Preprint:
Kumar, Vijay, Gunner, Sam, Spyridopoulos, Theodoros et al. (2023) Challenges in the Design and Implementation of IoT Testbeds in Smart-Cities: A Systematic Review. [Preprint]

Challenges in the Design and Implementation of IoT Testbeds in Smart-Cities: A Systematic Review

Vijay Kumar, Sam Gunner*, Theodoros Spyridopoulos*, Antonis Vafeas*, James Pope, Poonam Yadav, George Oikonomou, and Theo Tryfonas

***Abstract*—Advancements in wireless communication and the increased accessibility to low-cost sensing and data processing IoT technologies have increased the research and development of urban monitoring systems. Most smart city research projects rely on deploying proprietary IoT testbeds for indoor and outdoor data collection. Such testbeds typically rely on a three-tier architecture composed of the Endpoint, the Edge, and the Cloud. Managing the system's operation whilst considering the security and privacy challenges that emerge, such as data privacy controls, network security, and security updates on the devices, is challenging. This work presents a systematic study of the challenges of developing, deploying and managing urban monitoring testbeds, as experienced in a series of urban monitoring research projects, followed by an analysis of the relevant literature. By identifying the challenges in the various projects and organising them under the V-model development lifecycle levels, we provide a reference guide for future projects. Understanding the challenges early on will facilitate current and future smart-cities IoT research projects to reduce implementation time and deliver secure and resilient testbeds.**

***Index Terms*—Internet of Things, Smart Cities, Urban monitoring architecture**

I. Introduction

URBANIZATION has raised the demand for natural resources in cities, while increased pollution has also increased environmental impacts. As cities grow, the logistics to ensure the provision of essential services becomes more challenging for city councils [1, 2]. To improve city management and allow the development of relevant services, councils monitor city parameters such as air quality, road traffic, pedestrian movement, electricity usage, etc. Emerging technologies such as Internet of Things (IoT) provide the ability to understand the physical environment with more granular data and allow citizens and city councils to make better decisions. The opportunities offered by IoT implementations are numerous. For example, monitoring Volatile Organic Compounds (VOCS) and equivalent CO2 (eCO2) in households can help residents with long-term lung conditions (such as asthma) identify poor air quality and act accordingly.

City councils, in association with researchers and universities, participate in multiple research projects such as SPHERE [3, 4], REPLICATE [5] and Twinergy [6] that aim to improve energy use, mobility, human well-being and productivity, reduce energy footprint, and increase resilience [7] and sustainability of the city [8].

The Array of Things (AoT) team [8] has conducted various workshops with multi-disciplinary academics and citizen communities to understand how IoT technology comprising sensors, cameras, and computation capabilities can help modern cities. They concluded that scientific instruments (endpoint/edge IoT devices) deployed in an urban environment to provide spatial and temporal sensor data for analysis could benefit residents and city councils. Their emerging IoT platform ultimately forms an urban-scale instrument for research and development [8], simultaneously testing new sensors, communication, and computing devices.

Edge devices deployed in public spaces can also be used to test and support new technologies such as Vehicle-to-Infrastructure (V2I) communication in Co-operative is Intelligent Transportation Systems and Augmented Reality (AR) to display city information.

However, the development and management of urban monitoring systems pose many challenges. The collection of citizen data can lead to privacy violations if they are not properly managed [9]. The complexity of such systems and the integration of heterogeneous and in many cases, proprietary technologies further increase the data management problem and can also result in security issues that may ultimately disrupt services [10, 11, 12].

This paper aims to systematically identify the challenges in developing urban monitoring IoT testbeds based on the authors' experience in relevant Urban Observatory (UO), smart city projects, and the analysis of the relevant literature. These projects include: Harbourside water quality monitoring [13, 14], Clifton Suspension Bridge structural health monitoring [15], e-bike monitoring [16], damp residential detection [17] and Smart Citizen Kit (SCK) deployment in the Cotham Hill Pedestrianisation Programme, as well as others [5, 6, 18]. Table 1 lists the different research projects in which the authors were involved and provided their experiences. Table 2 lists similar smart city projects that the authors referred to understand the challenges faced in the projects mentioned in the research publications. We hope this work will benefit the design and implementation of future

V. Kumar, S.Gunner and T.Tryfonas are with the Department of Civil Engineering, University of Bristol, UK, e-mail: first.lastname@bristol.ac.uk.

J. Pope and G. Oikonomou are with the Department of Engineering Mathematics and Electrical and Electronics Engineering, University of Bristol, UK respectively, e-mail: first.lastname@bristol.ac.uk.

T. Spyridopoulos is with the School of Computer Science and Informatics, Cardiff University, UK, e-mail: spyridopoulost@cardiff.ac.uk.

P. Yadav is with Computer Science Department, University of York, UK, e-mail: poonam.yadav@york.ac.uk.

A. Vafeas is with Computer Science Department, University of Bristol, UK, e-mail: vafeas.2011@my.bristol.ac.uk.

* S. Gunner, T Spyridopoulos and A. Vafeas contributed equally to this work

smart cities research projects and IoT testbeds, reducing the implementation time.

The rest of the paper is structured as follows: § II provides the background knowledge of smart cities research projects, testbed and monitoring architecture. § III provides a brief about literature review. § IV provides the methodology followed to understand the challenges. § V provides the challenges faced by the research projects mapped to the V-model. § VI concludes the work.

II. Background

A. Smart Cities Research Projects

Multiple organisations collaborate with the city council to make a city smart and work on research projects to improve citizen lives and city council services. Smart city research projects can target different areas, for example, collecting environmental data to monitor air, noise, water pollution, residential dampness, energy monitoring, or structural health monitoring of buildings and bridges. Table 1 and Table 2 provide a list of innovative city projects, the data they collect, their architecture, and the size of the deployment. Multiple smart city research projects deploy testbeds to collect urban or citizen health data for different analyses. Major implementations occur in public places or citizens' homes. In public places, there have been multiple projects such as Smart Santander [38], UMBRELLA [39], and AoT [8], whereas projects such as SPHERE [3, 4] and REPLICATE [5] have deployed devices in citizen homes. Smart Santander deployed multiple IEEE 802.15.4 devices, General Packet Radio Service (GPRS) modules, and joint Radio-frequency identification (RFID) tag/Quick Response code (QR) code labels deployed at both static locations (streetlights, facades, bus stops) and mobile vehicles (buses, taxis) for different smart city use cases. Similarly, UMBRELLA deployed multiple edge nodes mounted on lamp posts containing wireless radio nodes and sensors, providing a real-world platform to test wireless algorithms and smart city sensing (temperature, air quality, and noise). The AoT project deployed edge nodes in Chicago to collect real-time data on the city's environment, infrastructure, and activity for research and public use. SPHERE deployed a multi-modal platform of non-medical home sensors to serve as a prototype for future residential healthcare systems. REPLICATE deployed edge devices to deploy energy efficiency, mobility, and Information and Communications Technology (ICT) solutions in city districts. Twinergy has installed house batteries and smart plugs in people's houses to improve their self-consumption of locally generated renewable energy and monitor their uptake of energy demand side management.

B. A Brief About Testbeds

Testbeds play an essential role in experimental research by allowing researchers to perform experiments, deploy multiple devices, set up realistic environments, and collect sensor data and insights [40, 41, 42]. The testbeds are made up of endpoints (sensors that sense the physical parameter), edge gateways (collect and process data from endpoints) and cloud infrastructure (collect and process data from endpoints/edge).

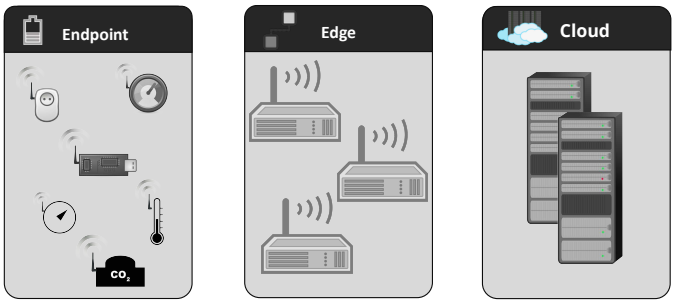

Figure 1: Typical three-tier architecture for a smart city

Managing such an infrastructure is challenging [26]. The challenges include the security and management of multiple devices, data security and privacy, user privacy controls, visualisation, multitenancy of applications, hardware malfunctions, programming bugs, software incompatibilities, network resilience, and plain misunderstanding of concepts [26, 43]. Furthermore, each research project implements the testbed differently based on the project team's requirements, usability, budget, time, and technical skillset.

A testbed should enable researchers to **i.** deploy and connect multiple devices at the edge and endpoint tier safely and securely, **ii.** deploy applications on the cloud and edge devices collecting and processing data from the endpoints (sensors) and sending it securely to the edge/cloud, **iii.** manage the devices for accounting and administrator purposes **iv.** provide data visualisation and insights to end users [44] and **v.** be adaptable to fulfill other requirements. The authors categorised the testbed into three different categories **i.** Distributed large-scale cloud resources testbed providing researchers the access to the bare metal and control over computing, storage, and networking resource, e.g., Chameleon [45], GENI [46], GRID5000 [47], FED4FIRE/FED4FIRE+ [48], FIT-Cloud [49], Emulab [50], PlanetLab [51], PRAGMA [52], DETER [53], NOR-NET Core [54], SAVI [55] **ii.** distributed large-scale endpoint Wireless Sensor Networks (WSN) testbed that provide access to the WSN nodes to conduct network experiments, i.e., FIT IoT-Lab [56], SmartSantander [38], City of Things [57], UMBRELLA [39] **iii.** data-collecting research testbed that collects data from citizen house or public spaces, i.e., SPHERE [3, 4], REPLICATE [5], Twinergy [6], 3E Houses [58], SONYC [28], AoT [8], Scallop4SC [59], Padova [29].

C. A General Three Tier Architecture

Testbeds can have different architectures based on the project requirements, such as endpoint-cloud, endpoint-edge, and endpoint-edge-cloud. In endpoint-cloud, devices at the endpoint tier communicate directly with the cloud tier; in endpoint-edge, the endpoint sends the data to the edge, and the cloud tier does not exist. In endpoint-edge-cloud, endpoints connect directly to edge devices, and devices at the edge tier connect to the cloud tier. Endpoint-edge-cloud is a standard architecture used by different projects such as SPHERE [3, 4], REPLICATE [5], Clifton Suspension Bridge [15], AoT [8]

Table 1: Research projects in which the authors participated. CS: Cloud Server, EN: Edge Node (Gateway), EP: Endpoints (IoT Node). CS may contain all or a subset of open-source components (Kafka, K3S, MQTT, InfluxDB, Grafana). EN may consist of SBC (RPi or Intel NUC)

Project	Size (Where)	Data collected	Architecture				
SPHERE	(100 Homes), (1 EN; multiple EP)/home	Environmental	EP (802.15.4)	→	EN(4G)	→	CS
UMBRELLA	200 EN (streetlamps) with on-board EP	Environmental, Camera	EP (I2C/SPI)	→	EN (Fibre/WiFi)	→	CS
Cotham Hill Pedestrianization	10 EP in (8 homes)	Noise and air pollution	EP (WiFi)	→			CS
Residential Dampness	(1 home), (1 EN with on-board EP)	Temperature, Humidity	EP (Analog)	→	EN		
Clifton Suspension Bridge	1 EN, 2 EP	Structural health monitoring data	EP (802.15.4)	→	EN (4G)	→	CS
Water quality monitoring	3 sites (1 device with 7 sensors)	Water quality	EP (Serial to WiFi)	→		→	CS
SYNERGIA	3 ENs, 15 EP (office)	Environmental	EP (802.15.4/LoRa)	→	EN (LAN)	→	CS
REPLICATE (Energy)	Smart appliances (151 Homes);	Energy consumption	EP (LAN)	→	EN (LAN)	→	CS
REPLICATE (eBike)	EN (12 e-bikes)	Battery level, motor power	EP (CAN)	→	EN (LoRa/WiFi)	→	CS
Bristol AoT	3 EN with on-board EP	Environmental, Camera	EP (I2C/SPI)	→	EN (4G)	→	CS
Twinergy	12 home	Energy consumption data	EP (WiFi)	→		→	CS
EurValve	(40 homes), (4 EN; 1 EP)/home	RSSI and accelerometer data	EP (Bluetooth)	→	EN(4G/WiFi)	→	CS

Table 2: Research projects referred by the authors. Based on the details provided in the papers. CS: Cloud Server, EN: Edge Node, EP: Endpoints.

Project	Size (Where)	Data collected	Architecture				
AoT [8]	130 EN (streetlamps) with on-board EP	Environmental, Camera	EP (I2C/SPI)	→	EN (4G)	→	CS
e-Agriculture [19]	EN (Lab deployment)	Light, temperature, soil pH and humidity	EP (Analogue)	→	EN		
Living Labs [20]	150 EN, 800 EP (120 location)	Air quality, microclimating, bat monitoring	EP (RPL)	→	EN (2G)	→	CS
Connected Vehicle Testbed [21]	3 Fixed EN (FEN), 2 Mobile EN (MEN)	Vehicle position data	MEN (wireless)	→	FEN (wired)	→	CS
Wireless Environmental Sensors [22]	1 EN, 7 EP (Lab deployment)	Environmental	EP (Bluetooth)	→	EN (LAN)	→	CS
Solar-powered WSN [23]	82 EP (real-world deployment)	Temperature, RSSI, battery level	EP (WSN) w/ sink	→			
Community Elderly Care [24]	EN, EP (70 elderly homes)	Motion, door contact	EP (Z-wave)	→	EN (cellular)	→	CS
IEEE802.15.4 Connectivity Traces [25]	350 EP (Office environment)	RSSI, PDR	EP (802.15.4) w/ sink	→			
LOFAR-Agro [26]	109 EP, 3 EN, (real-world deployment)	Temperature, humidity	EP (WSN) w/ sink	→	EN (WiFi)	→	CS
3E Houses [27]	(100 homes)(6 EP/ 1 EN)/home	Energy consumption data	EP (Zigbee)	→	EN (WiFi)	→	CS
New York Noise sensor network [28]	55 Nodes (1 EP and 1 EN)/node	Noise data	EP (USB)	→	EN (WiFi)	→	CS
Padova Smart City [29]	1 EN, 8 EP	Temperature, humidity, benzene	EP (802.15.4)	→	EN (WiFi)	→	CS
Flash Flood Monitoring [30]	3 iter. of IoT device deployed; EN, EP	Water levels	EP (USB)	→	EN (cellular)	→	CS
Smart Santander [31]	50+ EN, 700+ EP	Environmental	EP (802.15.4)	→	EN (Wired/Wireless)	→	CS
City of Things [32]	32 locations (1 node/location;multi-radio)	Air quality, traffic monitoring, parking	EP (WSN)	→	EN (multi-radio)	→	CS
SADMote [33]	5 EN, 12 EP	Environmental	EP (WSN)	→	EN (WiFi)	→	CS
SensorScope [34]	≈6EN, each serving ≈100 EP	Environmental	EP (WSN)	→	EN (GPRS)	→	CS
EpiFi [35]	≈ 18 locations (2 EP, 1EN)/location	Environmental	EP (WSN/WiFi)	→	EN (WiFi)	→	CS
Parking System [36]	2 EP, 3 EN	Parking, Light sensor	EP (Lora)	→	EN (Lora receiver)		
Residential Sensing [34]	≈ 20 homes ≈ 1200 EP	Temperature, light, door	EP (Z-wave)	→	EN (WiFi)	→	CS
Water consumption [37]	30 homes, 1EN, ≈ 4 EP	Water consumption	EP (433MHz)	→	EN (WiFi)	→	CS

and others [30, 60] and also mentioned in relevant literature [43, 61].

Fig. 1 presents a typical architecture of a data collection testbed consisting of cloud, edge, and endpoint tiers. We provide a brief introduction about each tier below:

Endpoint Tier: The endpoint tier consists of resource-constrained, battery-powered embedded devices with low-power wireless communications capability. The devices are generally inconspicuous and have a small nominal form factor for deployment in space-constrained environments [62]. They can sense different environmental parameters such as barometric pressure, temperature, humidity, light, motion (with an accelerometer, gyroscope, or compass) and presence (using an infrared sensor to detect the human body's heat). In addition, a reed relay or switches can sense the opening/closing of a window/door. Endpoints generate monitoring data and send it to a collection point at the edge/cloud tier for processing and analysis. The endpoints can be connected to the edge/cloud tier by different technologies such as **i.** an IEEE 802.15.4 network (in a mesh or star topology) created and controlled by an edge tier device, **ii.** low-power wide-area network (LPWAN) network technologies (Sigfox, LoRaWAN, NB-IoT, Wi-Fi, Bluetooth Low Energy (BLE)), **iii.** directly connected to the edge device using a Universal Serial Bus (USB), Inter-Integrated Circuit (I2C), Serial Peripheral Interface (SPI), Universal Asynchronous Receiver/Transmitter (UART).

Edge Tier: The edge level can consist of a single-board computer (SBC) (Raspberry Pi (RPi), Jetson Nano (JN), Grapeboard, Intel NUC) installed in a citizen's home or public spaces (street lamps, bus stops, city council vehicles) or private buildings [63]. The edge tier collects the data sent by the endpoints and either process it or sends it in a raw format to the cloud tier [64] for further analysis. Processing data at the edge reduces payload size and communication bandwidth, shortens latency, and simplifies data formatting and aggregation for the cloud [65]. The edge device can also run different applications, such as urban environment monitoring and counting people/vehicles, and is often designed to be application-agnostic. It provides end users with a sensing/processing element at the network edge that can service novel applications. Edge devices are typically connected to the cloud tier using higher bandwidth and more reliable communications technologies, such as 4G/5G, Wi-Fi, and fibre. According to the project requirements, the edge device can contain multiple radios onboard, such as IEEE 802.1ac on 2.4/5 GHz, DASH7 on 433/868 Mhz, BLE, IEEE 802.15.4, IEEE 802.15.4g, and LoRa [32]. Edge devices (based on their location) can also be used in infrastructure mode (endpoints connecting to edge tier) or ad hoc peer-to-peer (edge devices connecting to each other using radios).

Cloud Tier: The cloud tier consists of multiple servers, hosting all the applications and services required to manage

the devices at the edge, endpoint tier and the applications necessary to achieve the project objectives [64]. The servers will run multiple components on virtual machine (VM) or containers. The cloud tier can be hosted privately (OpenStack, VMWare) or on commercial cloud services (Azure, AWS). It contains the application logic and services required to operate and manage the testbed platform. The cloud tier should provide different services to the edge tier, such as credential management, data storage, provisioning of devices, networking, time synchronisation, secure remote software updating, configuring, and maintaining access to the edge and endpoint devices. It should also provide a secure communication channel to devices and services in the edge and endpoint tiers.

With the advancements in core networks, part of the functionality is distributed from the cloud tier to multiple geographical locations towards the edge network. In this case, the main point of distinction becomes Radio Access Network (RAN) and how the endpoint tier is connected to the cloud tier. Depending on the wireless and wired transmission network, some core network features, computation, and offloading can occur on the edge tier. Therefore, it is vital to address the challenges of edge-to-cloud connectivity and the architectural decisions that each testbed has chosen.

III. Literature Review

Santana et al. [66] surveyed multiple smart cities projects in the area of cyber-physical systems, IoT, Big Data and cloud computing. They provided challenges and open research problems in developing next-generation, robust software platforms for smart cities. They included privacy (data owners, data usage), data management (storing, processing a large amount of data and trusting it), heterogeneity (different devices, data sources), energy management (energy consumption failure), communication (network reliability), scalability (increase in the number of users, devices, data), security (safe from cyber-terrorism and cyber-vandalism), lack of testbeds, city models (effective and efficient city model), platform maintenance (mange devices).

There have been multiple lessons learned papers regarding deployment of battery-powered devices in the endpoint tier communicating over IEEE 802.15.4 [25, 26, 33, 67, 68], devices deployed in public spaces [8, 69], citizens houses [24, 67, 70, 71, 72], testbeds [73, 74], and others [20, 21, 22, 23, 75, 76, 77, 78, 79, 80]. However, the lessons learnt are from a project with specific requirements such as "challenges in flash flood monitoring" or does not include human engagement (deployment in citizen houses or public spaces) or has very small-scale deployment. They do not cover every stage or all the challenges faced during a smart-city research project. Our work focuses on a three-tier architecture (cloud, edge and endpoints) usually used in smart-cities research projects. It covers the end-to-end perspective and the challenges faced during the research project, from the requirement analysis, system design, implementation, and integration testing to the final deployment stage, keeping security and scalability in mind. It is more expansive and covers a larger scope, providing learnings from deploying multiple smart-city research projects.

IV. Methodology

The methodology authors used to understand the challenges faced in the smart-cities research project is interview-based and literature review-based. To collect the necessary data for our research, we conducted semi-structured interviews with the system architects and the deployment team of European research projects on IoT platforms and testbeds for urban monitoring [5, 6, 13, 15, 17, 18]. Table 1 lists the different research projects authors were involved in and provides their experiences. The interviews focused on the challenges the participants faced during the development, deployment and management of the IoT platforms and testbeds. The author asked simple open-ended questions with a free-flowing approach by asking a set of questions to the interviewee, and the conversation was continued based on the answers. The questions asked were about the challenges faced, such as "*What are the challenges faced during the projects*", "*How did we provision the devices*", "*What was the architecture of the project*", "*How did the devices communicate with each other*", "*How did we manage the storage, credentials*", "*Any challenges faced in the implementation, deployment*", "*What could have made your (system architect) life better*", "*any unexpected challenges*". The authors captured additional challenges based on their reflections on their experiences as members of smart-cities projects.

Moreover, we thoroughly reviewed the relevant literature on infrastructure deployment. Table 2 provides a summary of different projects that authors referred to understand the challenges faced in projects deploying IoT infrastructure.

To facilitate the exploitation of our work by future projects, we categorized the identified challenges based on the stage of the project lifecycle they appear. Almost all engineering projects follow a similar development lifecycle, from "requirement analysis" and "system design" to "integration and testing" and final project delivery. In our work, we identify the challenges in the various project and organize them under the V-model's level to formalize the development process and provide a reference guide for future projects.

A. The V-model

The V-model is one of several project life cycle models developed over time. Project life-cycle models try to visualise and map the different stages of a technology development project. They are an essential tool for the engineer and provide a standard conceptual framework of reference [81].

The V model [82] is based heavily on 'the waterfall model' [83] that preceded it but increased it by projecting the project cycle into a three-dimensional space. Fig. 2 shows the first two dimensions of this space, x and y, representing 'time' (or project maturity) and 'Design Detail', respectively. The 'Design Detail' axis has high-level design at the top and low-level (or detailed) design decisions at the bottom. The central elements of the model (referred to as the core of the Vee) are shown in blue. The specific phraseology used in these elements varies depending on the particular application. However, the general theme is always the same: as the project works down

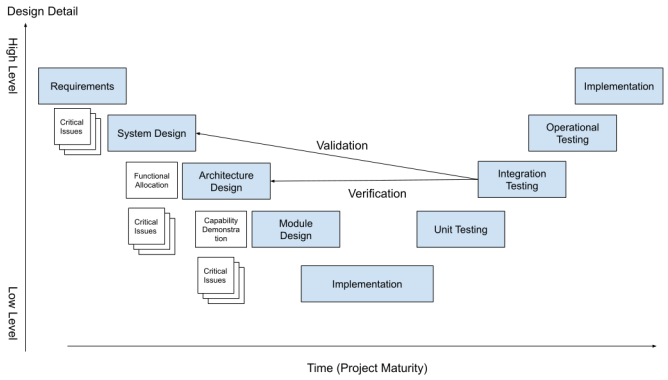

Figure 2: A graphical representation of the V-Model (modified from [84])

the left arm of the V, high-level system designs are converted into more detailed system designs.

One failure of the waterfall model was its embargo on any detailed design work before official approval of high-level design decisions [84]. The V-Model removes this restriction, allowing the detailed technical enquiry to inform higher-level decisions. This 'off-core' work can be seen in the white boxes below the core of V. The work in the off-core varies depending on the design stage, but it aims to derisk the decisions currently being made. Important off-core work [84] is the identification of 'critical issues'. Capability demonstrations are also important, demonstrating that technology can perform the desired functionality before it is written into a specification.

The final dimension of the V-Model, the z-axis pushing into the page, represents the different system design elements at that level of system decomposition. For example, architecture will consist of many modules, and each must be designed, so the V-model fans out to represent this, one branch for each module. Below the V, the z-axis represents the different and competing design options that must be evaluated before a selection can be made. The workflow moves down the left-hand side of V until the bottom is reached, which means that design decisions are completed and can now be implemented. This means that each piece of hardware can be built and each software package written.

Integrating these different components is necessary to form the final functioning system. It is performed by moving back on the right-hand side of the V. Each module is tested against the design from which it was created and then integrated with other modules to deliver more sophisticated functionality. That functionality is then tested against the higher-level design. Not only is 'verification' carried out (confirming that the module has been built according to its design specification), but 'validation' is also performed to ensure that the design captures the system's requirements.

V. Challenges

This section discusses the challenges captured. We use the V-model to classify challenges and map various phases of the research project. Fig. 3 summarises the challenges faced during different stages of the research project assigned to the V-model phases. Challenges can be categorised into multiple phases of a smart city research project, from understanding project requirements (requirement analysis) to designing how to fulfil those requirements (system design) and setting up defined infrastructure (implementation) to ensure that different infrastructure components work together (integration) and tested in the laboratory and initial small-scale deployment (operational testing) followed by deployment in the real world and operational challenges.

A. Requirements Analysis

The requirement analysis stage helps to understand the application and data requirements, collaboration dependency, and project use cases.

Application/Data requirements: Data is at the heart of urban monitoring research projects. Depending on the need, it can be collected from multiple sensors deployed in citizens' houses, streetlamps, or bus stops. The nature of data required to meet project objectives and expected results affects, in general, all aspects of the project, from the technology to be used to the security implications of the privacy achieved [85]. For example, in the SPHERE project, researchers created bespoke wearable devices with multiple components, many of which (e.g. second acceleration sensor, gyroscope, non-volatile flash memory, LED, button) were never used in real deployment [67]. During the REPLICATE and Twinergy project, it was found that it is essential to engage with the stakeholders of the project (e.g., the city council and citizens) at the beginning of the project, clarify their expectations, understand their needs, and translate them into requirements for data collection, processing, storage, sharing, and visualisation [63].

Once the type of data is clarified, it is essential to consider the relevant General Data Protection Regulation (GDPR) [86] implications. In the UK, the Data Protection Act 2018 implements the European GDPR. The Act introduces the terms "data controller" and "data processor" and clarifies the responsibilities around personal data collection, processing, and storage. These considerations will influence the system's design (e.g. employ mechanisms to ensure secure data collection, data anonymisation, or data destruction) and final deployment (e.g. deployment only after citizens' consent) in the subsequent development process steps. For example, the SPHERE project stored raw sensor data related to health in an external Linux Unified Key Setup (LUKS) encrypted solid-state drive (SSD) [87].

Great care must also be taken to ensure that the collected non-personal data cannot be used to infer information about individuals. For example, environmental/energy data can reveal citizens' behaviour and habits when not handled appropriately. Depending on the entities involved in the project, different actors may be interested in ensuring compliance with GDPR. Universities undergo an ethical approval process that involves a rigorous analysis of relevant implications and solutions. City councils may require a privacy impact assessment that describes the data the project aims to collect, potential privacy issues, and the related impact.

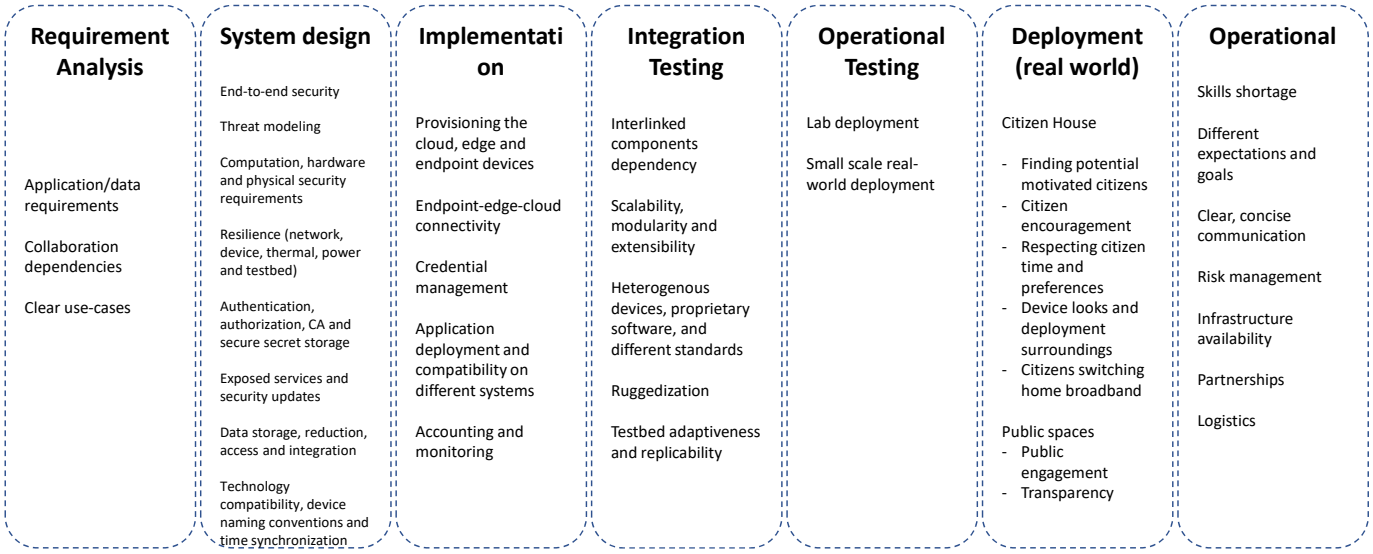

Figure 3: Summary of challenges in smart-cities research projects

In addition to legal implications around data collection, special care must be taken to clarify, understand, and comply with contractual agreements (e.g. data-sharing agreements) among the project's partners. The partnership agreement should detail the data each partner aims to collect, share, or process and the purpose of this activity, including potentially generated intellectual property and monetisation. This information should also be considered when considering the project's GDPR implications.

Another data-related requirement that must be addressed in the early stages of a project is the need to integrate the data collected by the platform into other existing city data platforms (such as Bristol Open Data [88] and London Datastore [89]). Capturing integration requirements with external systems early on ensures the use of appropriate technologies and the timely delivery of the project.

Stakeholders must agree on the data requirements to ensure that the system's development follows user needs.

***Collaboration dependencies**:* Urban monitoring research projects often involve multiple partners (e.g. universities, city councils, industries) and require collaboration between different departments between partners (e.g. IT support, estate team). For example, project servers are usually behind the university or company firewall. The opening of ports on the firewall can take a considerable amount of time, ranging from weeks to months. The process may require multiple approvals from different entities and involves cyber security risk assessments to understand the various threats to the system and identify possible mitigation techniques. In projects with multiple collaborators, it is essential to consider these interactions and dependencies and address them during the requirements analysis period of the project.

***Clear use cases**:* Once the requirement collection has been completed, the project team must develop use cases that address the requirements [63]. Below, we provide a few examples of use cases in urban monitoring projects:

- **Use case for deployment of sensors in Citizen Home (Indoor)**: Sensors provide details about indoor pollution and help citizens take action, such as opening windows for cross-ventilation.
- **Use case for deployment of sensors in a commercial building**: Assuming that a corporate building consists of multiple floors/rooms, the building management team can consist of a Heating, Ventilation, and Air Conditioning (HVAC) team, estates teams, admin team, fire safety, and different companies occupying the offices/floor. Data can be sent to different teams depending on their requirements. For example, temperature data to the HVAC team to ensure the optimal temperature in rooms/offices; battery data to the estate's teams to ensure that the sensor batteries are replaced on time; air quality data and occupancy data to respective companies on respective floors.
- **Use case for deployment of sensors in public spaces**: When sensors are installed inside citizen homes, outdoor data (vehicle traffic, pedestrian traffic, light levels, weather, atmospheric conditions) can be compared with indoor data and provide context [90]. The sensor data can validate and train the various micro-climate weather models. Citizens can also use noise and air pollution data to decide on the suitability of buying a house in the neighbourhood

B. System Design

The V-model system design phase provides a system overview, details of the different hardware, software, network protocols, applications, and logical components in the three-tier architecture mentioned in § II and the interfaces between them. It allows system architects to define testbed requirements from the perspectives of resources, security, resilience, data, and technology. System design decisions must be based on project requirements, and the requirements can always be referred back to understand and justify the design as specified

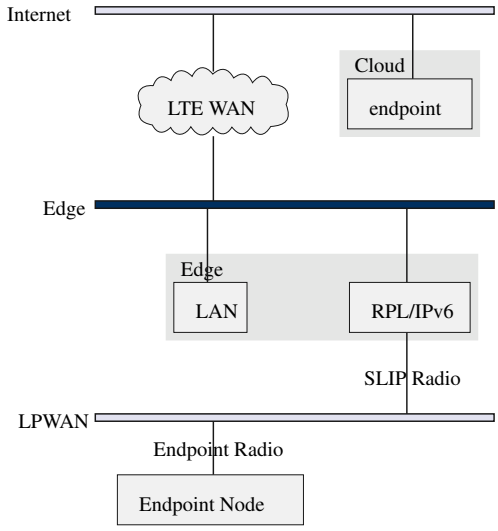

Figure 4: Local and remote threat models originate from the bottom up and up to bottom respectively

in the V-model. For simplicity, the architecture and module design are merged into the system design.

***End-to-End Security**:* Securing a testbed from end-to-end (endpoint, edge, cloud tier) is challenging. It includes the security of all devices at each tier and the communication between them, including physical and data security. Endpoint-Edge-Cloud or End-to-End testbeds should be secure by design and provide fundamental security blocks such as confidentiality, integrity, availability, and non-repudiation [76, 91]. Confidentiality requires data protection from unauthorised people; Integrity requires protecting data from being altered; Availability requires ensuring access to data to authorised users when needed; Non-repudiation requires an assurance that authentic communication requests cannot be denied. A chain of trust is established by validating each process and component of hardware/software from the base up to the final system, including the design, manufacture, and supply chain. A dependency graph (chain of trust) can be created by examining the component and services in which one trusted layer establishes the trust in the next by validating it and providing the core trusted functions on which it depends. Any security weakness at a lower level compromises the security of the higher levels dependent on it. This results in an untrusted base that compromises trust in the system. The roots of trust for a system are levels of trust origin – the root of the chains of trust. The roots would be the hardware or hosting environment, the Operating System (OS) and any applications, libraries, and compilers. For building a system in secure environments, the roots may be the factories and supply chains for the hardware, the software design processes for the libraries, the location of manufacture, the supply chain and delivery. Additionally, the data collected by the testbed can be sensitive such as patient health and environmental data. Processing sensitive data using data analysis and machine-learning techniques [92] makes the testbed a target for cyber-criminals [93] and adversarial machine-learning attacks [94].

***Threat modelling - "Identify threats, threat actors and determine risk acceptance"**:* Security of the testbed and the data collected is important. In projects that collect sensitive data, it is essential to understand the various threat actors, attacker models, and risks involved [95] that could compromise the security of the collected data or the testbed [96]. Creating a threat model is a crucial and challenging part of a research project and should be performed at the beginning of the project. It helps identify threats, attacks, vulnerabilities, and countermeasures that may affect the testbed infrastructure and its components. It can be performed in five significant threat modelling steps: defining security requirements, creating an infrastructure testbed diagram, identifying threats, mitigating threats, and validating that threats have been mitigated [97, 98, 99]. Threat modelling will enable testbed administrators and architects to communicate about the security design of the testbed, analyse those designs for potential security issues, and manage mitigations for security issues.

An example of architectural consideration of threat models for our three-tier approach is presented in Fig. 4. Some key security questions arise, particularly regarding the edge and endpoint interaction. Serial to IP (SLIP) bridges with the coordinator endpoint node require a multi-role endpoint node which requires separate firmware and networking behaviour for each node. The SLIP radio is the same hardware as other endpoint nodes but needs its firmware to be developed hand in hand with the edge device networking implementation to maximise security. The computing resources of the endpoint are minimal; therefore, communication with external devices must be tested with radio connectivity in full operation.

Fig. 4 also presents the concept of an edge network. Many challenges arise from the inability of the edge network to extend beyond a single computer (i.e. tunnel interface on a single RPi SBC). In this case, it is difficult to distinguish between the edge and the cloud and their interfaces. A strong firewall must be implemented and the network separation between Local Area Network (LAN) and Wide Area Network (WAN) must be enforced at the edge site.

***Computation, hardware and physical security requirements**:* Based on the use cases and the required functionality, it is essential to determine the computation capabilities (memory, storage, Central Processing Unit (CPU)) at each tier [9]. For example, cloud tier servers have high resources such as memory (8+ GB RAM), CPU power (multiple cores), and network (Internet speed 50 + MB). Edge devices are SBCs and have fewer resources (1-8 GB RAM, single or dual-core CPU) than the cloud tier. On the other hand, endpoints are typically low in power consumption, memory (128KB-2MB of programmable flash and 20-512 KB of volatile RAM), and processing power (Arm Cortex-M Microcontrollers (MCU)) [25].

Furthermore, devices at each tier should provide hardware security features such as a cryptoprocessor (Trusted Platform Module (TPM)), the hardware-based root of trust that allows secure boot, secure firmware, secure credentials storage, and an encrypted file system. Secure boot prevents the loading of unauthorised software onto the device during the boot process; Secure firmware ensures that only authorised code (signed images) from the manufacturer is booted. Secure boot and

firmware update capabilities ensure that the device does not run unauthorised or malicious code. Crypto-processor with a random number generator enables cryptographic functions such as encryption, decryption, and key generation for security purposes. However, generating random numbers in constrained embedded systems is a significant challenge due to the lack of resources and entropy. Modern endpoints provide a way to protect the integrity by providing a physically write-protect non-volatile memory with a mechanical switch. The end-user can switch to write-enable the memory for firmware update and then write-protect the device once the update is complete.

Furthermore, the physical security of the devices is essential, as they can contain confidential data such as Personally Identifiable Information (PII), log-in credentials and network information. An attacker who can gain physical access to devices can compromise and steal confidential information. Cloud tier servers hosted inside a secure perimeter (company offices) are physically more secure than the devices at the edge and endpoint tier deployed in the field (citizens' houses, bridges, streetlamps, or roadside). A determined attacker can reach the physically insecure edge and endpoint device and compromise its security. For example, an attacker having physical access to the edge device that contains an SBC (e.g. RPi) can easily remove the Secure Digital (SD) card and read its contents containing confidential information such as passwords and data. To provide another example, the AoT node (deployed on out-of-reach streetlamps) exposes a serial cable wrapped in a protected rubber cover connected to the UART of the SBC. It can provide access to the device enabling the node's root access and allowing access to the filesystem and possibly confidential data. During the AoT project, it was found that it is essential to place edge and endpoint devices outside public reach (where possible) and protect them with spikes, locked cabinets, and tamper-proof casing.

However, once attackers have physical access to the edge and endpoint device, they can physically manipulate it to compromise them. The edge and endpoint tier devices have a large attack surface area, such as exposed copper vias and unused connectors, such as serial/Joint Test Action Group (JTAGs) used for debugging. An attacker can extract confidential data and embedded firmware code from the device using physical probing signals on the exposed interfaces. Most endpoint devices contain a sticker detailing the hardware components that can provide additional information to hackers. Devices with adequate physical and hardware security make it difficult for attackers to compromise them.

Resilience (network, device, thermal, power and testbed): Edge and endpoint nodes deployed in citizen houses or public spaces connect to the Internet and cloud via home broadband, Fibre, 4G, or Wi-Fi. The average downtime of broadband per year ranges from 25.4 to 168.9 hours in the UK [100]. Suppose the edge and endpoint device sends the endpoint data directly to the cloud tier without storing it locally. In this case, data will be lost due to lack of network connectivity [20, 35, 70]. Furthermore, applications also suffer from latency problems [79] depending on the quality of the network. It is essential to have network resilience (multi-network such as Wi-Fi, 4G, LPWAN) built into the device to handle network loss and latency issues.

Furthermore, there can be scenarios where the edge node becomes unresponsive, does not connect to cloud services, and cannot be accessed using Secure Shell (SSH). In such cases, building resilience on edge devices is good. For example, AoT [8] implemented a waggle manager to monitor the health of the SBC (temperature, current draw, digital heartbeat), enclosure internal temperature and humidity. It supports changing the boot medium from SD card to Embedded MultiMediaCard (eMMC) and allows a hard and soft reset of onboard sensors. Rebooting the device often solves most problems [101]. In such cases, a mechanism to reboot the device remotely is required. For example, if the edge device has multi-network connectivity (LoRaWAN, Sigfox, NB-IoT) and is not responding, the cloud tier can use LPWAN to send a downlink packet destined for that device, instructing it to reboot the system. NFC or magnetic devices can be used to cold-reboot the device without opening the enclosure (helpful for cold-rebooting publicly deployed devices) [31]. If the devices are powered by Power-over-Ethernet (PoE), the ability to remotely turn the device on and off PoE is preferable. The edge device should also be able to operate independently if cloud tier services are unavailable due to network issues [8].

Edge and endpoint devices generally run 24×7 and are usually deployed on citizens' premises or streetlamps. Suppose that a processor-intensive application is performed on the endpoint or edge, and the amount of processing power on the device is not regulated. The device can be damaged due to overheating. For example, in SPHERE houses, the Kinect camera that captures the activities in the kitchen runs 24 hours a day, processing the data. The camera becomes quite hot, reducing the device's lifespan. The edge and endpoint device should be able to self-regulate its temperature by performing CPU throttling to reduce the temperature. For example, RPi performs CPU throttling when the device temperature reaches 60-80 degrees [102].

Another challenge is to provide electrical power to devices at the edge and endpoint tiers. Edge tier devices are usually powered by a mains or battery and must be safe from an electrical safety perspective. For example, AoT is powered and installed on the streetlamp with a 110/230V mains supply. An electrical hazard can occur should the device fall from the streetlamp or the transformer inside the device malfunction. The edge tier devices deployed on the streetlamp can be powered by PoE to reduce electrical risks. Running on the battery limits the device's capabilities. Battery lifetimes typically range from a few hours to a few days. For example, SCK kits provide a USB rechargeable battery that lasts for at least a day, depending on the sensing interval and the time to send the sensor data (after 30 seconds or 1 minute). Additionally, the use of solar panels can add resilience to power devices.

Additionally, a testbed can contain development, staging, and production environments. The testbed environment will often be compromised by an attacker creating a cyber security incident due to default credentials or misconfiguration [74]. Once the testbed is compromised, it is essential to understand the affected components, as the attacker might have installed difficult-to-detect rootkits. It is prudent to recreate

the entire testbed environment from scratch automatically. If done manually, the entire activity (setting up the VMs, configuring the applications, and ensuring that the end-to-end system is working) can take up to a week or more. To quickly recreate the testbed environment, it is essential to have version control [9], continuous integration, delivery, and automation.

Authentication, Authorisation, Certificate Authority (CA) and secure storage of secrets: Testbeds consist of multiple devices and numerous applications on the cloud or at the edge for data storage, analysis and visualisation and have multiple users/administrators accessing those applications and devices. Devices and applications should have proper authentication and authorisation, allowing trusted users to access services [103]. Authentication requires digital certificates or credentials to validate the identity of devices and users. Authorisation requires that only trusted nodes and users should be able to gain network access to the testbed. As the testbed also hosts different services (such as web servers, WebSockets, and authentication servers), it is essential to have a CA in the testbed that can be used to create public-private keys and sign certificates. Different users and devices can trust the CA to secure data transmission. Further, the testbed will need to protect stored cryptographic material. The encryption keys (public/private and symmetric) and credentials are usually hardcoded in the code or stored in files. To protect the credentials from hard coding and unsecured storage, they must be stored securely using a hardware security module or key management solutions.

Exposed services and security updates on the endpoint, edge, and cloud: Devices in each tier run multiple services (e.g. SSH, web servers) and are often insecure with weak authentication mechanisms. These mechanisms include using default passwords, running a vulnerable version, using old encryption methods, and misconfigured applications [28, 74]. The services exposed on the cloud, edge, and endpoint devices depend entirely on the project's requirements. Additionally, the greater the number of services, the greater the attack surface area for the attackers and the possibility of compromise. For example, the cloud can expose port 1194 (user datagram protocol (UDP)) and transmission control protocol (TCP) port 443 to provide Virtual Private Network (VPN) connectivity. The Grafana server (data visualisation) exposes port 3000. An edge node might expose port 1883 to allow communication with endpoint devices using Message Queuing Telemetry Transport (MQTT). The endpoints can also run a Web server [104]. As endpoints are resource-constrained, there is a possibility that they might be running a vulnerable version of the web server software.

There have been instances where attackers have compromised insecure services running at the cloud/edge tier. For example, an attacker compromised a cloud server providing authentication (Keycloak instance) running with default credentials and used the server for crypto-mining [74]. Alternatively, an internal attacker can connect to the insecure MQTT service running on the edge device and subscribe to the topics to collect the published data. Furthermore, a vulnerable application deployed on the cloud/edge poses a security risk.

However, such services and systems must be made secure by default. It is essential to ensure that there are no default passwords and that the OS, applications and firmware are configured securely and up to date. If the infrastructure contains many devices kept remotely (citizens houses, streetlamps), upgrading software/firmware is often challenging. Software updates should have rollback functionality, so the device will return to its previous state even if the update process goes wrong. Upgrading software is comparably easier than upgrading firmware. A poor firmware update mechanism can leave the device unusable when an update fails.

For endpoints, it is recommended to have Over-the-Air (OTA) functionality to allow remote upgrade and configuration for long-term deployments in urban environments [23, 33, 68]. The inability to upgrade or configure the firmware remotely means that the code/firmware must be perfect and thoroughly tested, and no new requirements can be applied. For example, the Cotham Hill Pedestrianisation Programme wanted to measure noise pollution due to pedestrianisation. However, the deployed SCK kits took sensor readings at 60-second intervals (by default) and did not capture noise pollution correctly due to the 60-second gap. The only way to reduce the reading interval was to revisit the citizen's houses and configure the settings resulting in disturbing the citizens. Remote management of the technology will minimise disruption for the participants.

Data storage, reduction, access, integration and visualisation: Research projects require data storage, analysis, and visualisation. Data must be encrypted in transit and rest at all tiers. Research projects often go through different data protection and research ethics, defining data collection and usage. The data owner's responsibility is to ensure data validity, quality, secure storage, access and maintenance, replication, processing, backup, and deletion policy. Having clear information and policies helps to ensure user privacy [105]. Policies should include what participant data will be acquired, where it will be stored, and how long it will be stored. User data should be deleted once the duration of data consent is over. However, Post Docs/Ph.D (staff joining and leaving) often manage research projects, and it becomes challenging to ensure data deletion. For example, in university-managed research projects, access to the data is usually restricted to university premises (IT services managed machines) and provided via jump host machines via different credentials, and might require hopping through multiple networks. The difficulty in accessing the data makes it challenging for the data analysis activity, resulting in researchers copying and processing the data locally, which may break user privacy and data agreements.

Further, sensitive data can attract attackers. It is ideal to identify potentially sensitive information in the collected data at the endpoint/edge tier and eliminate or limit its collection [75, 76]. Data reduction and compression methods, such as sending preprocessed data to the edge/cloud tier rather than raw data [22], can also help reduce data bandwidth and power consumption. For example, an edge tier device that measures the number of cars parked using image recognition should send only the count rather than the images to the cloud [75]. Another example would be when an endpoint only transmits the reading to the edge device when a significant change is

detected to improve the energy efficiency of battery-powered endpoints [87]. Data compression and reduction should maintain the initial data requirements required for the research project's objective.

It is a good practice to store all raw data for historical and future references [72]. As users frequently access the collected data of the last few days, it is a good practice to separate current and historical data for better application performance [24]. For example, 3E houses executed SQL queries on the sensor data recorded. Over time, the query response time changed from < 1s to > 8s, resulting in an unresponsive display leaving citizens less engaged [27].

From a data integration perspective, the platform should be able to integrate data streams from multiple heterogeneous data sources [106, 107, 108, 109]. Using similar data formats will allow better data interoperability [79, 85, 110]. Further, the testbed should provide an open Application Programming Interface (API) for the end-users/developers to access the data and build applications on top of that [21, 69, 79, 87, 111]. Furthermore, data transfer from the endpoint to the edge to the cloud should be reliable with minimal data loss [30, 35]. During the AoT and Cotham Hill Pedestrianisation project, it was found that providing flexible data query capabilities for users (such as extracting specific periods or a subset of measurements/nodes) is essential. Such capabilities allow the user to monitor conditions over a particular period, such as an ongoing event (e.g. a festival, severe storm, or emergency), and stream data to specific stakeholders (city-council/car-parking and others). Data should also be visualised for stakeholders using different methods (maps, line/bar charts, dashboards and others) [103].

Technology compatibility, Device naming conventions and Time synchronisation: The testbed comprises multiple components, including hardware, software and OS, to support various services such as data storage, analysis, visualisation, authentication, and authorisation. In addition, there could be different hardware platforms such as amd64, armhf (32 bits), arm64 architecture CPUs, graphics processing unit (GPU)s, and trusted execution environment (TEE). It is vital to support standard libraries, packages (for researchers to deploy their applications on the device), and control interfaces (USB, I2C, SPI, serial) to add new hardware modules with standard network technologies (Wi-Fi, wired, Bluetooth) [8, 19]. Creating an interoperability matrix that captures the different versions of software and the OS is important. For example, Debian 11 switched to cgroup v2, which broke some applications (docker monitor) [112].

The platform can contain hundreds of thousands of endpoint and edge devices. It is essential to have a good naming convention for devices at each tier to identify them uniquely and the data generated from the devices [23, 69]. Also, all devices in each level (cloud, edge and endpoint) must be synchronised in time for data integrity and audit log purposes [68, 71].

Requirement analysis helps to understand the research project's aims and objectives. System design helps to understand how the set of requirements can be achieved. Once a higher-level system design is defined, the testbed architect can start implementing the testbed architecture, functional model [113], and how devices at the endpoint, edge, and cloud tier will be managed, provisioned, and communicate with each other [21].

C. Implementation

The implementation phases bring challenges such as provisioning devices, ensuring secure network connectivity, credential management, application deployment, and compatibility between different hardware architectures (armhf, arm64, amd64), hardware and software accounting and monitoring. The challenges of the integration phase include ensuring that the platform is scalable, modular, extensible, adaptive, and reproducible and supports heterogeneous devices, proprietary software, and different standards.

Provisioning the cloud, edge and endpoint devices: Provisioning the cloud tier requires the installation and configuration of VMs on the on-premises hosted hypervisor (Hyper-V, Proxmox, OpenStack) or cloud hosting providers (AWS, Azure). The number of VMs depends on the services required to support the edge and endpoint tier and usually ranges from one to ten. Installing and configuring a VM is a tedious task and requires installing OS applications, configuring them securely, and configuring hardware allocation (e.g. RAM, CPUs, GPU passthrough). Most research projects currently provision the servers manually or using a bash script. The bash script installs the necessary packages and configures them with security. Those images can be packaged to support different hypervisor environments without requiring changes to the provisioning scripts and source code. Such platform-independent virtual machine image creation tools are Yocto and Packer.

Provisioning edge tier devices (Intel NUC or SBC) involves installing an OS on the SD card/Hard disk drive (HDD)/eMMC, with configured software packages, and ensuring stable and secure connectivity to the cloud tier. The number of edge devices depends on the sample size of the case study, such as the number of houses or streetlamps, and can range from one to hundreds. One way to provision edge devices is to create a base kernel image containing the installed OS and applications and flash it to the edge devices. Adding the Linux kernel headers in the base image is essential because future application installations might require building a kernel module (e.g. wireguard). Otherwise, the base image needs to be created and flashed again. For any further changes, the administrator logs in to the device using the SSH/serial console and configures it according to the requirements. Creating a base image and flashing it on multiple edge devices comes with security and administration challenges. The security challenge is that the credentials and other settings, such as Wi-Fi SSID, hostname on all the edge devices, will be the same until changed. If one of the edge devices is compromised and the attacker obtains the credentials, they can compromise all the edge devices by performing the lateral movement. The administration challenge is to log into the machine and make changes after flashing the base image. For example, deploying the edge device in the citizen's home could require changing parameters such as house number identification, Wi-Fi credentials, and IP address settings. Additionally, suppose

that the device is deployed on citizens' premises during pandemic outbreaks. In that case, minimising the time spent configuring the device is essential.

Endpoint tier devices are usually resource-constrained devices, such as SCK [114], Luftdaten [115], SensorTag [116], and Smart Plugs [117]. Endpoints are usually connected to the smart home platform or the edge device. The provisioning of endpoint devices depends on the capabilities of the device and the communication medium between the endpoint, edge, and cloud. It mainly includes configurations such as setting up the connectivity (using Wi-Fi/ZigBee/802.15.4), the MQTT server address to publish sensor data, and the time at the endpoint using Network Time Protocol (NTP). Moreover, standards such as Lightweight Machine to Machine (LWM2M) [118] have been developed to manage endpoints securely and in a mannered function. LWM2M provides device management capabilities (remote provisioning of security credentials, firmware updates, and connectivity management) and service establishment capabilities (sensor readings, remote actuation, and endpoint device configuration). Various papers [26, 33, 67, 68] have provided lessons learnt from experience by deploying battery-powered devices in the endpoint tier communicating over IEEE 802.15.4.

Endpoints could also be configured dynamically or bootstrapped by the device on the edge/cloud tier by providing configurations such as which endpoints are allowed to join the network, the encryption keys to encrypt the data, and the network address/port number of destination, and other settings. Additionally, communication between the endpoint and the edge must be encrypted. For example, if the endpoint connects to the edge via 802.15.4, the edge device requires a border router to communicate. If the endpoint connects to the edge via Wi-Fi, Wi-Fi encryption (WPA2) encrypts the data over the air. For example, the SPHERE [71] project deployed multiple endpoints connected using 802.15.4 in around 100 houses in Bristol and used one hard-coded encryption key per house to encrypt data over the air. They used media access control (MAC) address filtering to prevent external devices from joining the IEEE 802.15.4 Time Slotted Channel Hopping (TSCH) network.

***Endpoint-Edge-Cloud Connectivity**:* From the communication perspective between devices at each tier, it is essential to use encrypted protocols for communication from endpoint to edge to cloud tier [67, 71]. Secure transmission protects against packet sniffing, man-in-the-middle attacks, replay attacks [119], and unauthorised attempts to communicate with the node.

The servers that host the cloud tier must provide services to edge tier devices and expose them to IP addresses and ports. Services could range from Hypertext Transfer Protocol (HTTP), HTTPS, WebSockets, Lightweight Directory Access Protocol (LDAP), VPN, and others and may require different ports exposed to the Internet. Testbed administrators prefer to reduce the number of ports exposed to the Internet to reduce the attack surface area, which is better from a security perspective. An example of a WSN implementation providing the connectivity points between the three tiers is presented in Fig. 5. Both sensor LPWAN nodes and cloud addressable

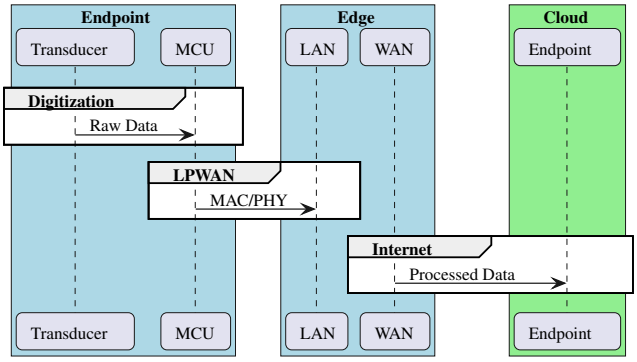

Figure 5: Connectivity points between the three tiers for a WSN use case

Uniform Resource Locator (URL) or IP can be considered endpoints. The challenge for the edge device is to distinguish between the two directions of communication. Routing tables for packet forwarding for LAN and WAN and also the SLIP bridge create complexity and are challenging to design, implement, and secure.

Edge to Cloud Connectivity: There are three ways to expose services hosted on the cloud tier. Firstly, by opening the ports on the cloud tier firewall. However, opening multiple ports on the firewall increases the attacker's surface area and is not preferred [120]. Second, connect the device through Demilitarized Zone (DMZ) to the cloud using a VPN [28]. However, in the case of a cyber-incident where an attacker compromises one edge tier device, they can explore and enumerate the internal network for vulnerabilities (depending on routing configuration and if the network is flat at the data-link layer). The third is to use a Software Defined Perimeter (SDP) that runs a client on the device using the authentication process. SDP defines a policy to determine who gets access to what resources and distributes access to internal applications based on a user's identity. It makes the application infrastructure invisible to the Internet, evades network-based attacks (DDoS [119], ransomware, malware, server scanning) and reduces the security risk. However, enterprise organisations often use SDP, which might be overkill for a research testbed. Furthermore, if the devices at the edge and cloud tier are in the same network connected over ethernet or Wi-Fi for demonstration purposes, edge and cloud tier devices will be in a trusted private network; VPN or firewall might not be required.

The typical way to connect edge devices to the cloud network is through a VPN. For example, if there are 50 edge devices in different houses or streetlamps, it is good to generate 50 unique credentials from a security perspective. However, more manual/scripted effort is required to create credentials and provision them to nodes. For example, the REPLICATE project used OpenVPN to provide secure connectivity and issued certificates through a CA. The administrator generated 150 credentials and stored them on a USB stick with 150 folders for each house. The deployment team (DT) was responsible for visiting a particular home and installing and provisioning the edge and endpoint devices. They executed the

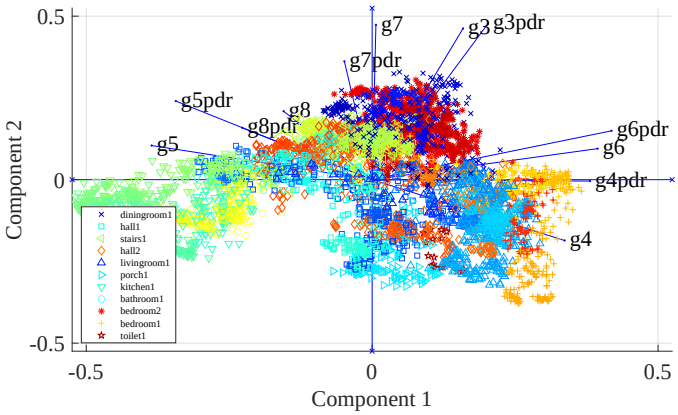

Figure 6: Mobility of BLE tags in a house, the association of the PDR and signal strength for eight listening gateways

bash script on the edge tier device that installs the certificate for that house and provides secure connectivity to the cloud tier.

Endpoint to Edge Connectivity: Endpoints are usually connected to the edge/cloud using mesh networks and LPWAN technologies. The choice of network technology depends on connectivity requirements such as range, bandwidth, power, interoperability, security, and reliability [85].

However, there are challenges when multiple endpoint devices communicate over various channels in an urban environment. An urban environment can have numerous networks such as cellular, LPWAN, mesh, and others. In a real-world deployment, connectivity between multiple devices in the vicinity of each other depends on external interference, frequency-selective multipath fading, and dynamics in the environment. The dynamics of the environment can include the number of people, the movement of people, the Wi-Fi traffic, the rooms, the layout, and the type of building materials used [36, 87]. A house deployment might initially function until further technology is deployed into a neighbouring house, causing disruptions due to radio interference. External interference can occur when a different technology or a deployment of the same technology operates within the same radio range (IEEE 802.11 Wi-Fi interferes with IEEE 802.15.4 at 2.4 GHz) [25, 121, 122]. Furthermore, in an 802.15.4 network, the mobility and activity of an endpoint can affect the throughput and data on the mesh infrastructure.

Fig. 6 presents the packet delivery ratio (PDR) calculated from packet sequence reconstruction for individual receivers in a home environment. The strength of the received signal and the packet loss patterns show the effect of mobility between rooms in the residential environment and the effect on PDR. The PDR is affected by the increased bandwidth requirements on the forwarding gateways when many packets are generated locally that require forwarding. In Fig. 6, four tags that require a fixed uplink bandwidth generated enough packets to saturate the uplink capacity allocated to the mesh network. In particular, gateway 8 is sharing uplink bandwidth with gateway 5, which is visible from the alignment of the two principal component analysis (PCA) components of PDR (g8pdr and g5pdr). In other words, gateway 8 uses gateway 5 in a mesh network topology to forward its traffic in the network. Since the available bandwidth is limited, there is a lot of packet loss in the data originating from gateway 8, making the PCA component **g8** the least significant in the overall entropy. The PDR, network usage, and packet loss have a dynamic nature in a dynamic environment [26]. For example, SPHERE has deployed a mix of network technologies such as 802.15.4 400 MHz, BLE channels 37, 38, 39, and 16 channels of 802.15.4, 5GHz Wi-Fi, and a router with an Ethernet interface. BLE packets were generated on the advertisement channels 37, 38, and 39 with an interval defined by the BLE 4.2 standard at about every 200 ms. The specification allows only a fixed interval with increments of 0.625 ms with a random delay of 0 ms to 10 ms. These packets are scanned from receivers that scan on one of the three channels at any particular time and rotate across those channels many times every second. Those packets are encapsulated in CoAP messages, which are forwarded to the 802.15.4e gateway from these intermediate receivers using a fixed uplink time-slotted schedule. The gateway uses a bridge to bring CoAP messages to a compute host using Contiki-NG [123]. Link quality is an important metric when connecting endpoint devices to the edge/cloud. When the security of the communication channel depends on the Radio-frequency (RF) channel, if an attacker gets physical access to the device or sniffs the network, they can learn the procedure for joining the network, such as the exchange of network keys. In particular, in IEEE 802.15.4, in the minimal implementation, the pattern of connecting a node to a network uses a fixed channel [124]. Information for the particular network in its formation [104] can be inferred by sniffing those 10 ms timeslots where routing is established [125].

Credential Management: After provisioning, the edge and cloud devices must be maintained and accessed occasionally. One of the ways to access the device is by SSH using authentication mechanisms or credentials such as a username, password, or digital certificates [28, 126]. The device can authenticate the user by storing the credential on the device or authenticating through a central server and storing it locally for a specific time. Using passwords is not recommended, as it allows the attacker to brute-force the username and password. Furthermore, when the password is sent to the device for authentication, it can be compromised by man-in-the-middle (MITM) attacks [120]. One preferred way of providing access is to store the administrator's public SSH keys[1] [127] in each of the devices. However, storing public SSH keys on the device is risky as if one of the private SSH keys is compromised, access to all edge devices may be compromised. In addition to using SSH, administrators also use remote management tools such as TeamViewer/AnyDesk to update scripts or perform functionality that requires Graphical User Interface (GUI). However, recently attackers compromised Florida City's water supply using remote access software (TeamViewer), which allowed staff to share screens and troubleshoot IT issues [128] by exploring systems from the Shodan search engine and outdated passwords.

[1]SSH has public and private keys, the public key is stored on the device, and the private key is kept with the user requiring device access.

Application deployment and compatibility on different systems: Research projects involve multiple researchers developing different applications (Python/R programs) [103] that need to be deployed on the edge device with different architectures (arm64, amd64, armhf). Researchers need to access edge device hardware (sensors, cameras, GPU) for edge processing and cloud resources for data analysis. Initially, developers work on sample data and develop applications that work fine on their machines. However, applications must be deployed on the edge and in the cloud to access real-world data. Deploying custom applications often requires installing library dependencies (e.g. pandas, scikit-learn) and may require administrative privileges, often resulting in the application not working correctly on the edge/cloud platforms.

The above results in scenarios where developers say, "It works on my machine!" resulting in numerous meetings and debugging of applications to determine the root cause of the problem. Python and Linux distributions have a lot of inter-component dependencies embedded into them. It is crucial to monitor those interdependencies and evaluate any security updates against those dependencies. Tools are being explored in the literature to evaluate those dependencies [129, 130] and provide early warning when changes lead to incompatibilities.

Additionally, the project must always store the data collected on designated machines to comply with data protection laws and user privacy. Many applications need access to a graphics card or more memory to process the data. This requires moving the data to a more computationally capable machine, which becomes challenging due to data management guidelines. Due to data management guidelines, application incompatibility often results in either no or delayed application execution on the whole dataset. The application code also needs to be consistently deployed on devices; one of the ways it is maintained is by using a remote git repository cloned on the device remotely updated as a batch process [35].

Accounting and Monitoring: The testbed can contain tens, hundreds, or thousands of devices on the cloud, edge and endpoint tiers. It is crucial to maintain an inventory of the number of devices at each tier, with their hardware and software details (make and model, OS versions, installed applications, and their version) [72]. The OS and application version can be used to actively monitor the National Vulnerability Database (NVD) database to detect vulnerabilities and patch the system proactively. Additionally, audit logs with synced timestamps should be collected to a central server and enabled to ensure forensic investigation during cyber-security incidents. Also, it is essential to maintain the details of who (i.e., which user) has logged into which machine and performed what activities for auditing purposes. However, it can depend on the remote management software's licence (free version/enterprise edition).

The infrastructure deployed for data collection requires that all hardware/software be working as expected and usable by researchers [127]. In addition, all endpoints must be connected to the edge, which should be connected to the cloud tier. If not, any loss of network connectivity can result in data loss. The monitoring infrastructure is essential to ensure this [28, 31, 34, 71, 87, 131]. Monitoring includes detecting whether devices are reachable and sending regular data. Monitoring also includes checking infrastructure components (such as web servers, adequate disk space, and system overload) [132]. The monitoring infrastructure should include an effective alert mechanism (email, slack, text messages). From the endpoints deployed through 802.15.4, it is good to have statistics about energy (battery), network (number of data/control packets, acknowledged packets), neighbourhood statistics (list of neighbour nodes and the link quality), per-channel per-neighbour packet reception rates, TSCH time synchronisation performance, background noise Received Signal Strength Indicator (RSSI) levels, stack usage, and others [68, 71]. For example, SPHERE [87] monitored the status (reachability) of the deployed endpoints by regularly polling various devices within the home network based on Nagios.

D. Integration Testing

After the system design and implementation of the testbed, it is vital to perform integration and testing at regular intervals, such as ensuring that interlinked components are working correctly; the platform is scalable, modular, and extensible; integration of heterogeneous devices, proprietary software, and different standards; ensuring endpoint and edge provide good ruggedisation; ensuring testbed adaptiveness and replicability.

Interlinked components dependency: Data gathering research projects have multiple interdependent components and interfaces installed on devices to ensure data transfer from the endpoint to the cloud. A component is the system's part/block (hardware/software). On the contrary, an interface is a part that connects two or more other components to pass information from one to another [133]. It is the mechanism through which the components of the block communicate. For example, a web server is a component, and the HTTP/WebSockets (method of communication) will be the interface. The glueing of software components requires considerable effort and in-depth knowledge of the components [26]. The data generated by the endpoint follow a pipeline and travel through multiple interconnected components to the cloud. Each component expects the data to be in a specific format or size. Often, a component might fail to pass the data to the next component in the desired form, failing the whole pipeline [19]. For example, an endpoint sends the data (such as temperature readings) through MQTT in JavaScript Object Notation (JSON) format to the edge device for processing and storage in an InfluxDB database. The edge device can run a Python script to check if the temperature is above a threshold and notify the cloud tier. There could be multiple points of failure in this example, such as issues in MQTT, wrong JSON format, InfluxDB server not running, python script error, and others.

An administrator often needs to buy several devices with different components and interfaces for a research project. They need to learn how the devices work, test them, ensure that the data can be fetched in a limited amount of time in a lab environment in a specific setting, and finally deploy them in the wild [23, 30]. For example, research projects that involve energy monitoring deploy multiple devices such as smart plugs [134], Tesla powerwall [135], OpenEnergyMonitoring [136]. When deployed in the real world, there

is a probability that a system component will not work as expected due to hardware or software failure [24]. Debugging and finding the misbehaving piece takes considerable time and is challenging [9, 23, 77, 137]. It requires detailed logs of different system components with timestamps, understanding what triggered the logs, and ensuring that the devices generate log messages representing various failures.

Therefore, performing regular automated integration and end-to-end testing is essential to prevent such failures [73]. Additionally, components and their functionality must be well defined and have robustness and resilience built in, saving time for system administrators [71, 85, 87]. It also helps minimise the number and duration of visits to the citizen's residence to repair the system [35]. The maintainability of the infrastructure and the consistency of the interfaces between all different components [40] (such as commercial off-the-shelf (COTS) of hardware/software) can help with the resilience of the infrastructure.

Scalability, modularity and extensibility: Research projects require the deployment of endpoints in multiple locations. The testbed platform is easy to manage when small and consists of only a house/streetlamp in one place. However, running a scalable trial that is supposed to scale to 100-200 houses/location becomes challenging. The system must be able to scale to tens to thousands and tens of thousands of homes/streetlamps in a reliable manner [24, 30, 43, 87]. In addition, software and hardware development occurs rapidly and can quickly become obsolete. The hardware and software components of the test bed must be designed with modularity and extensibility in mind to adapt to ever-evolving technology [24]. Hardware and software at the cloud/edge tier can be modular and extensible (for example, replacing the SBC at the edge with a newer, more powerful SBC) [21]. However, modularity, extensibility, and future-proofing at the endpoint tier is challenging because it is difficult to predict the exact requirements of future deployments and the electronics market progresses quickly. As a rule of thumb, testbed designers should follow the Keep it simple, stupid (KISS) principle [85].

Heterogeneous devices, proprietary software, and different standards: Projects can have different devices on edge and endpoints generating various types of data and formats [30, 34, 75, 106, 113, 138]. For example, edge tier devices can have SBCs (GrapeBoard, RPi, Coral boards, Intel NUC). Endpoints tier devices can have different devices such as Nordic Semiconductor nRF5340-DK2, Texas Instruments Launchpad (LAUNCHXL-CC2650/CC1310/CC1350), TI CC2650 SensorTag. The testbed requires the devices to be securely configured and connected to the network. In addition, the endpoints used to collect data can run open-source or proprietary software [30, 34, 113]. In the case of proprietary, they may not provide an open source script to take the sensor data and may have a GUI to download the data or allow it to be sent only to the endpoint manufacturer website. In such cases, the administrator must figure out how to extract the data from the proprietary device or the manufacturer's website. Some proprietary technology may not be designed or evaluated for cybersecurity purposes. In addition, it is always difficult to evaluate and secure different network connectivity (802.15.4, BLE) in IP networks.

As there may be different devices from different vendors on the testbed, they can be running on various standards and formats (sending data over MQTT, HTTP, WebSocket, proprietary protocol), resulting in a lack of interoperability between sensors [32, 77, 78, 79, 113, 139]. It is vital to use widely open standards and possibly the same standard and format to help reduce learning times for research personnel [21, 71].

Ruggedization: Ruggedisation is essential when deploying devices in citizen houses or outside on streetlamps. For example, any edge device installed indoors/outdoors requires specific Ingress Protection Ratings (IPR) and electrical testing [68]. It must be packaged in a form that can be securely mounted [8, 31] and still easily open if a battery or component change is required. IPR define levels of sealing effectiveness of the electrical enclosure sealing against foreign body intrusion (i.e., dust) and moisture. From the electrical safety perspective, it is crucial to have a Conformite Europeenne (CE) rating (for EU/UK) or country-specific certification rating on the endpoint and edge device. The certification mark ensures that the manufacturer has verified that the products have met country-specific safety, health, or environmental requirements. For example, Bristol Urban Observatory (BUO) had difficulty installing AoT nodes in streetlamps and on the university campus because the nodes did not have CE ratings (the electrical safety certification of the USA is different from the UK). Additionally, when designing enclosures for devices that contain sensors (such as air quality), it is essential that the airflow is optimal and allows the proper functioning of the sensors on board. The enclosure should protect the electronics from moisture and insects [8]. It might be a good idea to place the sensors in a Stevenson radiation shield[2] separate from the sealed waterproof electronic enclosure. Furthermore, it is recommended to identify a suitable enclosure first (accepted and visually aesthetics) and then fit the edge and endpoint device in it with minimal modification. Designing a custom casing is often challenging and more expensive than modifying a readily available casing [67]. During the Cotham Hill Pedestrianisation project, it was found that designing a 3D-printed enclosure, models, printing it, and post-processing the 3D print (cleaning up the support materials) is challenging and time-consuming.

Testbed adaptiveness and replicability: The testbed must be adaptive to the project requirements or the community demand. For example, change in hardware requirements (such as a powerful graphics card, more RAM, hard disk space, or low-power processors) or human-interaction interfaces (ways to visualise/process data). Also, supporting as many users as possible depends on two factors: cost of users, experiments, and adapting the testbed to the needs of different communities [74, 140]. Also, the testbed should be reproducible using open-source software and automation, allowing implementation of the testbed by other administrators using applicable documentation (e.g. wikis) and other supporting materials.

[2]shield instruments against precipitation and direct heat radiation from outside sources while still allowing air to circulate freely around them

E. Operational Testing

The next step is to develop a prototype testbed in a laboratory and a small-scale real-world environment before large-scale deployment in the wild [20, 21].

Time resource allocation: The concept of time as a resource available to the testbed can be interpreted as a CPU processing time at both the edge and the endpoint. Furthermore, this can be associated with radio utilisation time at the co-coordinating endpoint connected to the edge or other edge nodes. The available time is governed by the data rate related to the sensor sampling frequency and resolution. Monitoring tools enable observations such as CPU time use and radio usage, which is essential when scaling the testbed. To give some real-world perspective, a byte of data, when transmitted, is serialised into eight bits of 0's and 1's and sent over a medium such as wires or radio. Communication protocols are responsible for encoding/decoding the bytes and bit streams and depend on the medium's capacity in bits per second. This can create an interesting paradigm between radio use and environmental monitoring. Almost all analogue-to-digital converters support Layer 2 access control allowing many sensors to be connected to inexpensive System on Chip (SoC) micro-controllers. This reduces the cost of the Printed Circuit Board (PCB) design by reducing the number of wire traces and complexity. Similarly, the radios, where the MAC layer controls access to the radio medium. In both cases, consideration of time allocation applies.

Lab deployment: The testbed will contain multiple heterogeneous devices at each tier. Each device would have different interfaces, components, applications, and services running. It is essential to ensure that the system is working as a whole [141] and securely sending the data from the endpoint to the cloud with analysis and visualisation satisfying project requirements. The platform must be deployed in a laboratory environment before being deployed on a large scale. It helps to face the challenges early on and test any new software/application internally on the testbed rather than pushing it directly into production.

Assignment of a provisioning budget is essential for setting up a lab testbed, buying various spare devices and components, and conducting deployment site visits. Based on the budget, project scope, and the number of researchers working, it might be good to have more than one lab testbed (dev1, dev2). Multiple lab testbeds help keep work in progress, even if one testbed has broken down because of a misconfiguration or software/hardware failure. Additionally, the laboratory testbed must be set up and running as early as possible in the project to test the different devices, components, software updates and applications to ensure the final real-world deployment is completed on time. Although only sometimes possible, the testbed should be as close as possible to real environmental conditions. For example, the Living Lab project first deployed electrochemical air quality sensors using laboratory-based wall sockets; however, electromagnetic interference from the power supply caused interference in the sensors, affecting the readings when deployed in the field [20].

Small scale real-world deployment: Research projects often require the installation of sensors in the environment/infrastructure owned by a different party. However, before deploying a large-scale deployment, it is important to have a small-scale deployment to understand real-world challenges and build confidence with infrastructure owners. Devices may behave differently depending on external factors (power supply, network infrastructure, and physical environment) [9, 20]. The small-scale deployment could include one citizen house, streetlamp or vehicle. Deploying scientific infrastructure on others infrastructure (bridge - owned by a trust, streetlamps - owned by the council, citizens' house - rented or owned by tenants) requires partnership with the respective owner [8]. There could be two individual bodies governing the infrastructure, first, the management team (MT) (board of directors, members of C-suite) and second, the operations teams (OT) (people managing/implementing the infrastructure). We refer to the research team (the team that deploys the infrastructure) as DT for brevity. Fig. 7 provides the different teams and their relationships.

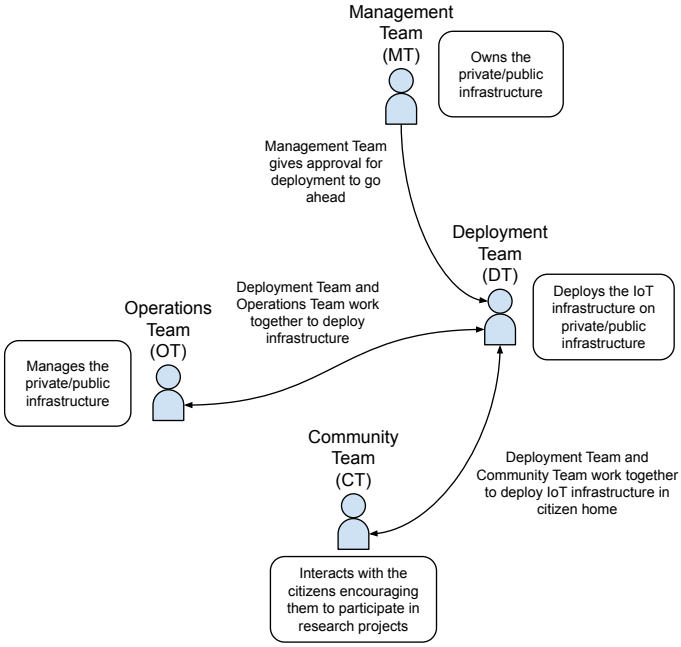

Figure 7: Different teams involved in smart city research projects and their relationships

During multiple projects involving device deployments, it was found that it is essential to gain the MT's trust (such as citizens and the city council) and inform them about the benefits of deploying the monitoring infrastructure. They will require assurance that the DT takes their work seriously and that installing the monitoring infrastructure will not disrupt their infrastructure working in any way.

Once the MT is on board, the DT must work with the OT. OT could be performing essential jobs such as keeping the city, a bridge running or operating their electric bicycle platform. The OT of different companies has their own key performance indicators (KPIs), processes, and structures. The challenge for DT personnel is to fit into that culture without causing problems. The DT should provide details (make, models, working, safety, security) of the monitoring infrastructure to

gain OT's confidence and trust. The DT should experiment with the OT infrastructure without disrupting them and not being a burden. They need to explain and provide realistic expectations about the research project and what and how they will be doing it. Furthermore, the relationship between DT and OT should be sufficiently positive so that the research team can fit the practise of the infrastructure operations team and that OT is happy to work with DT.

Finally, the DT should behave safely, securely, and carefully while working with the OT. The DT must be aware of health and safety concerns [20] and respect other people's time. For example, installing sensors on other infrastructures is often cancelled for non-technical reasons (e.g. violating health and safety requirements). Installing the sensors on an initial site (first house, streetlamp) will build up the DT's confidence and relationship with the OT/MT team.

F. Implementation/Deployment (in the real world)

Data-gathering research infrastructure can be deployed at citizens' houses, private buildings (offices), and public places (streetlamps, council vehicles). All have a different set of challenges. First, we cover the challenges faced in the deployment in citizen homes and public spaces. In addition to the deployment team (DT), we denote the community team interacting with citizens as CT. CT is often responsible for interacting with citizens and informing them about the project research objectives and results. They are the bridge between citizens and the DT.

From the perspective of citizen participation, privacy and transparency, it is also a good idea to display the data the device collects and how it is used by providing documentation near the device [20, 142]. It is also important to mention to whom the device belongs and where to contact for more information [68].

***Deployment in citizen houses**:* Challenges faced by the CT can be divided into **i.** finding a way to interact with citizens **ii.** encouraging and involving them to participate in the research project **iii.** providing adequate information to citizens **iv.** maintaining regular contact with citizens.

Finding potential motivated citizens: Recruitment and engagement of citizens (potentially motivated) is challenging, requires proper planning and often requires plenty of time. It is more manageable in areas with community cohesion or a coordinating body to promote the project [27]. Recruitment works best using various methods, from brochures and social media to door-knocking and face-to-face visits [143]. While interacting with citizens during the REPLICATE, Twinergy project, it was found that it is essential to consider literacy rates within the pilot area and to publish information/leaflets in the local language [143] for non-native English citizens. Also, over the years, the CT often knows citizens from previous engagements who would be happy to participate. Local events are a good way to attract interest. The CT organises small events or has a booth with information during open markets. Before engagement, it is essential to check whether there is a specific research project requirement, such as the deployment of devices in citizens' houses with diabetes or Parkinson's disease or citizens with solar PV or in an excellent socio-economical situation [6]. In such cases, CT interacts with different community groups through local community centres and social media applications, such as Facebook and Nextdoor [144]. Additionally, pandemic events such as COVID-19 make it difficult for CT to interact with citizens.

After identifying the recruitment method to build citizen interest, it is essential to consider the larger picture and connect people to these concepts. The CT also uses creativity and art to get that message across. The involvement of the physical and kinaesthetic aspects of the citizen often helps people become more involved, engaged, and excited about the research project. For example, Knowle West Media Centre (KWMC) CT installed a booth with a workshop of crafts activities to engage citizens during an open market. Once citizens are engaged and enjoying the craft activities, the CT asks for details about where they live and introduces the research project objectives. Additionally, citizens often drop out of the research study for multiple reasons, such as ill health, changes in circumstances, moving house, and occasional frustration with technology/process [27]. Therefore, having more participants than the project requires and having few citizens as a reserve is always good.

Citizen encouragement: The second challenge of CT is to get citizens excited about the project. It often comes to a fundamentally simple proposition: why they (citizens) would get involved and what is in it for them. Citizen participation becomes more complicated if the project requires a power supply or Wi-Fi (which costs money to citizens). When expenses are covered, there will still be a disruption in citizen life due to the installation of devices in houses [35]. In many cases, incentives (free Wi-Fi access, free tablets, shopping vouchers, or the opportunity to win a smartphone) will not convince citizens to participate. It is essential to think carefully about how citizens can be recruited and maintain interest among them [143]. For many people, simply getting involved is a barrier. For example, Twinergy [6] requires that citizens have solar PV connected to their homes. However, citizens who have solar PV will be early adopters and tech-savvy, so they may not be interested in the project. Citizen onboarding to the research project is challenging and can involve different efforts depending on citizens' eagerness and benefits.

Respecting citizen time and preferences: Deploying the endpoints in a home involves connecting up the sensors (using Wi-Fi, LPWAN or mesh networks). It can take a reasonable time, depending on the number of endpoints configured or connected and finding and deciding on a suitable place to keep the device, talk to the participants, and answer their questions [35]. Technology that is easy to install with little or no cabling is preferred. Radio transmission devices are preferred as citizens do not prefer additional cables in their staircases and dwellings [143]. During the Twinergy project, one participant decided not to install the

technology because it would spoil their minimalist decor.

In case of Wi-Fi connectivity, DT would need the credentials (SSID and password) and can collect them through phone calls, online forms, or in-person. However, remotely managing the Wi-Fi credentials often results in issues such as participants being uncomfortable entering their password into a document, participants needing to know their Wi-Fi credentials, and mistakes made during communication (such as mistaking O with 0 (zero)). An incorrect Wi-Fi credential is only detected when the deployment occurs. In this case, the endpoints must be returned to the DT and loaded with the correct network name and password, or a visit to the participant's house is required to correct the credentials [35].

Furthermore, the endpoint devices must remain placed throughout the deployment period without damaging the participant's house (delicate surfaces such as precious antique wood and wallpaper) [34, 72]. It is advised to anticipate objects and environmental conditions that can affect installation. This includes moisture, the quality of surface finishes, the typical movement of the object, and the methods of interaction of inhabitants with the object [31, 72]. Often, the citizen, pet, or robot vacuum cleaner accidentally or unknowingly disconnects the power supply to the devices, causing a failure, resulting in loss of connectivity and data [34]. Therefore, it is essential to identify the location of the device deployment at home carefully. The DT must respect the citizen's house and time [26]. The longer the DT takes at a citizen's home, the more inconvenient it is for the citizen and their regular routine [35]. Home visits of citizens for deployment and maintenance purposes must be highly optimised and efficient with preparation done beforehand [34].

Expecting user participation at all times is futile; expecting users to accurately record their activities for labelling data (such as who cooked dinner at what time) is challenging, as it requires citizens to remember and observe their lives [34].

Device looks and deployment surrounding: User comfort, acceptance, and aesthetics of deployed devices are paramount for a successful deployment (especially for wearable endpoints or visible devices) [67, 85, 87]. The citizen usually prefers the devices to look aesthetically or hidden away. When there are deployments in the citizen home, there must be no light emitting from devices deployed in bedrooms, as they can disturb users' sleep or affect user behavior [67, 72]. Furthermore, LEDs also consume a good amount of energy [68]. It would be good to have the ability in the endpoints to turn on/off the LEDs so that they can be on during debugging and off during real deployments [19]. For example, SCK deployed on the Cotham Hill citizen's house emits red light in case of setup issues; a senior resident was concerned and asked if it is safe to operate and has no fire hazard. In addition, it is essential to ensure that the device does not make any noise that can affect the lives of citizens [34].

It is also essential to note the device deployment conditions or the surrounding location to understand the sensor readings [72]. For example, a temperature reading in an area with direct sunlight will vary from a temperature reading in the shade [8]. To provide another example, anomalies in the SCK noise sensor readings in the Cotham-Hill deployment were observed because of the direct sunlight on the SCK kit kept near the window. Direct sunlight leads to device heating and can affect sensor readings [22]. In public deployments, context is also essential (near an intersection, highway, garbage can, and recycling centres). It is critical to understand how local environmental conditions (indoor/outdoor/sunshine/rain/snow) will affect the deployments [68].

Citizens switching home broadband provider: The device installed in the house often connects to the Internet through the ethernet port of the broadband router or Wi-Fi (which requires broadband Wi-Fi credentials) [37, 120]. For example, in REPLICATE, the endpoint connects to the edge device using ethernet to forward and route all traffic from VPN to the smart city platform. Most edge devices are SBCs with one ethernet port and a Wi-Fi adapter. Therefore, when the Ethernet port is occupied, the device must connect via Wi-Fi to connect to the Internet.

Citizens often change their broadband providers from one to six months to a year, leading to the change of Wi-Fi credentials (SSID and password) and loss of Internet connectivity and data. The DT does not have any mechanism to replace the Wi-Fi passphrase but requests the household owners to change the Wi-Fi passwords to what it was before, including the SSID, so that the device can connect to the Wi-Fi network. The other way is to plug the edge device into a monitor, attach a keyboard/mouse, provide credentials to the household owner and ask them to run the script to change the Wi-Fi password. However, most homeowners are not tech-savvy, making changes difficult. In addition, many citizens are unfamiliar with the technology introduced to their homes. For example, citizens might not have the experience of using a tablet or have problems accessing their information via the Internet [27].

Deployment in private building and public spaces: The deployment of any devices on the city's infrastructure (buses, garbage trucks, streetlamps) requires the willingness and collaboration of the city council [31]. Similarly, deploying devices on private buildings requires the building management team's approval. During the Clifton Suspension Bridge project, it was found that it is essential to ensure that any device deployed does not hinder the functioning of city infrastructure or private buildings. The power source for the deployed device must be planned (such as streetlamp power or car batteries when deployed on buses/trucks, mains powered, battery powered) [31]. It takes time and effort to secure permissions with the relevant infrastructure owners to deploy devices. Therefore, it is essential to identify the locations with the most significant impact to deploy the edge/endpoint that provides the most value to the stakeholders of the research/project [23, 35]. Suppose the device is deployed on the streetlamps and contains a downward camera. In that case, it might be a good idea to mount the device at a higher position to protect it from vandalism or theft [23, 30]. This would also allow an extensive view from the camera, allowing images of the entire intersection/park.

For a successful public deployment of infrastructure, policies, agreements, processes, public engagement, and interac-

tions are necessary.

Public engagement: Public engagement is essential for the success of the research project. It brings city residents closer to the project and makes them active participants. It helps citizens without technology experience to discuss and learn the use of data and technology. This broader citizenry can explore and develop solutions to urban issues by proposing ideas for how collected data can be used. Community centres or community outreach help to publicise the project. There must be a named person to whom participants can go with any questions [143]. Face-to-face meetings help people identify and assign a named person to a project. Throughout the project, excellent and responsive personal support from a friendly and accessible coordinator (in the form of a building manager, a housing association contact, or even a community leader) can increase engagement. Any research project aiming to impact citizens' lives or affect behaviour change must build a relationship with participants and a deep understanding of their contexts and motivations to increase engagement and participation levels. Users must feel involved in each stage of project development and see that their participation is valued and that their input can have a real impact [143]. In addition, periodic reinforcement of the message and encouragement by contact between the neighbours and the central coordinator helps keep the motivation and the participants interested [27]. It is vital to provide ongoing support through visits, calls, and workshops, especially for those who find technology difficult or have literacy problems. Creating a relationship with participants based on trust and responsibility for communicating bad and good news [143] helps the researcher and the citizen.

Also, there is a possibility that the research projects engage with people from underserved or disadvantaged socio-economic or minority ethnic backgrounds. It is crucial not to lump them into one group. The CT must treat everyone equally and ensure that communication with the citizens is appropriate and accessible, and no one should be offended.

Furthermore, the amount of information must be provided in an easily digestible fashion (short video, infographics, a mechanism with which citizens can engage and interact) to get comfortable with the idea and not overwhelm them. The research project results depend heavily on the interaction and feedback of the participants. Hence, it is essential to ensure that easy-to-understand and straightforward messages are used to communicate with citizens (communication is key) [27]. For example, SPHERE created a 3 min animated video [145] to provide information to the citizens. Being active on social media, such as Twitter, responding to media requests for interviews, and publishing detailed information about the research project on the website/pamphlets/leaflets helps improve public perception and participation [90].

In the case of deployment in citizen houses, once citizens are on board and have signed the consent forms (ensures commitment and guarantees confidentiality), and the DT has installed the devices in their house, it is still essential to maintain regular contact with the citizens to ensure devices are working and they can use the technology and data provided for their benefit. Another minor challenge for CT is managing the signed consent forms provided to citizens for participation. Encouraging all participants to return completed questionnaires is always challenging and must be considered for any citizen attitude/behavioural analysis [27].

Transparency: Deploying any public infrastructure requires transparency, privacy protection, and system security. The public usually suspects publicly deployed devices based on fears about surveillance and data collected by the node [20]. It is essential to develop and provide privacy and governance policies to show the project's commitment to transparency and privacy. The privacy policy should provide what data are being collected, processed, used, destroyed, or made available to city residents. Additionally, allowing open comments from the citizens and community on the policy drafts help gain citizen confidence. DT/CT can arrange community meetings for citizens to ask questions about the draft policies. It is essential to resolve all the comments and questions publicly, consider citizen feedback for policy revision, and include a report of the public engagement process. The public/government cybersecurity centre can assess the deployed system security and privacy practices to ensure system security and gain public trust [90].

In the case of deployment at home, citizens will have questions about the different endpoints, frequencies used, data collected, and how data will be used [37] and stored. On the contrary, the DT requests information from the CT on the house floor plan to design/customise the sensors according to the requirements [34, 35]. The above situation can often land the CT in a dilemma, as projects often decide which sensors will be deployed and data collected late in the project. Furthermore, the CT cannot tell the citizens about the sensors until the project's data and requirements are well defined. Citizens can only decide whether they want to participate in the project once they have clarity on what is collected, which means that the CT cannot provide house details to the DT. Therefore, it is better to perform a requirement analysis (§ V-A) earlier in the project to understand data collection and be transparent with citizens.

G. Operational Challenges

Research projects also have operational challenges, which are problems that arise and can render a project less efficient.

Skills shortage: A significant challenge is the shortage of people with the appropriate skill set to act as system architects in urban monitoring research projects. Research projects (a collaboration between universities, industry, and city councils) are often for 1-5 years. The people who develop and manage the urban monitoring platform are research associates and doctoral students, who mainly cover only part of the required skill set. Furthermore, students who maintain the project often work part-time due to semesters and other courses, leading to staffing problems [63]. Experience and knowledge in system administration, cloud infrastructure, networking, DevOps, and cybersecurity are required [131, 146, 147].

Different expectations and goals: Research projects can have multiple partners and collaborations. Each partner can have a different set of expertise, business models, expectations and their own project agenda on how it benefits them [146].

There may be cases where collaboration priorities are different, which can create challenges in communication and work completion. Teamwork is essential for project success [9, 146]. Furthermore, research members can have other KPIs on which their managers judge their performance. If the delivery of the research project is not one of them, it can affect the researcher's commitment to the project. There will always be members in the project who will be hard working, average working, and who would cause trouble; always good to identify the right person for the right work.

Clear, concise communication: Research projects often include multiple meetings to discuss various objectives and goals of the project. It is crucial to have clearly defined agendas and final takeaways. Also, it is a good practice to invite only a few key people or technical leads to the meeting for clear and concise communication. In addition, face-to-face meetings are preferred over online discussions, especially brainstorming sessions. Things become delayed if the parties involved do not communicate clearly and concisely.

Risk Management: The research project should also have risk management that considers different issues in the project schedule. Risks could include COVID-19 affecting people, datasets not available for analysis, delays in setting up the testbeds, deployment of devices in public spaces, and related safety issues (electrical hazards, devices falling from streetlamps), among others. Furthermore, it should include critical personal backup plans if someone gets sick or leaves the project/company. Furthermore, suppose that the deployed devices are expected to work after the end of the research phase. In that case, it is essential to have a handover-takeover (HOTO) (including hiring and transferring skills) to continue a successful project. Often, the platform and devices require some human intervention to operate [20].

Infrastructure availability: There will be inevitable situations outside the control of the research team. For example, infrastructure suffers from an outage, a global internet outage, or installed devices affected by weather [20]. As another example, there is little the DT can do if the cloud tier is hosted on city-council infrastructure and an outage occurs with their main administrator on leave. Case in point, the Internet recently suffered a significant outage of approximately one hour [148], leaving multiple cloud services unavailable.

Devices required for deployment must be purchased early. Importing devices from another country and connecting them to the home network is expensive and challenging. A significant amount of time is lost in the shipment of devices across continents, exacerbated by having to work in multiple timezones [30].

Partnerships: It is essential to have the support and partnership of the city council [80]. The city council officials can act as a catalyst for informing and organising discussions with other city departments (electricity board, hospitals, recycling). These other departments can update the city council about the project and ask for their input on deployment locations or how the project can support a particular department in solving its challenges. The research project, depending on its objective, can support the vision of the city plan (usually published year-by-year, such as the Bristol city plan [149], Belfast Agenda [150], Chicago Technology plan [151]) in terms of how the research project and the deployment of the public infrastructure can allow the city to use technology and data for engagement, innovation, inclusion, and opportunity.

In addition, it is essential to engage and win the confidence of city departments and employees by involving them in the project. For example, suppose that the infrastructure will be installed on city streetlamps. In that case, it is important to bring prototype units to the electrical department and seek their input on electrical safety and mounting procedure, effectively gaining their confidence and working as a team toward a common goal.

Logistics: The DT should be aware of the design of the nodes, the installation procedures, the node deployment locations, and other information. In addition, they should have ownership and power to make decisions on the fly, such as moving a node to a different street corner due to a blocked view during installation. Interactions and conversations can lead to collaborations and understanding of how research data collected by public deployment can be used and integrated into existing city data platforms (such as Bristol Open Data [88], London Datastore [89]).

Furthermore, DT can create communication channels such as surveys and forms to collect the location of the node deployment, the type of data, and the problems to be solved from the project stakeholders, city departments, communities, research groups, and residents [90].

VI. Conclusion

The continued growth of wireless technologies has resulted in significant research into urban monitoring via data-gathering IoT testbeds. These research testbeds follow a typical three-tier architecture, and many designs and implementation challenges remain, including data privacy controls, network security, and device updates. We extracted these challenges and associated lessons learned by considering several real-world IoT testbed projects. We analyzed the projects in the context of the V-model development life cycle phases. We presented the project challenges and lessons learned organized by requirements analysis, system design, implementation, testing and deployment phases. We believe this will assist other urban monitoring researchers in planning future testbeds. We hope this research will prove valuable and reduce these projects' design and implementation costs.

Acknowledgment

This work has been supported in part by EPSRC through grant no EP/P016782/1 (UKCRIC Urban Observatories) and an Industrial CASE award sponsored by BT.

References

[1] U. D. of Economic and S. A. P. Division, "World urbanization prospects: The 2018 revision," 2018.

[2] Z. Khan, A. Anjum, and S. L. Kiani, "Cloud based big data analytics for smart future cities," in *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*. IEEE, 2013, pp. 381–386.

[3] P. Woznowski, A. Burrows, T. Diethe, X. Fafoutis, J. Hall, S. Hannuna, M. Camplani, N. Twomey, M. Kozlowski, B. Tan *et al.*, "Sphere: A sensor platform for healthcare in a residential environment," in *Designing, developing, and facilitating smart cities*. Springer, 2017, pp. 315–333.

[4] A. Elsts, X. Fafoutis, G. Oikonomou, R. Piechocki, and I. Craddock, "Tsch networks for health iot: Design, evaluation, and trials in the wild," *ACM Transactions on Internet of Things*, vol. 1, no. 2, pp. 1–27, 2020.

[5] "Replicate: Renaissance of places with innovative citizenship and technology," https://replicate-project.eu/, accessed: 2021-06-30.

[6] "Twinergy - digital twin," https://www.twinergy.eu/, accessed: 2021-06-30.

[7] V. Kumar, P. Yadav, and L. S. Indrusiak, "Resilient edge: Building an adaptive and resilient multi-communication network for iot edge using lpwan and wifi," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2023.

[8] C. E. Catlett, P. H. Beckman, R. Sankaran, and K. K. Galvin, "Array of Things: A Scientific Research Instrument in the Public Way: Platform Design and Early Lessons Learned," *Proceedings of the 2Nd International Workshop on Science of Smart City Operations and Platforms Engineering*, pp. 26–33, 2017. [Online]. Available: http://doi.acm.org/10.1145/3063386.3063771

[9] S. Kurkovsky and C. Williams, "Raspberry Pi as a platform for the internet of things projects: Experiences and lessons," *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, vol. Part F128680, pp. 64–69, 2017.

[10] J. Wan, D. Li, C. Zou, and K. Zhou, "M2m communications for smart city: An event-based architecture," in *2012 IEEE 12th International Conference on Computer and Information Technology*, 2012, pp. 895–900.

[11] L. Gurgen, O. Gunalp, Y. Benazzouz, and M. Gallissot, "Self-aware cyber-physical systems and applications in smart buildings and cities," in *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2013, pp. 1149–1154.

[12] I. Butun, P. Österberg, and H. Song, "Security of the internet of things: Vulnerabilities, attacks, and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 616–644, 2020.

[13] E. Coraggio, D. Han, W. Liu, and T. Tryfonas, "Smart cities: Real-time water quality monitoring and prediction. paper presented at international association for hydro-environment engineering and research world congress 2019, panama city, panama." *International Association for Hydro-Environment Engineering and Research World Congress 2019: Water-Connecting the World*, 2019.

[14] Y. Chen and D. Han, "Water quality monitoring in smart city: A pilot project," *Automation in Construction*, vol. 89, pp. 307–316, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580517305988

[15] S. Gunner, P. J. Vardanega, T. Tryfonas, J. H. Macdonald, and R. E. Wilson, "Rapid deployment of a wsn on the clifton suspension bridge, uk," *Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction*, vol. 170, no. 3, pp. 59–71, 2017.

[16] S. Gunner, "Using telematics to gather user behaviour data from a fleet of electric bicycles," 2021, [Online; accessed 30-May-2022]. [Online]. Available: https://ercim-news.ercim.eu/en127/special/using-telematics-to-gather-user-behaviour-data-from-a-fleet-of-electric-bicycles

[17] D. Nepomuceno, T. Tryfonas, and P. Vardanega, "Residential damp detection with temperature and humidity urban sensing," in *International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making*. ICE Publishing, 2019, pp. 605–611.

[18] "Sphere - a sensor platform for healthcare in a residential environment," https://www.bristol.ac.uk/engineering/research/digital-health/research/sphere/, accessed: 2021-06-30.

[19] Y. Oyedele, P. Dlamini, D. V. Van Greunen, and T. Chizema, "Lessons learnt from deploying an IoT sensing system for e-Agriculture in South Africa," *2021 11th IEEE Global Humanitarian Technology Conference, GHTC 2021*, pp. 208–212, 2021.

[20] G. Jackson, D. Wilson, S. Gallacher, and J. A. McCann, "Tales from the Wild: Lessons Learned from Creating a Living Lab," *FAILSAFE 2017 - Proceedings of the 1st ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems, Part of SenSys 2017*, p. 62, 2017.

[21] M. Chowdhury, M. Rahman, A. Rayamajhi, S. M. Khan, M. Islam, Z. Khan, and J. Martin, "Lessons learned from the real-world deployment of a connected vehicle testbed," *Transportation Research Record*, vol. 2672, no. 22, pp. 10–23, 2018.

[22] S. C. Folea and G. D. Mois, "Lessons Learned from the Development of Wireless Environmental Sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 3470–3480, 2020.

[23] A. H. Dehwah, M. Mousa, and C. G. Claudel, "Lessons learned on solar powered wireless sensor network deployments in urban, desert environments," *Ad Hoc Networks*, vol. 28, pp. 52–67, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2015.01.013

[24] A. Valera, P. Lee, H. P. Tan, H. X. Tan, and H. Liang, "Real world, large scale iot systems for community eldercare: Experiences and lessons learned," *Elderly Care: Options, Challenges and Trends*, pp. 53–80, 2018.

[25] T. Watteyne, C. Adjih, and X. Vilajosana, "Lessons learned from large-scale dense ieee802.15.4 connectivity traces," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 2015, pp. 145–150.

[26] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes experiences from a pilot sensor network

deployment in precision agriculture," *20th International Parallel and Distributed Processing Symposium, IPDPS 2006*, vol. 2006, 2006.

[27] M. R. Porto, D. Hildebrandt, M. Arias, A. Fuentes, M. Perez, K. O'Malley, R. Knights, and J. Brookes, "3e-Houses FINAL REPORT," European Commission, Joint Research Centre, Smart Electricity Systems and Interoperability, Tech. Rep., 2013.

[28] C. Mydlarz, M. Sharma, Y. Lockerman, B. Steers, C. Silva, and J. P. Bello, "The life of a new york city noise sensor network," *Sensors (Switzerland)*, vol. 19, no. 6, pp. 1–24, 2019.

[29] A. Cenedese, A. Zanella, L. Vangelista, and M. Zorzi, "Padova smart city: An urban internet of things experimentation," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, 2014, pp. 1–6.

[30] B. Basnyat, N. Singh, N. Roy, and A. Gangopadhyay, "Design and Deployment of a Flash Flood Monitoring IoT: Challenges and Opportunities," *Proceedings - 2020 IEEE International Conference on Smart Computing, SMARTCOMP 2020*, pp. 422–427, 2020.

[31] P. Sotres, J. R. Santana, L. Sanchez, J. Lanza, and L. Munoz, "Practical Lessons from the Deployment and Management of a Smart City Internet-of-Things Infrastructure: The SmartSantander Testbed Case," *IEEE Access*, vol. 5, pp. 14 309–14 322, 2017.

[32] S. Latré, P. Leroux, T. Coenen, B. Braem, P. Ballon, and P. Demeester, "City of things: An integrated and multi-technology testbed for IoT smart city experiments," *IEEE 2nd International Smart Cities Conference: Improving the Citizens Quality of Life, ISC2 2016 - Proceedings*, 2016.

[33] A. Elsts, R. Balass, J. Judvaitis, R. Zviedris, G. Strazdins, A. Mednis, and L. Selavo, "Sadmote: A robust and cost-effective device for environmental monitoring," in *Architecture of Computing Systems – ARCS 2012*, A. Herkersdorf, K. Römer, and U. Brinkschulte, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 225–237.

[34] T. W. Hnat, V. Srinivasan, J. Lu, T. I. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse, "The hitchhiker's guide to successful residential sensing deployments," *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems - SenSys '11*, p. 232, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2070942.2070966

[35] P. Lundrigan, K. T. Min, N. Patwari, S. K. Kasera, K. Kelly, J. Moore, M. Meyer, S. C. Collingwood, F. Nkoy, B. Stone, and K. Sward, "EpiFi: An in-home IoT architecture for epidemiological deployments," *Proceedings of the 43rd Annual IEEE Conference on Local Computer Networks, LCN Workshops 2018*, pp. 30–37, 2019.

[36] Y. Huang, Y. C. Chen, C. W. You, D. X. Wu, Y. L. Chen, K. L. Hua, and J. Y. J. Hsu, "Toward an easy deployable outdoor parking system - Lessons from long-term deployment," *2017 IEEE International Conference on Pervasive Computing and Communications, PerCom 2017*, pp. 227–236, 2017.

[37] S. H. Yang, X. Chen, X. Chen, L. Yang, B. Chao, and J. Cao, "A case study of internet of things: A wireless household water consumption monitoring system," *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, pp. 681–686, 2015.

[38] "Smartsantander experimental test facilities," https://www.smartsantander.eu/index.php/testbeds/, accessed: 2021-06-30.

[39] "Umbrella testbed - open, programmable iot testbed," https://www.umbrellaiot.com/what-is-umbrella/umbrella-testbed/, accessed: 2021-06-30.

[40] L. Nussbaum, "Testbeds support for reproducible research," in *Proceedings of the Reproducibility Workshop*, ser. Reproducibility '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 24–26. [Online]. Available: https://doi.org/10.1145/3097766.3097773

[41] P. Yadav, J. A. McCann, and T. Pereira, "Self-synchronization in duty-cycled internet of things (iot) applications," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2058–2069, 2017.

[42] A. Feraudo, P. Yadav, V. Safronov, D. A. Popescu, R. Mortier, S. Wang, P. Bellavista, and J. Crowcroft, "Colearn: Enabling federated learning in mud-compliant iot edge networks," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, ser. EdgeSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 25–30. [Online]. Available: https://doi.org/10.1145/3378679.3394528

[43] P. Bellavista, J. Berrocal, A. Corradi, S. K. Das, L. Foschini, and A. Zanni, "A survey on fog computing for the internet of things," *Pervasive and mobile computing*, vol. 52, pp. 71–99, 2019.

[44] J. Lin and J. Anderson, "From IoT to Cloud : Research Platform for IoT / Cloud Experiments," *Scientific Programming*, pp. 1–2, 2019.

[45] "Chameleon - a configurable experimental environment for large-scale edge to cloud research," https://www.chameleoncloud.org/, accessed: 2021-06-30.

[46] "Geni - open infrastructure for at-scale networking and distributed systems research and education," https://www.geni.net/, accessed: 2021-06-30.

[47] "Grid5000 - large-scale and flexible testbed for experiment-driven research," https://www.grid5000.fr/w/Grid5000:Home, accessed: 2021-06-30.

[48] "Fed4fire+," https://www.fed4fire.eu/, accessed: 2021-06-30.

[49] "Fit cloudlab," https://www.cloudlab.us/, accessed: 2021-06-30.

[50] "Emulab - network testbed," https://www.emulab.net/portal/frontpage.php, accessed: 2021-06-30.

[51] "Planetlab - open platform for developing, deploying, and accessing planetary scale services," https://planetlab.cs.princeton.edu/, accessed: 2021-06-30.

[52] "Pragma - pacific rim applications and grid middle-

ware assembly," http://www.pragma-grid.net/resources/testbed/, accessed: 2021-06-30.

[53] "Deter lab - cyber defense technology experimental research laboratory," https://deter-project.org/about_deterlab, accessed: 2021-06-30.

[54] "Nornet - core - real world, large scale, multi-homing testbed," https://www.nntb.no/nornet-core/, accessed: 2021-06-30.

[55] "Savi - smart applications on virtual infrastructure," https://www.savinetwork.ca/, accessed: 2021-06-30.

[56] "Fit-iot - a very large scale iot testbed," https://www.iot-lab.info/, accessed: 2021-06-30.

[57] "Citylab, the city of things smart cities fire testbed," https://doc.lab.cityofthings.eu/wiki/Main_Page, accessed: 2021-06-30.

[58] "3e-houses," https://ses.jrc.ec.europa.eu/3e-houses, accessed: 2021-06-30.

[59] S. Y. H. S. S. MATSUMOTO and M. NAKAMURA, "Scallop4sc: Data platform for storing and processing large-scale house log in smart city."

[60] Y. W. Lee and S. Rho, "U-city portal for smart ubiquitous middleware," in *2010 The 12th International Conference on Advanced Communication Technology (ICACT)*, vol. 1. IEEE, 2010, pp. 609–613.

[61] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial internet of things: Architecture, advances and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.

[62] I. Bevers, "Intelligence at the edge part 1: The edge node | analog devices," pp. 1–6, 2019. [Online]. Available: https://www.analog.com/en/technical-articles/intelligence-at-the-edge-part-1-the-edge-node.html

[63] S. Gvk, T. Adhisaya, P. Aswini, J. Bapat, and D. Das, "Challenges in the design of an IoT testbed," *2019 2nd International Conference on Intelligent Communication and Computational Techniques, ICCT 2019*, pp. 14–19, 2019.

[64] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, and D. Pfisterer, "SmartSantander: IoT experimentation over a smart city testbed," *Computer Networks*, vol. 61, no. January, pp. 217–238, 2014.

[65] I. Beavers, "Intelligence at the Edge Part 2 : Reduced Time to Insight," 2019. [Online]. Available: https://www.analog.com/en/technical-articles/intelligence-at-the-edge-part-2-reduced-time-to-insight.html

[66] E. F. Z. Santana, A. P. Chaves, M. A. Gerosa, F. Kon, and D. S. Milojicic, "Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture," *ACM Comput. Surv.*, vol. 50, no. 6, nov 2017. [Online]. Available: https://doi.org/10.1145/3124391

[67] X. Fafoutis, A. Elsts, R. Piechocki, and I. Craddock, "Experiences and Lessons Learned from Making IoT Sensing Platforms for Large-Scale Deployments," *IEEE Access*, vol. 6, pp. 3140–3148, 2017.

[68] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "The hitchhiker's guide to successful wireless sensor network deployments," *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*, pp. 43–56, 2008. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1460412.1460418

[69] B. Cheng, S. Longo, F. Cirillo, M. Bauer, and E. Kovacs, "Building a Big Data Platform for Smart Cities: Experience and Lessons from Santander," *Proceedings - 2015 IEEE International Congress on Big Data, BigData Congress 2015*, pp. 592–599, 2015.

[70] J. Schleich, M. Klobasa, and S. Gölz, "Lessons Learned on Home Energy Monitoring and Management: Smartcity Málaga," *9th International Conference on the European Energy Market, EEM 12*, pp. 263–264, 2012.

[71] A. Elsts, G. Oikonomou, X. Fafoutis, and R. Piechocki, "Internet of Things for smart homes: Lessons learned from the SPHERE case study," *GIoTS 2017 - Global Internet of Things Summit, Proceedings*, 2017.

[72] J. Beaudin, S. Intille, and E. M. Tapia, "Lessons learned using ubiquitous sensors for data collection in real homes," in *CHI'04 extended abstracts on Human factors in computing systems*, 2004, pp. 1359–1362.

[73] K. Webb, M. Hibler, R. Ricci, A. Clements, and J. Lepreau, "Implementing the emulab-planetlab portal: Experience and lessons learned," *1st USENIX Workshop on Real, Large Distributed Systems, WORLDS 2004*, 2004.

[74] K. Keahey, J. Anderson, Z. Zhen, P. Riteau, P. Ruth, D. Stanzione, M. Cevik, J. Colleran, H. S. Gunawi, C. Hammock, J. Mambretti, A. Barnes, F. Halbach, A. Rocha, and J. Stubbs, "Lessons learned from the chameleon testbed," *Proceedings of the 2020 USENIX Annual Technical Conference, ATC 2020*, pp. 219–233, 2020.

[75] S. Y. Kumar R and C. H. N, "An Extensive Review on Sensing as a Service Paradigm in IoT: Architecture, Research Challenges, Lessons Learned and Future Directions," *International Journal of Applied Engineering Research*, vol. 14, no. 6, pp. 1220–1243, 2019. [Online]. Available: http://www.ripublication.com

[76] R. C. Staudemeyer, H. C. Pöhls, and M. Wójcik, "What it takes to boost Internet of Things privacy beyond encryption with unobservable communication: a survey and lessons learned from the first implementation of DC-net," *Journal of Reliable Intelligent Environments*, vol. 5, no. 1, pp. 41–64, 2019. [Online]. Available: https://doi.org/10.1007/s40860-019-00075-0

[77] U. Ekedahl, R. C. Mihailescu, and Z. Ma, "Lessons learned from adapting "things" to IoT platforms in research and teaching," *Proceedings of the ACM Symposium on Applied Computing*, pp. 1457–1460, 2018.

[78] M. R. Palattella, X. Vilajosana, T. Chang, M. A. R. Ortega, and T. Watteyne, "Lessons learned from the 6TiSCH plugtests," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommu-*

nications Engineering, LNICST, vol. 170, pp. 415–426, 2016.

[79] J. Kim, J. Yun, S.-C. Choi, D. N. Seed, G. Lu, M. Bauer, A. Al-Hezmi, K. Campowsky, and J. Song, "Standard-based iot platforms interworking: implementation, experiences, and lessons learned," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 48–54, 2016.

[80] P. Lago, R. Verdecchia, N. Condori-Fernandez, E. Rahmadian, J. Sturm, T. van Nijnanten, R. Bosma, C. Debuysscher, and P. Ricardo, "Designing for sustainability: Lessons learned from four industrial projects," in *Progress in IS*. Springer International Publishing, dec 2020, pp. 3–18. [Online]. Available: https://doi.org/10.1007%2F978-3-030-61969-5_1

[81] K. Forsberg, H. Mooz, and H. Cotterman, *Visualizing Project Management: Models and Frameworks for Mastering Complex Systems*. John Wiley & Sons, Nov. 2005.

[82] P. Rook, "Controlling software projects," *Software Engineering Journal*, vol. 1, no. 1, pp. 7–16, Jan. 1986.

[83] W. W. Royce, "Managing the development of large software systems: concepts and techniques," in *Proceedings of the 9th international conference on Software Engineering*. blog.jbrains.ca, 1987, pp. 328–338.

[84] K. Forsberg and H. Mooz, "The relationship of system engineering to the project cycle," *INCOSE Int. Symp.*, vol. 1, no. 1, pp. 57–65, Oct. 1991.

[85] S. Chatterjee, J. Byun, K. Dutta, R. U. Pedersen, A. Pottathil, and H. Q. Xie, "Designing an Internet-of-Things (IoT) and sensor-based in-home monitoring system for assisting diabetes patients: iterative learning from two case studies," *European Journal of Information Systems*, vol. 27, no. 6, pp. 670–685, 2018. [Online]. Available: https://doi.org/10.1080/0960085X.2018.1485619

[86] European Parliament and the Council of the European Union, "2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation," *Regulation (EU)*, 2016.

[87] A. Elsts, X. Fafoutis, P. Woznowski, E. Tonkin, G. Oikonomou, R. Piechocki, and I. Craddock, "Enabling Healthcare in Smart Homes: The SPHERE IoT Network Infrastructure," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 164–170, 2018.

[88] "Open data bristol," https://opendata.bristol.gov.uk/pages/homepage/, accessed: 2021-06-30.

[89] "London datastore - greater london," https://data.london.gov.uk/, accessed: 2021-06-30.

[90] P. Committee, K. Cagney, C. Catlett, P. Beckman, K. K. Galvin, M. Potosnak, D. Work, D. Pancoast, R. Team, P. Committee, W. Barbour, J. Dunn, N. Ferrier, V. Forgione, D. Gloudemans, R. Kotamarthi, R. Mitchum, R. Sankaran, and V. Welch, "c," *University of Chicago*, no. August, 2018.

[91] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Data collection and wireless communication in internet of things (iot) using economic analysis and pricing models: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2546–2590, 2016.

[92] M. Mendula, S. Khodadadeh, S. S. Bacanli, S. Zehtabian, H. U. Sheikh, L. Bölöni, D. Turgut, and P. Bellavista, "Interaction and behaviour evaluation for smart homes: Data collection and analytics in the scaledhome project," in *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2020, pp. 225–233.

[93] E. Bout, V. Loscri, and A. Gallais, "How machine learning changes the nature of cyberattacks on iot networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 248–279, 2022.

[94] J. Liu, M. Nogueira, J. Fernandes, and B. Kantarci, "Adversarial machine learning: A multilayer review of the state-of-the-art and challenges for wireless and mobile systems," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 123–159, 2022.

[95] R. Román, J. Zhou, and J. López, "On the features and challenges of security and privacy in distributed internet of things," *Comput. Networks*, vol. 57, pp. 2266–2279, 2013.

[96] C. Kalloniatis, H. Mouratidis, M. Vassilis, S. Islam, S. Gritzalis, and E. Kavakli, "Towards the design of secure and privacy-oriented information systems in the cloud: Identifying the major concepts," *Computer Standards & Interfaces*, vol. 36, no. 4, pp. 759–775, 2014, security in Information Systems: Advances and new Challenges. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920548913001840

[97] N. Shevchenko, T. A. Chick, P. O. Riordan, T. P. Scanlon, and C. Woody, "Threat Modeling : a Summary of Available Methods," *Research Report*, no. July, p. 26, 2018. [Online]. Available: https://apps.dtic.mil/sti/citations/AD1084024%0Ahttps://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf

[98] "A guide to threat modelling for developers," https://martinfowler.com/articles/agile-threat-modelling.html, accessed: 2020-06-30.

[99] "Threat modeling," https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling, accessed: 2020-06-30.

[100] M. Sweney. (2020) Bristol is worst uk city for broadband outages with 169 hours a year. Accessed:2020-06-30. [Online]. Available: https://www.theguardian.com/technology/2020/aug/13/bristol-is-worst-uk-city-for-broadband-outages-with-169-hours-a-year

[101] U. Communications, "Reboot your computer for best performance," 2019. [Online]. Available: https://www.cu.edu/blog/tech-tips/reboot-your-computer-best-performance-0

[102] "Thermal testing raspberry pi 4," https://www.raspberrypi.org/blog/thermal-testing-raspberry-pi-4/, accessed: 2021-06-30.

[103] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko *et al.*, "Openiot: Open source internet-of-things in the cloud," in *Interoperability and open-source solutions for the internet of things*. Springer, 2015, pp. 13–25.

[104] "Tutorial: Rpl border router," https://github.com/contiki-ng/contiki-ng/wiki/Tutorial:-RPL-border-router, accessed: 2021-06-30.

[105] K. H. Law and J. P. Lynch, "Smart city: Technologies and challenges," *IT Professional*, vol. 21, no. 6, pp. 46–51, 2019.

[106] C. Wu, D. Birch, D. Silva, C.-H. Lee, O. Tsinalis, and Y. Guo, "Concinnity: A generic platform for big sensor data applications," *IEEE Cloud Computing*, vol. 1, no. 2, pp. 42–50, 2014.

[107] F. J. Villanueva, M. J. Santofimia, D. Villa, J. Barba, and J. C. Lopez, "Civitas: The smart city middleware, from sensors to big data," in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, 2013, pp. 445–450.

[108] W. Apolinarski, U. Iqbal, and J. X. Parreira, "The gambas middleware and sdk for smart city applications," in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, 2014, pp. 117–122.

[109] J. A. Galache, T. Yonezawa, L. Gurgen, D. Pavia, M. Grella, and H. Maeomichi, "Clout: Leveraging cloud computing techniques for improving management of massive iot data," in *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*. IEEE, 2014, pp. 324–327.

[110] S. Girtelschmid, M. Steinbauer, V. Kumar, A. Fensel, and G. Kotsis, "Big data in large scale intelligent smart city installations," in *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, 2013, pp. 428–432.

[111] I. Vilajosana, J. Llosa, B. Martinez, M. Domingo-Prieto, A. Angles, and X. Vilajosana, "Bootstrapping smart cities through a self-sustainable model based on big data flows," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 128–134, 2013.

[112] Home Assistant, "Docker monitor broken on debian 11," 2022, [Online; accessed 30-May-2022]. [Online]. Available: https://community.home-assistant.io/t/docker-monitor-broken-on-debian-11/333880

[113] T. Gea, J. Paradells, M. Lamarca, and D. Roldan, "Smart cities as an application of internet of things: Experiences and lessons learnt in barcelona," *Proceedings - 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2013*, pp. 552–557, 2013.

[114] "Smart citizen docs," https://docs.smartcitizen.me/Smart%20Citizen%20Kit/, accessed: 2021-06-30.

[115] "Luftdaten sensor.community," https://sensor.community/en/, accessed: 2021-06-30.

[116] "Tidc-cc2650stk-sensortag," https://www.ti.com/tool/TIDC-CC2650STK-SENSORTAG, accessed: 2021-06-30.

[117] "Tp-link smart plugs," https://www.tp-link.com/uk/home-networking/smart-plug/, accessed: 2021-06-30.

[118] O. M. Alliance, "Lightweight machine to machine requirements," *Candidate Version*, vol. 1, 2013.

[119] S. Kim, K.-J. Park, and C. Lu, "A survey on network security for cyber–physical systems: From threats to resilient design," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1534–1573, 2022.

[120] S. P. Le Blond, A. Holt, and P. White, "3eHouses: A smart metering pilot in UK living labs," *IEEE PES Innovative Smart Grid Technologies Conference Europe*, 2012.

[121] K. Brun-Laguna, P. Minet, T. Watteyne, and P. Henrique Gomes, "Moving beyond Testbeds? Lessons (We) Learned about Connectivity," *IEEE Pervasive Computing*, vol. 17, no. 4, pp. 15–27, 2018.

[122] A. Baid, S. Mathur, I. Seskar, S. Paul, A. Das, and D. Raychaudhuri, "Spectrum mri: Towards diagnosis of multi-radio interference in the unlicensed band," in *2011 IEEE Wireless Communications and Networking Conference*, 2011, pp. 534–539.

[123] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, "The contiki-ng open source operating system for next generation iot devices," *SoftwareX*, vol. 18, p. 101089, 2022.

[124] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6tisch: deterministic ip-enabled industrial internet (of things)," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 36–41, 2014.

[125] A. Verma and V. Ranga, "Security of rpl based 6lowpan networks in the internet of things: A review," *IEEE Sensors Journal*, vol. 20, no. 11, pp. 5666–5690, 2020.

[126] L. Belli, S. Cirani, L. Davoli, A. Gorrieri, M. Mancin, and M. Picone, "Design and Deployment Oriented Testbed," *IEEE Computer*, vol. 48, no. 9, pp. 32–40, 2015. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.704.5835&rep=rep1&type=pdf

[127] E. G. Gran, T. Dreibholz, and A. Kvalbein, "NorNet Core - A multi-homed research testbed," *Computer Networks*, vol. 61, pp. 75–87, 2014.

[128] "Turns out that florida water treatment facility left the doors wide open for hackers," https://www.theverge.com/2021/2/10/22277300/florida-water-treatment-chemical-tamper-teamviewer-shared-password, accessed: 2021-06-30.

[129] A. Del Sole, *Building Applications with Python*. Berkeley, CA: Apress, 2021, pp. 211–233. [Online]. Available: https://doi.org/10.1007/978-1-4842-6901-5_10

[130] P. Abate and R. Di Cosmo, "Predicting upgrade failures using dependency analysis," in *2011 IEEE 27th International Conference on Data Engineering Workshops*, April 2011, pp. 145–150.

[131] K. Keahey, J. Anderson, P. Ruth, J. Colleran, C. Ham-

mock, J. Stubbs, and Z. Zhen, "Operational lessons from chameleon," *ACM International Conference Proceeding Series*, 2019.

[132] H. J. Syed, A. Gani, R. W. Ahmad, M. K. Khan, and A. I. A. Ahmed, "Cloud monitoring: A review, taxonomy, and open research issues," *Journal of Network and Computer Applications*, vol. 98, pp. 11–26, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804517302783

[133] *The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition*, 2000.

[134] "Tp-link smart plug," https://www.tp-link.com/uk/home-networking/smart-plug/, accessed: 2020-06-30.

[135] "Tesla powerwall," https://www.tesla.com/en_gb/powerwall, accessed: 2020-06-30.

[136] "Open energy monitoring," https://openenergymonitor.org/, accessed: 2020-06-30.

[137] M. Conti and S. Giordano, "Mobile ad hoc networking: Milestones, challenges, and new research directions," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 85–96, 2014.

[138] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous internet of things build our future: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2011–2027, 2018.

[139] S. Wahle, T. Magedanz, and F. Schulze, "The openmtc framework—m2m solutions for smart cities and the internet of things," in *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2012, pp. 1–3.

[140] T. Benzel, R. Braden, D. Kim, C. Neuman, A. Joseph, K. Sklower, R. Ostrenga, and S. Schwab, "Design, deployment, and use of the DETER testbed," *DETER Community Workshop on Cyber Security Experimentation and Test 2007, DETER 2007*, no. August, 2007.

[141] T. Auer and M. Felderer, "Towards a Learning Environment for Internet of Things Testing with LEGO® MINDSTORMS®," *Proceedings - 2020 IEEE 13th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2020*, pp. 457–460, 2020.

[142] P. Emami-Naeini, "Privacy and security nutrition labels to inform IoT consumers." USENIX Association, Feb. 2021.

[143] K. W. M. Centre, "3-E HOUSES Best practice guide," European Commission, Joint Research Centre, Smart Electricity Systems and Interoperability, Tech. Rep., 2013.

[144] "Nextdoor - tap into your neighbourhood," https://nextdoor.co.uk/, accessed: 2021-06-30.

[145] "Sphere: sensors for the home environment," https://www.youtube.com/watch?v=dsIxMBYoo84, accessed: 2021-06-30.

[146] M. Sony and P. S. Aithal, "Practical Lessons for Engineers to adapt towards Industry 4.0 in Indian Engineering Industries," *International Journal of Case Studies in Business, IT, and Education*, no. 102874, pp. 86–97, 2020.

[147] P. Ballon, J. Glidden, P. Kranas, A. Menychtas, S. Ruston, and S. Van Der Graaf, "Is there a need for a cloud platform for european smart cities," in *eChallenges e-2011 Conference Proceedings, IIMC International Information Management Corporation*, 2011, pp. 1–7.

[148] "Major internet outage 'shows infrastructure needs urgent fixing'," https://www.theguardian.com/technology/2021/jun/08/security-warning-error-cloud-websites-offline-outage, accessed: 2020-06-30.

[149] Bristol City Council. (2020) One City Plan 2020.

[150] Belfast CIty Council. (2014) The Belfast Agenda. [Online]. Available: https://smartbelfast.city/wp-content/uploads/2018/04/Belfast-Agenda-3.9MB-PDF.pdf

[151] The City of Chicago. (2013) The City Of Chicago Technology Plan. [Online]. Available: https://techplan.cityofchicago.org/wp-content/uploads/2013/09/cityofchicago-techplan.pdf%0Apapers3://publication/uuid/01F5BFDF-24BE-4F94-8256-4A619D58A87D