Internet of Things Devices Management for Smart Cities

Nilson Rodrigues Sousa[1,2], George P. Pinto[1,2][a] and Cassio V. S. Prazeres[1][b]
[1]Institute of Computing, Computer Science Department, Federal University of Bahia (UFBA), Salvador, Bahia, Brazil
[2]Federal Institute of Bahia, Brazil
{nilson.sousa, george.pacheco, prazeres}@ufba.br

Keywords: Internet of Things, Smart City, Management, Device.

Abstract: In this work, we address the critical challenge of managing IoT devices within Smart City infrastructures. We propose a comprehensive solution tailored to the specific requirements of IoT device management, different from traditional network device management. Our approach integrates hundreds of devices across urban areas, leveraging telecommunications and information technologies (ICT) to improve urban services and citizens' quality of life. We reviewed existing architectures and platforms and developed a prototype to demonstrate the practical application of our solution. Our prototype ensures consistent service availability and efficient resource management. The insights gained from our work provide valuable guidance for future developments and implementations of IoT device management strategies in Smart Cities.

1 INTRODUCTION

Smart Cities are urban environments that utilize telecommunications and information technologies (ICTs) to provide better public services and more efficient use of existing resources. According to Janani et al. (RP et al., 2021), a Smart City combines advances in technologies such as the Internet of Things (IoT), big data, social networks, and cloud computing with the demand for cyber-physical applications in the public interest, such as health, public safety, and mobility.

Many cities, such as Seoul (Joo, 2023), Barcelona (Kadiri et al., 2023), Tokyo (Wolniak and Grebski, 2023) , Singapore (Ang-Tan and Ang, 2022), and Dubai (Sahib, 2020), adopt ICT-based solutions to address various urban challenges. Each solution employs dozens or hundreds of IoT devices and various software and communication protocols (Jabbar et al., 2024). In this context, the increasing number of devices employed in various IoT solutions is a trend that highlights the need for the management of such devices.

According to Ashraf (Ashraf, 2021), numerous IoT devices are revolutionizing urban environments, transforming cities into smart cities where every event becomes part of an interconnected network. A crucial element in constructing these smart cities is the integration of wireless sensors within the IoT devices. As highlighted by Gustin and Jasperneite (Gustin and Jasperneite, 2022), management has become a significant research area within IoT, mainly due to the connection of a large volume of heterogeneous devices with limited resources.

In addition to the IoT device amount problem, such devices can present limitations, such as storage capacity, processing power, energy source, and communication method (Sehgal et al., 2012; de Andrade Junior et al., 2014; Mershad and Cheikhrouhou, 2023). Also, the geographical distribution of devices throughout the city represents a significant challenge and underscores the importance of managing and maintaining the IoT infrastructure in Smart Cities (Biswas and Giaffreda, 2014).

However, traditional management solutions struggle to handle these environments due to resource, communication, device diversity, and scalability limitations, as highlighted by Zhang et al. (Zhang et al., 2022). On the subject, Zahoor and Mir (Zahoor and Mir, 2021) emphasize developing lightweight and straightforward approaches to transparently manage the entire Smart City infrastructure, maintaining its operation and performance. As a result, several solutions have been proposed that aim to manage IoT devices efficiently.

Nevertheless, management approaches often operate independently within Smart City solutions. For

[a] https://orcid.org/0000-0002-6082-9211
[b] https://orcid.org/0000-0003-0197-0909

Sousa, N. R., Pinto, G. P. and Prazeres, C. V. S.
Internet of Things Devices Management for Smart Cities.
DOI: 10.5220/0013193700003944

In *Proceedings of the 10th International Conference on Internet of Things, Big Data and Security (IoTBDS 2025)*, pages 15-26
ISBN: 978-989-758-750-4; ISSN: 2184-4976

example, an intelligent parking management approach may not interact, at the management level, with an intelligent traffic solution, even though both are deployed in the same city or even in the same neighborhood (Pacheco Pinto and Prazeres, 2019). Given this context, this work presents a proposal for managing devices that comprise the IoT solution infrastructure in the context of Smart Cities.

Our proposal consists of an approach for device management from two perspectives: first, local, where management is carried out within the context of a specific Smart City solution; second, grouped, where two or more solutions can be managed together. Consequently, if any problem occurs in the infrastructure of these solutions, it can be identified, addressed, and corrected as necessary. In addition to the developed proposal, we implemented an application prototype to demonstrate how management can generally occur in this context.

The remainder of this article is organized as follows: Section 2 presents the related works that have proposals for device management, where a brief comparison between them is also made. Section 3 describes the monitoring and management proposal developed in this work. Next, Section 4 provides a general description of the application environment of the proposal, presents the organization of the simulated environment used in this work, and describes the developed implementation. Finally, future works and final considerations are presented in Sections 5 and 6, respectively.

2 RELATED WORKS

Various platforms offer management solutions for IoT devices to optimize their handling in different contexts. In the context of Smart Cities, the following sections highlight some platforms with management solutions referenced by Gustin and Jasperneite (Gustin and Jasperneite, 2022). The platforms KAA, ThingsBoard, Eclipse Kapua, OpenBalena, and Fiware appear as related works in this study due to their characteristics, which align with the proposed solution. These platforms share essential functionalities for efficient IoT device management, such as scalability, interoperability, and real-time monitoring, which are crucial for Smart City applications. Additionally, the study compares the different solutions in Section 2.6, including the proposed approach.

2.1 KAA

Kaa is an open-source platform designed to assist in developing IoT applications and connected services. According to Agarwal and Alam (Agarwal and Alam, 2020), the platform offers features for managing IoT devices, processing data, and integrating with other systems. In device management, Kaa covers connection, configuration, and monitoring routines (KaaIoT, 2024). Kaa also provides interoperability with third-party systems and security features that include authentication, authorization, and encryption. Its modular and extensible architecture allows the platform to adapt to the specific needs of projects. He also supports several communication protocols like MQTT, CoAP, and HTTP.

2.2 ThingsBoard

According to Okhovat and Bauer (Okhovat and Bauer, 2021), ThingsBoard is a flexible and scalable open-source IoT platform. It offers a wide range of features for IoT networks, such as provisioning and managing both real and virtual sensors and devices. The platform also supports data collection, analysis, visualization, and device control. Furthermore, ThingsBoard can work with various IoT communication protocols, including MQTT, CoAP, and HTTP (ThingsBoard, Inc., 2024). Key functionalities related to device management in ThingsBoard are described in Table 1.

2.3 Fiware

Fiware is an open-source platform designed to create intelligent IoT solutions in cities, industries, and agriculture (Ahle and Hierro, 2022). It provides an infrastructure based on modular components that enable data collection, processing, and management. Third-party components can be added to the context broker to enhance compatibility with various solution needs. Fiware also offers components for IoT device management, each with specific functionalities aimed at monitoring and controlling devices. Table 2 describes some of the Fiware components.

2.4 Eclipse Kapua

Developed by the Eclipse Foundation, the Eclipse Kapua platform, according to Nardis et al. (De Nardis et al., 2022), is a modular platform focused on integrating IoT devices into an infrastructure. It is scalable, and among its features is the management of IoT devices, which provides an infrastructure adapted

Table 1: ThingsBoard key functionalities.

Functionality	Description
Support for Multiple Protocols	ThingsBoard supports protocols like MQTT, CoAP, and HTTP, facilitating the integration of various devices
Remote Management	Enables the configuration, monitoring, and remote control of IoT devices
Device Provisioning	Eases the addition and initial configuration of new devices
Real-Time Monitoring	Offers customizable dashboards to visualize the status and performance of devices
Automation and Rules	Features a rule engine to automate actions based on data received from devices

Table 2: Fiware components.

Component	Description
Orion Context Broker	The primary component for managing information, it collects, updates, and distributes data from IoT devices
IoT Agent	Interface that allows devices to connect to the Orion Context Broker using different communication protocols (MQTT, HTTP)
Cygnus	Connector used to persist historical data in various storage systems, such as Hadoop, MySQL, and others
Quantum Leap	Connector for storing data to aid in historical data analysis
Wilma PEP Proxy	Security provider that manages authentication and authorization for accessing services within the FIWARE ecosystem

for various operations. Kapua provides such management through an open application protocol running over MQTT, which allows one to control, configure, start, and stop devices remotely from applications. These features help in efficient and real-time management of the performance of connected devices.

2.5 OpenBalena

OpenBalena is an open-source platform that facilitates the management of large-scale IoT devices, according to Roda-Sanchez et al. (Roda-Sanchez et al., 2023), it offers features such as device provisioning, remote updates, continuous monitoring, application management via Docker containers, secure communication, control access, and scalability. These features allow for the efficient and secure administration of many devices, ensuring they are always up-to-date and operating correctly. However, configuring and managing the platform can be complex depending on the solution's implementation level.

2.6 Comparison Between Works

In Table 3, it is possible to identify the type of proposal for each work with frameworks, architectures, and platforms presented. We have defined our proposal as a management method.

Table 3 also describes the coverage of management proposals. Coverage refers to managing different solutions as a single integrated solution. In this case, only our proposal covers grouped management solutions. The other solutions perform management individually. The Managed Components column shows the types of entities the platforms manage. It is possible to notice that they all deal with any device type. The routines column presents the general functionalities of each platform. Except for Fiware, the other platforms have the same functionalities. Finally, in the Protocol column, the different supported protocols are presented. In this column, our proposal is to have a manager adapt to the protocols provided by the solutions infrastructure.

3 SCM: IoT DEVICE MANAGEMENT

Smart City Management (SCM) is our proposal for organizing and managing devices in the various IoT solutions implemented for Smart Cities. The SCM aims to facilitate the development of managerial solutions in the complex context of smart cities.

The SCM has been structured into four fields, each addressing an aspect necessary for device management. Figure 1 presents each field, which must be well defined according to the management needs

Table 3: Related works comparison.

Work	Year	Coverage	Managed Components	Routines	Management Routines
KAA (Agarwal and Alam, 2020)	2020	Individual Solution	Devices and Gateway	Connection, configuration, and monitoring routines	MQTT, CoAP, HTTP, WebSocket
ThingsBoard (Okhovat and Bauer, 2021)	2021	Individual Solution	Devices, Software and Gateway	Maintenance of IoT networks, management of real and virtual sensors and devices	MQTT, CoAP, HTTP, SNMP, LoRaWAN
Fiware (Ahle and Hierro, 2022)	2021	Individual Solution	Devices and Gateways	Monitoring and device control	MQTT, CoAP, HTTP, AMQP, LwM2M, LoRaWAN
Eclipse Kapua (De Nardis et al., 2022)	2022	Individual Solution	Devices, Softwares and Gateway	Performs control, configuration of hardware and software, startup and shutdown of devices	MQTT, CoAP, AMQP, HTTP, WebSocket, LwM2M
OpenBalena (Roda-Sanchez et al., 2023)	2023	Individual Solution	Devices, Software and Gateway	Device maintenance, remote updates, continuous monitoring, software management	MQTT, CoAP, HTTP, AMQP, WebSocket
Our Proposal	**2024**	**Grouped Solution**	**Devices and Gateways**	**Registration, monitoring, remote control, device maintenance, and management data administration**	**The protocol implemented by the solutions**

of the proposed solution. The definitions made will guide the development of the management platform.

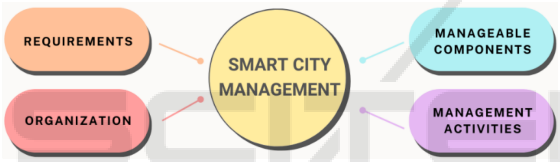

Figure 1: Fields defined in SCM.

3.1 Requirements

The SCM must cover several requirements to effectively manage IoT devices in a Smart City. These requirements are Heterogeneity, Scalability, Protocols, Management, and Control.

Heterogeneity concerns the need for management solutions to enable independent management of all infrastructure components. In this sense, each solution must deal with different types of devices, sensors, actuators, and forms of communication, among others.

The scalability requirement dictates that a management solution must be capable of managing any number of devices within the infrastructure's forecast without compromising overall performance. The protocol requirement states that management solutions should support traditional IoT device protocols. This requirement enables management to utilize communication optimization techniques, such as opportunistic methods.

The device management requirement states that one must clearly define the components and characteristics that require management. In the case of SCM, this entails specifying physical devices, their configurations, states, and modes of operation. Finally, the device control requirement specifies that both remote and non-remote access must be available to enable maintenance to sustain the existing infrastructure's full functionality.

3.2 Manageable Components

The manageable components refer to the assets that need to be managed. In the context proposed by SCM, management covers all the physical components of a solution. It is worth noting that only hardware characteristics, such as memory, processing power, energy capacity, and geographic location of the device, are being considered for management at the moment.

Figure 2 presents an example of an IoT solution for public lighting in a Smart City. In this example, we can identify various manageable hardware components covered by SCM. This study's identified components of interest include devices, sensors, actuators, and gateways.

A Device encompasses any hardware component comprising sensors and actuators. A sensor is a more straightforward component that can capture environmental data. An actuator represents components that can effect changes in the environment where they are installed, such as opening and closing a door. Finally, a Gateway is a component that facilitates communication between simple devices with limited communication capabilities, such as sensors and actuators. The gateway may also perform certain types of simple data processing.

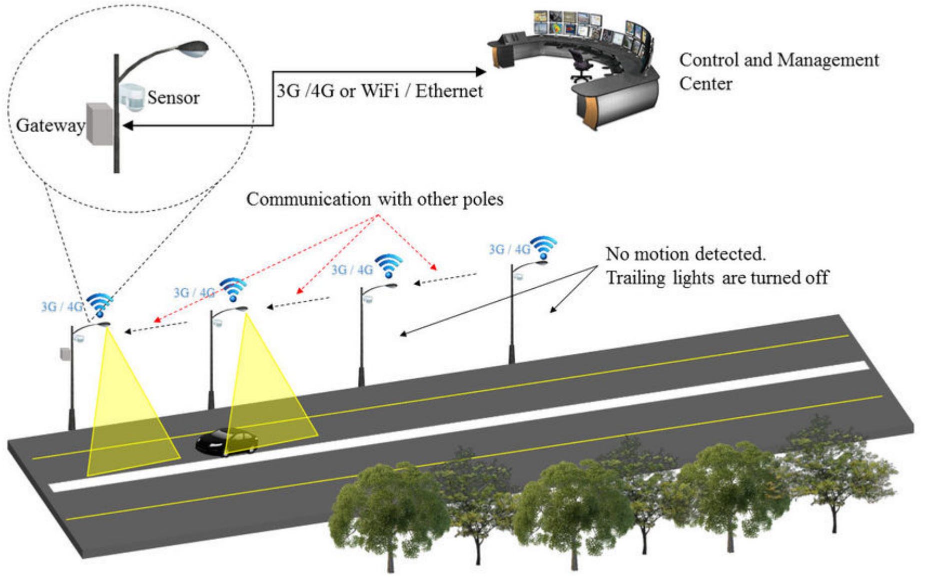

Figure 2: Example of smart street light scenario. Obtained from Taiwan Sourcing Service Provider (Taiwan Sourcing Service Provider, 2015).

3.3 Organization

A Smart City comprises solutions that cover different areas and meet the different needs of a city, which means that each solution has its characteristics. Therefore, each SCM-based management solution must meet a specific set of requirements presented in Section 3.1 and the specific aspects of the solution that are relevant to management. In this context, each IoT solution will have its implementation for managing its infrastructure based on SCM. This management solution, defined as a Smart Manager (SM), can make the management of different solutions more flexible. In this context, flexibility refers to the possibility of centralizing or not the management of different solutions. To exemplify this situation, Figure 3 presents three IoT solutions for a Smart City with their characteristics.

In Figure 3, the three examples of solutions for Smart City present different characteristics regarding communication protocols, types of existing devices, communication methods, connectivity with the cloud, and version of Smart Manager. These differences mean that only the Smart Traffic and Smart Parking solutions, for example, offer flexibility, with a central controller capable of managing both solutions as if they were one. On the other hand, the Water Quality solution lacks a connection to the cloud, and although it has its version of the Smart Manager, it can only be managed separately from the others.

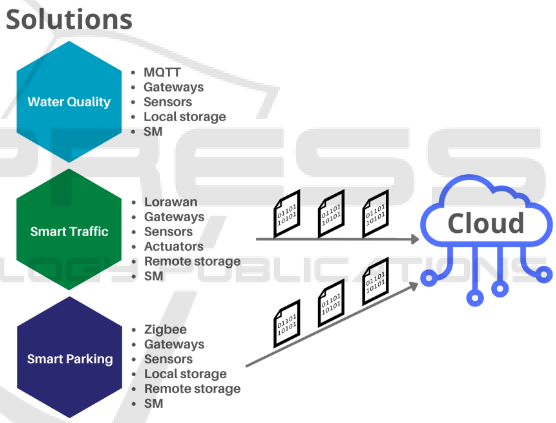

Figure 3: Smart City solutions – simplification.

3.4 Management Activities

In the context of this work, management activities encompass routines and practices aimed at monitoring, controlling, and maintaining the efficiency and availability of devices connected to the infrastructure of any Smart Solution. We have identified device registration, data storage management, device monitoring, and device control as essential activities to achieve this.

Device registration concerns the ease of registering devices in management databases. Monitoring is the activity responsible for maintaining supervision over a device. The data storage management activity corresponds to executing routines for efficient data storage management, including registration and monitoring data. Device control refers to routines for controlling device parameters registered and managed by a Smart Manager. Finally, device maintenance is the activity that enables correcting problems and carrying out updates.

4 ENVIRONMENT SIMULATION AND IMPLEMENTATION

During the development of our management proposal, we created an initial prototype of a Smart Manager for managing IoT devices in IoT Smart City solutions (referred to as Smart Solution from this point forward). Figure 4 provides an overview of how applications are organized in the context of a Smart City.

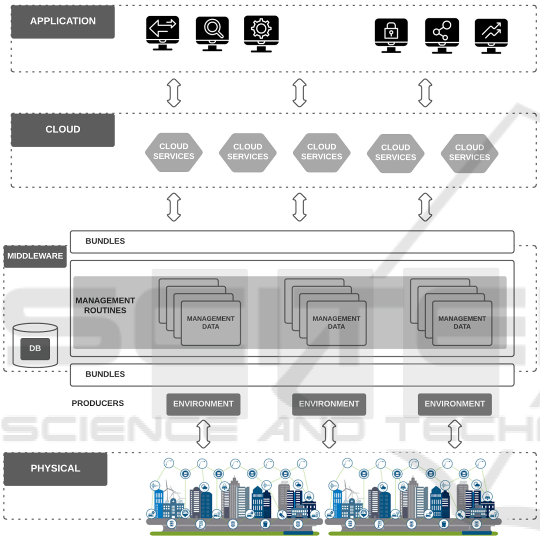

Figure 4: Overview of management in smart cities.

As depicted in Figure 4, we structured the prototype into four layers: physical, middleware, cloud, and application layers. The quantity of functionalities in each layer may vary depending on the specific organization of each solution. However, the presented organization aligns with the standards in which applications, in this context, are implemented.

The physical layer encompasses the devices present in all Smart City solutions, which may be geographically dispersed depending on the implementation. These devices exhibit a high degree of heterogeneity, possessing distinct characteristics.

The middleware layer facilitates communication and interaction among diverse devices within a Smart Solution and between different Smart Solutions. The bundles are software units that include management routines. These routines may be distributed across various components (devices, gateways, servers, cloud) within a Smart Solution.

The database in the middleware layer encompasses all possible valuable forms of data persistence, including temporary storage on devices, permanent storage on local or distributed servers, and cloud storage.

The cloud layer represents the utilization of internet services, which can execute all processing for the Smart Solution and/or the Smart Manager. An alternative is using the cloud only to perform more complex tasks requiring greater processing power.

Lastly, the application layer embodies the essential software within the domain of Smart Solutions. This software encompasses the specific solutions, the Smart Manager, and any third-party software leveraging resources via APIs.

4.1 Environment Simulation

We developed an initial version of a Smart Manager as a proof of concept. It was used in a simulated environment, as illustrated in Figure 4. Furthermore, Figure 5 illustrates the organization of the Smart City environment along with its Smart Solutions.

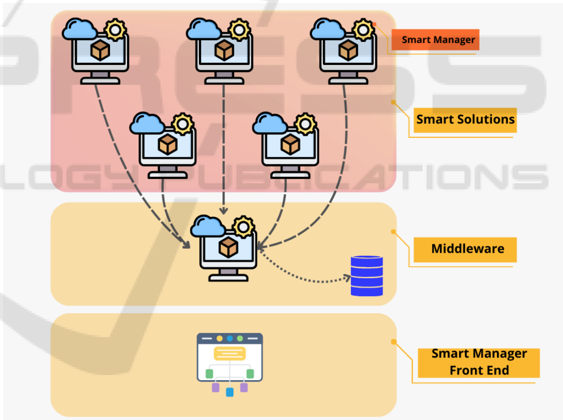

Figure 5: Organization of the Simulation Environment.

As we can observe in Figure 5, there are three different layers. Each layer consists of virtual machines with specific functions in the simulated environment. The Smart Solutions layer is made up of virtual machines (VMs) that simulate different solutions within a Smart City. Each solution comprises virtual sensors, actuators, and gateways that simulate management data.

Figure 6 presents an example of simulated management data representing a gateway in the simulated solution. For simplicity, only data related to the MAC address (used as the gateway identifier), IP, manufacturer, hostname, operational status, registration date, associated solution, and location coordinates were

```
{
  "mac": "11:11:3:22:11:30",
  "ip": "2.2.165.95",
  "manufacturer": "WiserUFBA",
  "hostName": "GT88",
  "status": true,
  "date": {
    "year": 2024,
    "month": 9,
    "dayOfMonth": 2,
    "hourOfDay": 15,
    "minute": 5,
    "second": 8
  },
  "solution": "air pollution",
  "coordinates": "57.0030, 86.2199"
},
```


Figure 6: Gateway data simulation.

used to represent the gateway in the prototype.


```
{
  "date": {
    "year": 2024,
    "month": 9,
    "dayOfMonth": 2,
    "hourOfDay": 15,
    "minute": 5,
    "second": 9
  },
  "gatewayMac": "11:11:3:22:11:30",
  "baterryLevel": 55.0,
  "usedMemory": 77.0,
  "usedProcessor": 41.0
},
{
  "date": {
    "year": 2024,
    "month": 9,
    "dayOfMonth": 2,
    "hourOfDay": 15,
    "minute": 5,
    "second": 10
  },
  "gatewayMac": "11:11:3:22:11:30",
  "baterryLevel": 81.0,
  "usedMemory": 46.0,
  "usedProcessor": 16.0
},
```


Figure 7: Gateway status data simulation.

Figure 7 shows the operational status of the simulated gateway depicted in Figure 6. In the figure, the "date" key allows the identification of the recorded status at two different time points. The remaining data represent battery level, total memory usage, and processor usage rate (assuming the gateway has these capabilities).

Just as we simulated the gateway, we also simulated the devices. Figure 8 presents the data used to create a simulated device. Each device has seven attributes: an identifier that distinguishes the device from others connected to the same gateway; a location that may differ from the gateway's location; the device type, which refers to its function; the category, indicating whether it is an actuator or a sensor; the status, representing the device's state at a specific moment; and the registration date of the device on the platform.


```
{
  "id": "1301",
  "location": "122.9689, -44.7436",
  "description": "DTM",
  "typeDevice": "atmospheric",
  "category": "actuador",
  "status": true,
  "date": {
    "year": 2024,
    "month": 9,
    "dayOfMonth": 2,
    "hourOfDay": 15,
    "minute": 5,
    "second": 8
  },
```


Figure 8: Device data simulation.

The simulation of the device's operational status uses data such as those presented in Figure 9. The device attribute, with details omitted for space considerations, includes the same information as in Figure 8; however, among the hidden data, the 'status' remains the only value that may change. The 'date' indicates when the system recorded the operational data of the device, while the 'situation' describes the device's mode of operation (whether it is in use, on standby, in testing, among others).


```
{
  "device": {↔ 8 ↔},
  "date": {
    "year": 2024,
    "month": 9,
    "dayOfMonth": 2,
    "hourOfDay": 16,
    "minute": 13,
    "second": 24
  },
  "situation": "operational"
},
```


Figure 9: Device status data simulation.

It is worth noting that the generated data include only those relevant for executing management routines. Thus, standard data generated by sensors, such as temperature, luminosity, and others, are not produced by this simulation. Additionally, cloud resources were not used in the environment shown in Figure 5.

4.2 Back-End Implementation of the Smart Manager

The implementation of the Smart Manager, as shown in Figure 5, utilizes Apache Karaf, according to (Edstrom et al., 2013), is a modular platform built on

OSGi[1] (Open Service Gateway Initiative). It is designed to run applications in projects that require runtime flexibility and management—a crucial feature for device management, as managed solutions demand maximum availability. Karaf functions as an OSGi container, managing modular software packages (bundles) and enabling distributed applications to run with support for dynamic updates and configurations without requiring a reboot.

Initially, each virtual machine representing a distinct Smart Solution runs a Smart Manager instance (OSGi bundle). In the simulated environment, this instance communicates with all simulated gateways; however, in a real environment, each gateway should have its own Smart Manager instance.

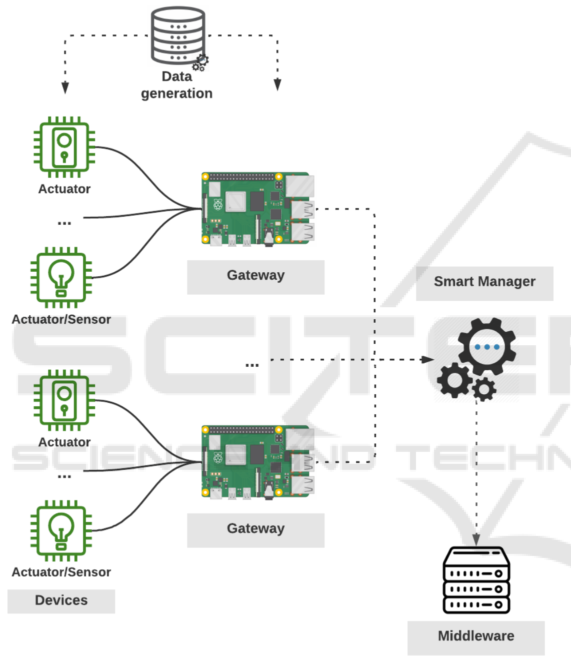

Figure 10: Details on Smart Manager's Operation on Virtual Machines.

Figure 10 illustrates the organization of Smart Manager within each smart solution, specifically within each virtual machine in the simulation. Each virtual machine hosts a defined number of device and gateway instances, with each device connected to a single gateway. The data generator produces both the management data for the gateways and devices, along with their respective instances. The virtual machine sends the management data from the gateway to the local Smart Manager instance, which then directs the data to the middleware layer.

A virtual machine representing the middleware layer, as shown in Figure 5, runs a distinct instance of Smart Manager. This instance receives data from Smart Solutions, regardless of the communication protocol used, processes it, and stores it in the database. The Smart Manager on the middleware layer also provides an API for accessing data across all Smart Solutions and enables sending commands to devices (a feature not yet implemented in the current simulation).

4.3 Front-End Implementation of the Smart Manager

The prototype developed, as presented in the organization of the simulation environment in Figure 5, also includes a web application[2] in the layer, responsible for aggregating and displaying data from all managed Smart Solutions.

The developed front end consists of a ReactJS application that uses the Smart Manager API in the middleware layer to access device data for each Smart Solution. It's important to note that any application designed to manage devices, regardless of platform, only needs to connect to the Smart Manager API to seamlessly access various types of devices within the Smart Solutions.

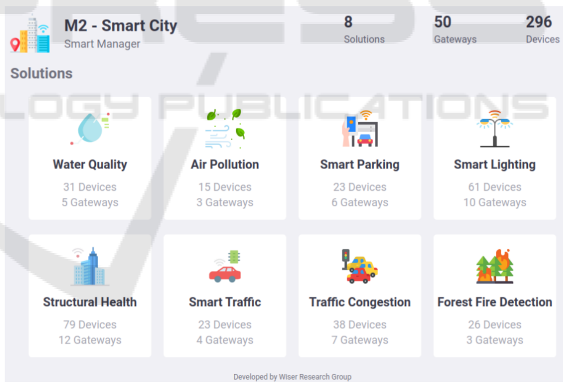

Figure 11: Visualization of all solutions in one city.

Figure 11 shows the Smart Manager front-end home screen. It displays eight different types of Smart Solutions in operation, each with a specific number of devices and gateways. It's important to note the distinction made here between devices and gateways. A device includes any hardware (in this case, sensors and actuators) without direct network and/or internet connectivity. A gateway, on the other hand, is hardware with network and/or internet connectivity that also allows devices to connect through it. The total number of managed devices and gateways is also dis-

[1]https://www.osgi.org/resources/where-to-start/

[2]https://m2dashboard.netlify.app/

played.

Figure 12 shows the specific gateways of a Smart Solution, including their names, IPs, and owners. Accessing these details also provides further gateway information. Since the gateways enable network connections for devices, each gateway has a set of connected devices. In the developed interface, users can select a gateway to view all devices connected to it, including the number of sensors and actuators within the solution.

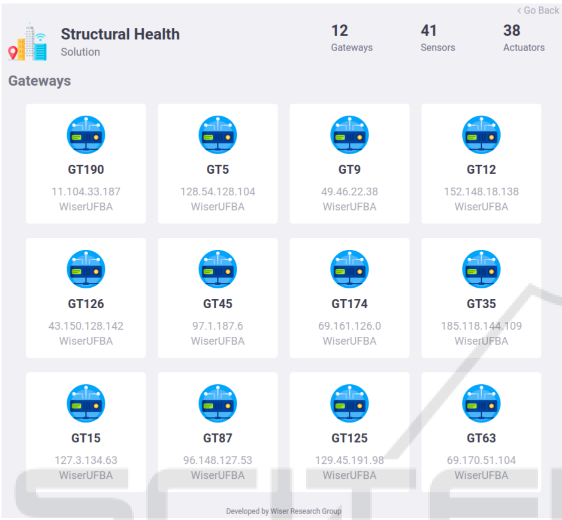

Figure 12: List of gateways for a specific solution.

The screen in Figure 13 displays detailed information about a selected gateway. It first lists the devices connected to the gateway, allowing users to identify the sensors and actuators linked to it and access their details. There is also a map showing the approximate location of the gateway under review. Next, a performance graph displays memory usage and battery level. This graph shows random performance values since, as mentioned, both the gateways and their data were simulated. Finally, Figure 13 includes a table with detailed gateway performance data.

Figure 14 presents the detailed view for a single sensor or actuator. This screen features three fields with device information. The first field is a status history table, where users can see when the device was active, its location, and the gateway it is connected to. The second field is a map showing the device's current location, as it does not necessarily need to be in the same place as the gateway it connects through. The third field, located at the top of the page, identifies the device name, category (sensor or actuator), type, and current status.

However, as previously mentioned, our implementation only covered the general monitoring functions of devices, consolidating their monitoring into a single interface, the presented front end. This implementation represents one of the possible organizational approaches for managing Smart Solutions.

5 FUTURE WORKS

This section outlines key areas for expanding the IoT device management system, focusing on evaluating user acceptance through the Technology Acceptance Model (TAM) and identifying broader research opportunities.

5.1 Planning for Future Experimentation

The evaluation of the IoT device management platform will be based on the Technology Acceptance Model (TAM), as introduced by Davis (Davis, 1989). This evaluation focuses on the constructs of Perceived Usefulness (PU) and Perceived Ease of Use (PEOU) (Davis, 1989). In summary, we aim to assess user acceptance of the platform by measuring these constructs through a structured questionnaire followed by statistical analysis.

As a result, we will measure the users' feelings about the usefulness and ease of use of our IoT management platform. We also aim to understand how these perceptions affect their willingness to adopt and continue using the platform to manage IoT devices in Smart City environments.

We think that participants in the study should include professionals who manage IoT systems, such as city infrastructure managers and technical staff involved in Smart City operations. These individuals will use the IoT management platform in a controlled setting, where they will oversee IoT devices throughout urban areas. To do that, we plan to use a simulation/emulation tool based on the Mininet (Lantz and O'Connor, 2015)[3] and developed in previous works (Sousa and Prazeres, 2018; Coutinho et al., 2018; Batista et al., 2022), where we can deploy our IoT management platform. Afterward, the users will be asked to complete a questionnaire designed to assess their views on how helpful and easy the system is to use.

The questionnaire will feature a Likert (Likert, 1932) scale (1 = Strongly Disagree, 5 = Strongly Agree) to capture their responses and will focus on two key areas: Perceived Usefulness (PU) and Perceived Ease of Use (PEOU). There will also be a sec-

[3]https://mininet.org/

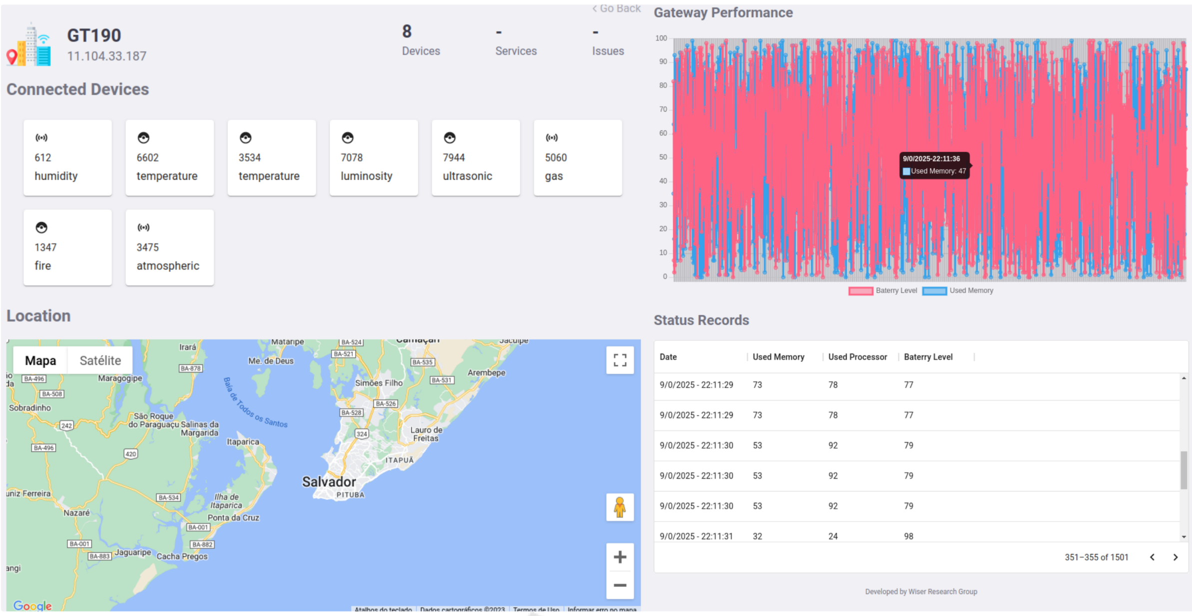

Figure 13: Details of a given gateway.

Table 4: Example Questions for Each Category.

Category	Example Questions
Perceived Usefulness (PU)	- The platform improves my efficiency in managing IoT devices. - Using this platform enhances the quality of urban service management. - The platform helps me to monitor a large number of distributed devices effectively.
Perceived Ease of Use (PEOU)	- The platform is easy to learn and use. - I can quickly and easily perform my tasks on the platform. - The system's interface is intuitive and user-friendly.
Intention to Use	- I plan to continue using the platform to manage IoT devices. - I would recommend this platform to others in the IoT management field.

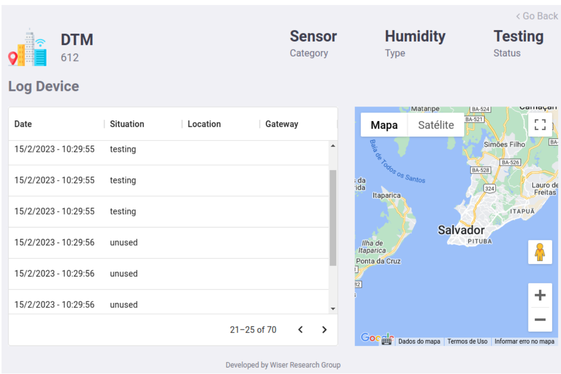

Figure 14: Details of a particular device connected to a gateway.

tion for general feedback and the participants' intention to continue using the platform. The questions also will evaluate how the system helps improve task performance and how simple it is for users to operate. A summary of the survey design is provided in Table 4.

After the survey, the collected responses will be analyzed using statistical tools to determine average PU and PEOU scores. Correlation analysis will explore how ease of use impacts perceived usefulness and how both influence users' willingness to keep using the system. Regression models will further evaluate the effect of these factors on overall user acceptance.

This approach ensures a thorough review of the system's usability and its real-world benefits. The findings will offer valuable insights into both the platform's strengths and areas for improvement, ultimately helping to make it more effective for Smart City applications. These insights will also guide future development and refinements to the IoT management system.

5.2 Other Future Works

This work also identifies some possibilities for future research. First, researchers could expand Smart Manager management routines by conducting more comprehensive investigations into potential management activities within the discussed context. Second, there is a need to formalize an architecture that describes the management of existing solutions in Smart Cities. Achieving this requires research into essential management aspects, including management routines, devices, and solution operations. Lastly, there is a need to develop a platform for simulating and generating IoT devices capable of producing management information. This necessity arises because existing simulators focus solely on generating data like temperature and humidity rather than data derived from the infrastructure.

6 FINAL REMARKS

In this study, we investigated device management across different IoT solutions in the context of Smart Cities. Throughout the work, we presented the management aspects of IoT devices and their implications in the Smart Cities environment.

As a result, we developed a device management method that covers various existing solutions in a Smart City, setting it apart from other management solutions. Additionally, we presented an initial version of a device manager, described throughout the work alongside the method. Also, as a contribution to this study, we provided insights that can guide other works in the development and implementation of effective strategies for device management in the context of Smart Cities.

ACKNOWLEDGMENTS

The authors would like to thank FAPESB, CAPES, and CNPq organizations for supporting the Graduate Program in Computer Science at the Federal University of Bahia. This study was financed in part by the FAPESB INCITE PIE0002/2022 grant.

REFERENCES

Agarwal, P. and Alam, M. (2020). Investigating iot middleware platforms for smart application development. In Ahmed, S., Abbas, S. M., and Zia, H., editors, *Smart Cities—Opportunities and Challenges*, pages 231–244, Singapore. Springer Singapore.

Ahle, U. and Hierro, J. J. (2022). *FIWARE for Data Spaces*, chapter 24, pages 395–417. Springer International Publishing, Cham.

Ang-Tan, R. and Ang, S. (2022). Understanding the smart city race between hong kong and singapore. *Public Money & Management*, 42(4):231–240.

Ashraf, S. (2021). A proactive role of iot devices in building smart cities. *Internet of Things and Cyber-Physical Systems*, 1:8–13.

Batista, E., Figueiredo, G., and Prazeres, C. (2022). Load balancing between fog and cloud in fog of things based platforms through software-defined networking. *Journal of King Saud University - Computer and Information Sciences*, 34(9):7111–7125.

Biswas, A. R. and Giaffreda, R. (2014). Iot and cloud convergence: Opportunities and challenges. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 375–376.

Coutinho, A., Greve, F., Prazeres, C., and Cardoso, J. (2018). Fogbed: A rapid-prototyping emulation environment for fog computing. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340.

de Andrade Junior, N. V., Bastos, D. B., and Prazeres, C. V. S. (2014). Web of things: Automatic publishing and configuration of devices. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*, WebMedia '14, page 67–74, New York, NY, USA. Association for Computing Machinery.

De Nardis, L., Mohammadpour, A., Caso, G., Ali, U., and Di Benedetto, M.-G. (2022). Internet of things platforms for academic research and development: A critical review. *Applied Sciences*, 12(4).

Edstrom, J., Goodyear, J., and Kesler, H. (2013). *Learning Apache Karaf*. Packt Publishing.

Gustin, D. and Jasperneite, J. (2022). Iot device management based on open source platforms - requirements analysis and evaluation. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–4.

Jabbar, W. A., Tiew, L. Y., and Ali Shah, N. Y. (2024). Internet of things enabled parking management system using long range wide area network for smart city. *Internet of Things and Cyber-Physical Systems*, 4:82–98.

Joo, Y.-M. (2023). Developmentalist smart cities? the cases of singapore and seoul. *International Journal of Urban Sciences*, 27(sup1):164–182.

KaaIoT (2024). Kaa iot documentation. Accessed: 2024-05-20.

Kadiri, D., Pap, M., and Baletić, B. (2023). Smart cities: London, paris, barcelona, milan; definitions and strategies. *Prostor: znanstveni časopis za arhitekturu i urbanizam*, 31(2 (66)):236–247.

Lantz, B. and O'Connor, B. (2015). A mininet-based virtual testbed for distributed sdn development. *SIGCOMM Comput. Commun. Rev.*, 45(4):365–366.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):1–55.

Mershad, K. and Cheikhrouhou, O. (2023). Lightweight blockchain solutions: Taxonomy, research progress, and comprehensive review. *Internet of Things*, 24:100984.

Okhovat, E. and Bauer, M. (2021). Monitoring the smart city sensor data using thingsboard and node-red. In *2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI)*, pages 425–432.

Pacheco Pinto, G. and Prazeres, C. (2019). Web of things data visualization: From devices to web via fog and cloud computing. In *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 140–145.

Roda-Sanchez, L., Garrido-Hidalgo, C., Royo, F., Maté-Gómez, J. L., Olivares, T., and Fernández-Caballero, A. (2023). Cloud–edge microservices architecture and service orchestration: An integral solution for a real-world deployment experience. *Internet of Things*, 22:100777.

RP, J., K, R., A, A., and K, L. N. (2021). Iot in smart cities: A contemporary survey. *Global Transitions Proceedings*, 2(2):187–193. International Conference on Computing System and its Applications (ICCSA-2021).

Sahib, U. (2020). *Smart Dubai: Sensing Dubai Smart City for Smart Environment Management*, chapter 3, pages 437–489. Springer Singapore, Singapore.

Sehgal, A., Perelman, V., Kuryla, S., and Schonwalder, J. (2012). Management of resource constrained devices in the internet of things. *IEEE Communications Magazine*, 50(12):144–149.

Sousa, N. R. and Prazeres, C. (2018). M2-fot: a proposal for monitoring and management of fog of things platforms. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 01038–01043.

Taiwan Sourcing Service Provider (2015). Green ideas technology introduces blind-spot-free smart lighting control system. `http://www.cens.com/cens/html/en/news/news_inner_48111.html`. Accessed May 2024.

ThingsBoard, Inc. (2024). Device management - thingsboard. Accessed May 2024.

Wolniak, R. and Grebski, W. (2023). Smart mobility in smart city–singapore and tokyo comparison. *Scientific Papers of Silesian University of Technology. Organization & Management/Zeszyty Naukowe Politechniki Slaskiej. Seria Organizacji i Zarzadzanie*, 9(176).

Zahoor, S. and Mir, R. N. (2021). Resource management in pervasive internet of things: A survey. *Journal of King Saud University - Computer and Information Sciences*, 33(8):921–935.

Zhang, X., Noaman, M., Khan, M. S., Abrar, M. F., Ali, S., Alvi, A., and Saleem, M. A. (2022). Challenges in integration of heterogeneous internet of things. *Scientific Programming*, 2022:8626882.