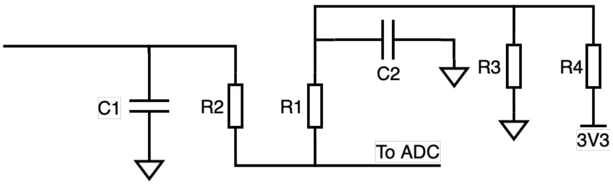


Figure 4: Conceptual schematic for the current metering circuit

The Sigfox backend receives every message and forwards them to a server. In this server, measurements are stored in a MongoDB database, where time stamping is set according to the 15-minute period and the temporal reference Sigfox sends along with the message.

## III. VALIDATION

In order to prove the proper functioning of the device, three main validation steps have been conducted: i) power budget validation, ii) current metering certainty verification, and iii) real-world application final testing.

### *A. Power Budget*

An Agilent 34410A multimeter has been used to measure and record device's working current consumption. There are four main behaviours for the MCU depending on the buffer state and battery level. These are —apart from sleep mode (15 minutes, 23.56µA):

1. Standard measuring process (2.26s, 59.61mA).
2. $VBAT_{OK}$ is low, so an error code is saved (1.85s, 60.55mA).
3. $VBAT_{OK}$ is low, and this error code saving makes the buffer be full, so it must be erased and the counter reset (1.81s, 60.54mA).
4. A measurement which makes the buffer be full is taken, so the Sigfox sending process is launched (12.69s, 70.57mA).

Afterwards, the actual energy injected into the battery has been characterised, checking the battery life of the device. The energy stored into the battery depends, as previously stated, on the mains wire current. Measured charging currents —for a half-charged battery— are presented in Figure 5.

As it can be observed, charging current follows a linear pattern until it reaches the maximum current that the boost converter is able to inject into the battery. In fact, the charging current behaves as a PWM signal with a top current of about 1.13mA, which increases its duty cycle as the mains wire current increases. This is the reason why once the charging

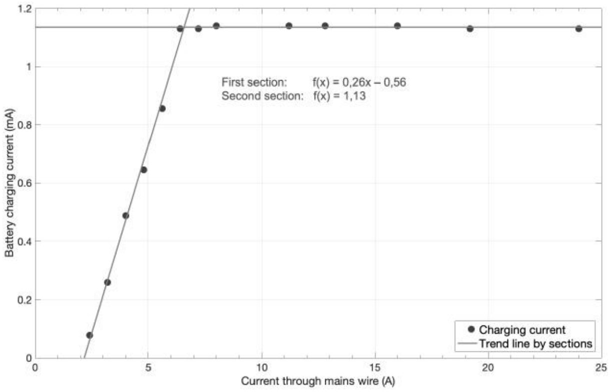


Figure 5: Charging currents for different mains line currents

current provides a 100% duty cycle, it cannot become higher as the top limit is imposed by the charging circuit itself.

With both consumption and generation characterisations, battery life can now be determined. For this calculation, the most energy demandant scenario is used, which is also the most feasible one: that where 12 complete measures are done, sending the buffer in the last one via Sigfox.

The aim of this device is being able to work for a lifetime, so the energy harvested into the battery must compensate the energy consumed in the process. Taking into account the measured consumptions for every iteration, the current needed to be injected into the battery happens to be (2):

$$\Gamma = \frac{2610.28 mAs}{12 \cdot 900s - (11 \cdot 2.26s + 12.69s)} = \\ = 242.5 \mu A \leftarrow \sim 3.08A \qquad (2)$$

If an average current of 3A (690W, e.g. a workstation) is being carried by the mains line wire, an infinite battery lifetime can be guaranteed for 15-minute measuring period. This average current would increase if shorter periods were wished. For instance, for a metering period of 5 minutes, the average current needed would be 4.8A. Considering the 2200mAh battery used in the prototype, if no energy at all were harvested, the device would provide a battery life of more than one year.

### B. *Current Metering*

For metering veracity, 15 different current values have been probed, composed of 10 consecutive measurements each. Standard deviation happens to be 0.032A and the relative error is 3.28% in the whole range of measurement. This relative error is suitable for most cases. Still, it is important to note that it is greatly increased due to the first current values in the lowest range. If measurements below 0.5A were dismissed, the relative error would result in just about 0.8%. Regarding the standard deviation, it is in fact below the encoding format precision limit (0.1A), so it shall not be noticed.

### C. *Real World Test*

The prototype is put under real world conditions working for a week, monitoring a kitchen electrical line (refrigerator, microwave ovens, water heater, coffee machines and lighting). During this period, there were no error messages and the battery level had increased from 3.78V to 3.81V. The average current measured during the test was nearly 4A (920W).

The device is still under test in order to obtain more measures and confirm its good functioning for longer periods of time. A picture of the prototype can be seen in Figure 6.

## IV. CONCLUSION

Bringing an easy way of metering power consumption is a crucial pillar to counteract the increase of energy consumption and $CO_2$ emissions. Having a record on consumptions, people's consumption patterns and electricity flows, among other aspects, is essential to detect energy sinks, failures and irresponsible habits.

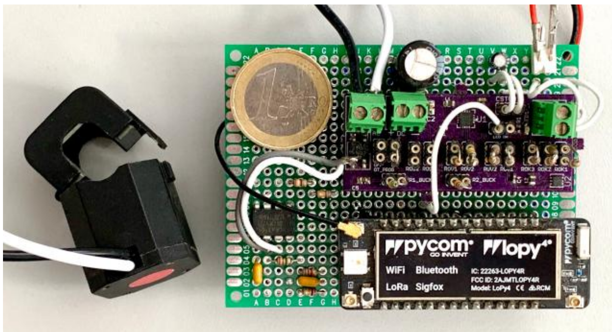


Figure 6: Functioning prototype

In this paper, a self-powered, Sigfox-compliant smart meter is presented. A fully functional design is demonstrated to work. Its electronic design has been developed in order to pursue an efficient, simple design. This design allows harvesting enough energy from the mains line so that the device can work autonomously. Its installation simplicity, scenario versatility and low-cost entails an optimal solution for developing countries and rural areas. The cost for the prototype can be stated to be around 80€, which shall be lowered to around 20€-30€ for a chain-produced device with OEM elements.

Experimental results show an overall precision of about 3.3% for the whole range of measurement, which is lowered to just around 0.8% if the range 0.5A-25A is considered. However, due to the selected encoding restrictions, final recorded values have 0.1A steps which are fairly above the standard deviation values.

Final tests prove the device is able to self-power itself perpetually when a current of 3A is carried by the mains electricity line, whilst metering current every 15 minutes.

## REFERENCES

[1] "World Power consumption | Electricity consumption | Enerdata," [Online]. Available: https://yearbook.enerdata.net/electricity/electricity-domestic-consumption-data.html. [Accessed January 2020].

[2] European Comission, Joint Research Centre and Netherlands Environmental Assessment Agency, "EDGARv4.3," 2015.

[3] A. Reinhardt, D. Burkhardt, P. S. Mogre, M. Zaheer, R. Steinmetz, "Smartmeter.kom: A low-cost wireless sensor for distributed power metering," in *IEEE 36th Conference on Local Computer Networks (LCN), 2011*, 2011, pp. 1032–1039.

[4] CeDInt UPM, "Industrial IoT IP open solution for smart production," Madrid, 2017.

[5] "Technology | Sigfox," [Online]. Available: https://www.sigfox.com/ /en/what-sigfox/technology. [Accessed January 2020].

[6] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover and K. Huang, "Energy Harvesting Wireless Communications: A Review of Recent Advances," IEEE Journal on Selected Areas in Communications, vol. 33, no. 3, pp. 360-381, 2015.

[7] E. Jordan and K. G. Balmain, Electromagnetic Waves and Radiating Systems, Prentice-Hall, 1968.

[8] W. Hayt, Engineering Electromagnetics, McGraw-Hill, 1989.

[9] M. N. Vo and M. A. Noras, "Energy Harvesting from Electromagnetic Field Surrounding a Current Carrying Conductor," in ESA Annual Meeting on Electrostatics, Cocoa Beach, FL, USA, 2013.

[10] D. Porcarelli, D. Brunelli, and L. Benini, "Clamp-and-measure forever: A mosfet-based circuit for energy harvesting and measurement targeted for power meters," in *Advances in Sensors and Interfaces (IWASI), 2013 5th IEEE International Workshop on*, 2013, pp. 200–205.