An Urban Sensing Architecture as Essential Infrastructure for Future Smart Cities Research

By

VIJAY KUMAR

Department of Civil Engineering
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

FEBRUARY 2023

Word count: fifty thousand two hundred thirty two

Abstract

Half of the world's population lives in urban environments, and about 3 million people move to cities each week. Maintaining public services with increased urbanisation is challenging. Smart cities enable the city council to improve services and citizen lives. However, the smart cities concept has been present for decades and has yet to reach its full potential. The vast majority of city initiatives to date are pilot projects funded by research and innovation grants rather than sustainable and repeatable solutions. In addition to political and policy challenges, there are challenges related to urban data collection, analysis, and maintenance of city infrastructure in a reliable, secure and resilient way. Our work presents four contributions.

First, we document the length and breadth of urban data from the citizen to the city scale that can be collected, analysed, and interlinked to make a city smart and reap its full benefits. Digitalisation and the Internet of Things help collect urban data and provide opportunities to analyse and provide information to policymakers, reduce costs, and increase productivity.

To collect the above urban data and make a city smart, research organisations, in collaboration with the city council, deploy proprietary IoT infrastructure often composed of the Endpoint, the Edge, and the Cloud. Managing the operation of the IoT infrastructure while considering the security and privacy challenges that emerge, such as data privacy controls, network security, and device security updates, is challenging. Second, we systematically review the above challenges to facilitate future smart city research projects to reduce implementation time and deliver secure and resilient infrastructure.

Similarly, other organisations also deploy infrastructure in public spaces for various purposes (free Wi-Fi, advertising and display systems). However, these are independent solutions with significant duplication in infrastructure and similar requirements. Suppose there is a way to share the IoT infrastructure between multiple projects and organisations. It can allow us to create new services and have better co-created smart cities, speeding up the implementation of new services economically, enabling better resource management, and reducing costs for city councils and other organisations. Third, we provide a smart city framework to solve the smart city challenges and requirements and share it between different organisations and research projects. The framework consists of multiple modules and sub-modules and based on the organisation requirements a specific module/sub-module can be integrated with an existing pre-built infrastructure. The modules/sub-modules are implemented using open-source components having a particular shell life and there could be multiple components serving the same purpose that can be interchanged based on technological compatibility.

The above infrastructure often has an edge component connecting endpoints to the cloud, storing, processing data, and running multiple urban applications. The resiliency and reliability requirements of applications running on the edge vary from non-critical to safety-critical with time-bounded guarantees. The network connectivity of IoT edge devices remains the critical component that needs to be met. Fourth, we investigate how to meet IoT applications' mixed-

criticality QoS requirements in multi-communication networks.

The thesis aims to highlight and attempt to solve the challenges faced by the smart city research project and bring the smart city milestones closer to reality. Therefore, we hope the work will inspire future research on shared infrastructure, resilience, smart cities, and efficient deployment and management of smart city infrastructure. The key contribution is the conceptual architecture derived from smart city projects. The other contribution is the use of a multi-protocol gateway at the edge to increase network resilience.

Dedication and acknowledgements

This PhD would not be possible without the support of kind, talented and supportive people who have contributed in some way or another. Thank you to James Pope, Theodoros Spyridopoulos, Sam Gunner, Antonis Vafeas, Maria Pregnolato, and Leandro Soares Indrusiak. These are the lovely talented people with whom I have collaborated during my PhD. Thank you for your contributions, guidance, reviewing of articles and, more importantly, providing hope that we will get there.

Thank you to Poonam Yadav for the opportunity to work with her in an internship that also led to the content included in this thesis. The energy, genuineness, turnaround time, positivity and working with you have been possibly the best part of my PhD journey.

Thank you to George Oikonomou for the lovely discussion, for calming me down during stressed-out periods, in general being there, and for providing helpful feedback on articles.

The PhD journey would not have been possible without the trust and belief of super-kind and calm Theo Tryfonas. Thank you for providing me with an opportunity to work on the PhD. Your support and guidance in finalising the thesis and navigating throughout PhD has been excellent.

Thank you to Nektarios Georgalas (Aris) for the support and feedback on the work and my Industrial Sponsor, British Telecom (BT). The PhD was only possible with BT's financial support. Your funds enabled us to buy hardware for this PhD and helped me to implement the concepts in the real world. Thank you!

In my life, I never thought that an average student like me would come so far and do a PhD. Someway or another, the credit goes to Ragini and Divya because of whom I came to do the postgraduate and someway PhD too. Thank you!

Thank you to Priyanka Badva for cooking delicious food during my PhD, for her love and constant support and being there in good and bad times.

Lastly, my parents and my sister, who probably never thought I would do a PhD or come so far. Thank you for your patience, belief, support and love for all these years.

Thank you, everyone! It has been an honour for me to work with you and have you in my life.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: .....VIJAY KUMAR..... DATE: .....27-FEB-2023.....

TABLE OF CONTENTS

List of Tables

List of Figures

LIST OF PUBLICATIONS[1]

Papers (Included)

Chapter 2

Vijay Kumar, Sam Gunner, Maria Pregnolato, Patrick Tully, George Oikonomou, and Theo Tryfonas. Sensing and the City: From grassroots IoT sensors and city-fostered Open Data to Urban Observatories . *IET Smart Cities*, 2023.

The first author wrote the abovementioned paper and proposed the ideas/approaches, design, and experiments. The other authors provided their valuable reviews and suggestions to improve the paper. Sam Gunner and Patrick Tully guided the § 2.3.5 and § 2.4.3 respectively.

Chapter 3

Vijay Kumar, Sam Gunner, Theodoros Spyridopoulos, Antonis Vafeas, James Pope, Poonam Yadav, George Oikonomou, and Theo Tryfonas.
Challenges in the Design and Implementation of IoT Testbeds in Smart-Cities: A Systematic Review (**Under Review**).
IEEE Communications Surveys & Tutorials, 2023.

The first author wrote the abovementioned paper and proposed the ideas/approaches, design. The other authors provided their valuable reviews and suggestions to improve the paper. Sam Gunner and Theodoros Spyridopoulos guided the § 3.2.4. Antonis Vafeas guided the Fig. 3.5, Fig. 3.6 and Fig. 3.4.

Chapter 4

Vijay Kumar, George Oikonomou, and Theo Tryfonas.
An urban sensing architecture as essential infrastructure for future cities.
In *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing Companion*, UCC '21, New York, NY, USA, 2022. Association for Computing Machinery.

Vijay Kumar, James Pope, Poonam Yadav, Theodoros Spyridopoulos, George Oikonomou, and Theo Tryfonas.

[1]The status will change based on the acceptance or rejection in coming weeks/months.

SMARF: A Smart-City Research Project Framework for Sharing IoT Infrastructure among Research Projects from different Organisations (**Under Review**).
IEEE Communications Surveys & Tutorials, 2023.

The first author wrote the abovementioned paper and proposed and implemented the ideas/approaches, designs, and experiments. The other authors provided their valuable reviews and suggestions to improve the paper.

Chapter 5

Vijay Kumar, Poonam Yadav, and Leandro Soares Indrusiak.
Poster: Building iot resilient edge using lpwan and wifi.
In *ACM IMC*, 2021.

Vijay Kumar, Poonam Yadav, and Leandro Soares 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, 2022.

The first author wrote the abovementioned paper and proposed and implemented the ideas/approaches, designs, and experiments. The other authors provided their valuable reviews and suggestions to improve the paper. Poonam Yadav and Leandro Soares Indrusiak guided the § 5.3 and § 5.4. They designed the system model and the ILP formulation.

Papers (Not included)

Katharina Burger, Theo Tryfonas, Vijay Kumar, James Thomas, and Ute Leonards.
Revealing experiential plurality with citizen-sensed data: toward a socially-just 'shared space' (**Under Review**).
Science, Technology, & Human Values, 2023.

Ufuk Erol, Francesco Raimondo, James Pope, Samuel Gunner, Vijay Kumar, Ioannis Mavromatis, Pietro Carnelli, Theodoros Spyridopoulos, Aftab Khan, and George Oikonomou.
Multi-sensor, multi-device environmental indoor dataset of a smart building.
Data in Brief, 46:108775, 2023.

Ioannis Mavromatis, Adrian Sanchez-Mompo, Francesco Raimondo, James Pope, Marcello Bullo, Ingram Weeks, Vijay Kumar, Pietro Carnelli, George Oikonomou, Theodoros Spyridopoulos, and Aftab Khan.
Le3d: A lightweight ensemble framework of data drift detectors for resource-constrained devices.
September 2022.
IEEE Consumer Communications & Networking Conference, CCNC 2023 ; Conference date: 08-01-2023 Through 11-01-2023.

Chetankumar Mistry, Bogdan Stelea, Vijay Kumar, and Thomas Pasquier.
Demonstrating the practicality of unikernels to build a serverless platform at the edge.

In *2020 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 25–32, 2020.

James Pope, Francesco Raimondo, Vijay Kumar, Ryan McConville, Rob Piechocki, George Oikonomou, Thomas Pasquier, Bo Luo, Dan Howarth, Ioannis Mavromatis, Pietro Carnelli, Adrian Sanchez-Mompo, Theodoros Spyridopoulos, and Aftab Khan.
Container escape detection for edge devices.
In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, SenSys '21, page 532–536, New York, NY, USA, 2021. Association for Computing Machinery.

Ges Rosenberg, Neil Carhart, Helen Manchester, Lyndsay Grant, Eli Hatleskog, Elisa Coraggio, Vijay Kumar, and David Nepomuceno.
Bristol City's Future and the Role of Digital: Workshop Report.
2019.

CHAPTER 1

INTRODUCTION

Urbanization brings several issues, such as overcrowding, traffic congestion, air and water pollution, and shortage of resources. The concept of intelligent cities aims to address these challenges and improve the quality of life in cities. Smart cities use technology and data to manage assets and resources effectively, reduce waste and carbon footprints, and provide citizens with better services. By using IoT devices, AI and data analytics, smart cities aim to create sustainable, efficient, and livable urban environments. The thesis aims to research smart cities further and bring intelligent city milestones closer to reality. In this chapter, we introduce the research motivation, contributions, and the thesis outline.

1.1 Research Motivation

Urbanisation, human immigration, and climate change are changing the world as well as the resources available to humankind. There are different smart city initiatives that aim to solve the above challenges by collecting data and conducting research. Research organisations also deploy various urban research deployments to determine the next wave of innovations and support the development of essential infrastructure services to adapt to an ongoing climate change and technological landscape. Such infrastructure aims to bridge the gap between academic research, data analysis and strategic infrastructure planning [5]. This urban infrastructure enables the research community to perform research, analysis, generate new insights, and provide impactful results. Additionally, it must be efficient, reliable, resilient, and affordable. In cities, multiple challenges are faced repeatedly by the same or different groups that deploy devices to collect urban data. The infrastructure requires the management of devices ensuring security, connectivity, dipped battery and other relevant issues. So, there is a need for research to understand the challenges and how to deploy a testbed and infrastructure in a resilient, secure way that would

respect privacy and other essential parameters, thereby being able to develop functional data and serve its purpose.

The thesis explores the different data collected by citizens and city councils and its derived benefits that can solve the challenges related to urbanisation and climate change. It also explores the challenges faced in deploying infrastructure to collect urban data. It provides a smart city framework to solve software requirements and challenges in the deployment of urban infrastructure. Then it focuses on improving one particular challenge of edge device network resilience.

1.2 Contributions

This thesis makes four separate contributions to the field of smart city research. We choose to investigate these four aspects, as they play an essential role in the success of smart cities and in solving urban challenges related to carbon neutrality and net zero [6–9].

Understanding data collection and its derived benefits[1]

Urbanisation, population growth, ageing, and international migration result in increased demand for natural resources (water and energy), pollution, and environmental impacts. In confronting urban challenges, cities embrace technology to expand their services and enable citizens to share the benefits. In this context, an open research problem is understanding the data that can be collected and its respective benefits that can be derived [6, 10]. This impacts how various stakeholders use the data and may need to apply further analysis and techniques to make results impactful, leading to a change in behaviour. Therefore, a wrong answer to the question "What analysis can we do with the data collected" can make a difference between average and impactful results.

It has long been known that data and its analysis with impactful and clear visualisations help humans understand data [11], can lead to a change in behaviours and result in efficient use of resources and reduction of pollution and environmental impacts [12]. However, it often needs to clarify what data can be collected using IoT and digitisation and the impact it can make. Further, the data collected must be available for researchers for analysis. The collected data can be repurposed in addition to the infrastructure. For example, if we are collecting a particular kind of data and there are multiple data sources (subject to anonymization), they can be potentially correlated. For example, looking at the number of people on the street, we can understand about events going on or sudden events such protests. Another example would be the Automatic Plate Number Recognition (APNR) data can repurposed to understand how citizens are moving to electric vehicles and supporting net-zero initiatives. For such analysis to

[1]This work has been submitted to IET Smart Cities entitled “Citizen-to-City Sensing and the Role of Emerging Sociotechnical Infrastructures”.

happen, the data collected from the research projects or by city council can be made open-source or follow open-data initiatives [13] and upload data on Open Data Portals [14, 15] for other to use. Research funding bodies can make a provision requesting research projects to follow open-data initiatives.

In our first contribution, we measure and understand IoT and digitalisation at different levels (personal, building, district and urban) as used by the citizens and city council. The challenge here is to understand and document the different ways the data can be used and analysed by other stakeholders and how the data generated at one level is interconnected at another level. Additionally, data analysis can only happen when the data is collected and available to analyse securely and safely. We also briefly document the challenges faced in the collection of data.

To accomplish this task, we first examine priorities from a citizen's perspective, such as personal health, mental well-being, home, office, and community. We then explore the various services provided to the citizen (e.g., environmental data, recycling services) and their dependence on Internet of Things (IoT) derived information. We review innovative city digitalisation strategies and analyse the steps taken by various city councils to improve city services and become carbon neutral and net zero. Lastly, we review research papers from collaborative projects, and other Urban Observatory (UO)s that explore how citizens, businesses, and city authorities operate jointly to develop digital solutions to urban challenges. In this work, we have established answers to the following research questions:

RQ2.1 How does the use of IoT and the digitalisation process affect both citizens and local authorities, particularly the government, within the remit of smart cities and IoT?

RQ2.2 What are the key parameters (activities, entities, benefits derived) in the urban context and its governance, e.g., from well-being monitoring to environmental pollution levels and rubbish collection?

RQ2.3 What could be the initiatives that enable the curation of actionable data and help better understand the different phenomena (such as environmental and climate change)?

The current contribution helps us understand "what parameters are essential to be collected"; in the following contribution, we explore "what stops people from collecting the above data?".

Understanding the challenges in data collection[2]

In association with researchers and universities, city councils participate in multiple intelligent city research projects that aim to improve energy use, mobility, human well-being and productivity, reduce energy footprint, and increase the resilience and sustainability of the city.

[2]This work has been submitted to IEEE Communications and Survey entitled "Challenges in the Design and Implementation of IoT Testbeds in Smart-Cities: A Systematic Review"

Bristol Infrastructure Collaboratory (BIC) participates in multiple smart city projects. As part of BIC, we realised that the development and management of urban monitoring systems pose many challenges in the project stages related to requirement analysis, system design, implementation, deployment and operational challenges. Research projects often spend considerable time setting up and configuring the infrastructure to collect the data resulting in delayed execution of the project and weakening the impact of the project. Further, a few projects also face unexpected challenges that have resulted in a project failing to fulfil its goals and resulting in project abandonment [16].

In this part of the thesis, to understand the challenges, we performed a systematic analysis of the challenges in developing urban monitoring IoT testbeds, relying on our experiences from relevant UO projects, innovative city projects, and the study of pertinent literature.

First, we conducted semi-structured interviews with system architects and the implementation team of European research projects on IoT platforms and testbeds for urban monitoring. The discussions focused on the challenges the practitioner faced during the development, implementation, and management of IoT infrastructure and testbeds. The author asked open-ended questions with a free-flowing approach by asking the interviewee questions, and the conversation continued based on the answers. The author captured additional challenges based on their reflections on their experiences as members of smart-cities projects. In addition, we thoroughly review the relevant literature on infrastructure deployment.

To ensure that our understanding of the challenges can help future projects, we categorised the identified challenges based on the stage of the project lifecycle in which 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 projects and organise them under the V-model’s [4, 17] level to formalise the development process and provide a reference guide for future projects. V-model is a systems development process model that presents different stages of a technology development project. In this work, we have established answers to the following research questions:

RQ3.1 What are the challenges that a smart city research project faces in deploying IoT infrastructure that collects urban data?

RQ3.2 Can we classify the above challenges in different phases of research projects to help future smart city projects?

The current contribution identifies the challenges; in the following contribution, we provide an architecture to help solve some of the technical challenges.

Resolving the IoT infrastructure challenges[3]

Many smart cities research project (SCRP) involves the deployment of the IoT infrastructure in three tiers, including endpoints (sensors that measure the physical environment), edge gateway (collect and process data from endpoints), and cloud (collect and process data from endpoints/edge) [18–20]. On the other hand, other organisations also deploy similar three-tier (Cloud-Edge-Endpoint) or two-tier (Cloud-Edge) architectures in the city. For example, in collaboration with an advertising company, British Telecom (BT) has installed free Wi-Fi InLinkUK kiosks in the public area. Transport (bus companies) display estimated bus arrivals at bus stops using a bus display system. Managing the IoT infrastructure to support the smart city research project or city infrastructure takes time and effort. We present the challenges in the previous contribution. Although each project implements the IoT infrastructure differently (depending on the project requirements, usability, budget, time, and technical skillset), the IoT infrastructure deployed is often quite similar. There is a similarity in the required services, such as the deployment of applications, data storage, analysis, and visualisation.

Suppose there was a way to share the infrastructure on the cloud and edge tier between multiple SCRP, organisations, and local communities. In that case, new services could be created to speed up the implementation time for SCRP and local community projects. Sharing infrastructure can help reduce deployment costs and allows multiple parties (private organisations, community support groups, and individual citizens) to deploy their solutions and resolve public issues. However, it also adds challenges in data ownership, data management, ownership of devices, and access to the data. It also creates cyber security issues related to infrastructure and management, requiring coordination and collaboration with different people.

For our third contribution, we focus on solving a few challenges solvable by technology and those requirements that are common between different research projects. First, we understand the software requirements of a smart city platform/research project and combine them with the challenges faced in implementing innovative city projects. Second, we propose a smart city framework that resolves both the requirements and the challenges solvable and implementable by technology using open-source components. Then, we provide information about sharing the IoT infrastructure between multiple research projects and organisations. However, before the infrastructure can be shared, it needs to be secure and reliable and resolve the challenges of maintaining a single non-shared infrastructure. We deployed a small-scale infrastructure testbed to test our framework and support a small project gathering air quality data from the endpoints deployed in citizen's houses that is analysed by data scientists and visualised for the citizens. In this work, we have established answers to the following research questions:

RQ4.1 Given the challenges explored in chapter 3, what could be a solution that can solve the smart city software requirements and the challenges faced in a smart city research project?

[3]This work is submitted to IEEE Communications Surveys & Tutorials entitled "SMARF: A Smart-City Research Project Framework for Sharing IoT Infrastructure among Research Projects from different Organisations"

RQ4.2 Can we share the above smart city infrastructure between multiple organisations and smart city research projects to use the resources and reduce costs efficiently?

RQ4.3 What are the other solutions that exist, and how are they compared to our work?

The current contribution provided an intelligent city framework; The framework contains an edge tier, and network connectivity is one of the other challenges, so we provide a way to address that in the following contribution and improve the network resiliency of the edge device.

Improving the network resiliency at the edge[4]

For the fourth and final contribution, we work on improving the network resiliency of the edge device. The edge device (gateway) could run different urban applications ranging from air pollution monitoring to video analytics and safety-critical applications. Many safety-critical IoT applications, such as self-health monitoring through wearable IoT devices, connect to an edge device via Bluetooth, ZigBee or Wi-Fi and send the data to a cloud service through the Internet.

The challenge here is to ensure resilient network connectivity and reduce down-time. In the event of a network failure, e.g., a power outage or any other incidental connection failure, the network connectivity of the edge device could be disconnected temporarily, resulting in either data loss or delayed data communication. However, for safety-critical applications, it is essential to maintain resilient data connectivity at all times to deliver a time-critical message.

When the edge device has multi-network connectivity, such as low-power wide-area network (LPWAN), application traffic can be routed through a specific network medium based on the application requirements and available network medium. Further, in case of a particular network medium unavailability or failure, the application can be informed of the network state, decide on the network's suitability, and adapt accordingly. For instance, assuming the application is sending data over Wi-Fi and because of power failure Wi-Fi is disconnected, the application can choose to send data over Long-Term Evolution (LTE)(LTE for Machines (LTE-M)/NarrowBand-IoT (NB-IoT)), LoRa (Long Range), Sigfox and adapt parameters such as payload size and frequency accordingly.

We explore if we can achieve network resiliency at the Edge using LPWAN and Wi-Fi for time-critical IoT applications (data-flow simulated using surveyed real applications). We present the use cases for resiliency requirements of the IoT edge networks, provide a detailed analysis of many state-of-the-art LPWAN technologies, and evaluate their bandwidth, latency, throughput and maximum packet size using an experiment. Then, we identify and compare resource management approaches that consider Quality of Service (QoS) requirements at multiple levels of criticality and define an adaptive system *Resilient Edge* to meet the application resiliency requirements using

[4]We explore this subject in the research paper "*Resilient Edge*: Building an adaptive and resilient multi-communication network for IoT Edge using LPWAN and WiFi" published in IEEE Transactions on Network and Service Management

underlying LPWAN technologies. We also provide an open-source implementation of *Resilient Edge* and detailed insights considering hardware and network limitations. In this work, we have established answers to the following research questions:

RQ5.1 What are the different resiliency requirements for different applications using shared IoT edge networks and understand and evaluate the state-of-the-art LPWAN technologies in terms of their bandwidth, latency, throughput and maximum packet size?

RQ5.2 Can we identify and compare resource management approaches that consider QoS requirements at multiple levels of criticality?

RQ5.3 Can we define an adaptive system to meet application resiliency requirements using low power, energy-efficient networks such as LPWAN technologies; also provide an open-source implementation of *Resilient Edge* and detailed insights considering hardware and network limitations?. The system represents the software running on the edge responsible for making the communication more resilient between the applications running on the edge and the cloud by adapting the type of network protocol.

The thesis starts with high-level challenges faced by cities (urbanisation) in chapter 1. We move to applications (such as IoT and digitisation data collection) in chapter 2, then essentially to requirements (such as challenges in the implementation of such infrastructure) in chapter 3, then to system architecture (a way to implement the infrastructure) in chapter 4, and then to a much lower level communication technologies (improving network resilience) in chapter 5.

To summarise, in chapter 4 and chapter 5, we help to overcome some of the challenges identified in chapter 3. We hope to enable researchers and the city council to deliver the services in an efficient, reliable with reduced cost identified in chapter 2, and therefore tackle some of the large-scale issues of urbanisation highlighted at the beginning of chapter 1.

Summary of Contributions

Below is a summary of the contributions of our work.

C1 We present an list of IoT data collected at different levels (personal, building, district and urban), the benefits derived from them, how they are interconnected, and the different initiatives for solving urban data challenges and challenges because of Urbanisation.

C2 We present a systematic review of the challenges faced in designing, implementing, and deploying IoT infrastructure to collect urban data in smart cities research projects.

C3 We present a smart city framework with its implementation using open-source components that aim to solve the software requirements of smart cities and the challenges faced during the design, implementation and deployment. Additionally, we present how the implemented

Table 1.1
Mapping of publications to relevant chapters. The table contains the paper that are directly corresponding to RQs and a full list of publications produced during the period of research is included in the List of Publications.

Research output	**Research Questions**	**Chapter**	**Status**
Citizen-to-City Sensing and the Role of Emerging Sociotechnical Infrastructures	RQ 2.1, 2.2, 2.3	2	Submitted - Under review
Challenges in the Design and Implementation of IoT Testbeds in Smart-Cities: A Systematic Review	RQ 3.1, 3.2	3	Submitted - Under review
SMARF: A Smart-City Research Project Framework for Sharing IoT Infrastructure among Research Projects from different Organisations	RQ 4.1, 4.2, 4.3	4	Submitted - Under review
Resilient Edge: Building an adaptive and resilient multicommunication network for IoT Edge using LPWAN and WiFi	RQ 5.1,5.2,5.3	5	Published

IoT infrastructure can be shared among research projects and organisations to reduce cost and improve resource efficiency.

C4 We present a resilient edge system that improves the network resiliency at the edge tier using LPWAN connection for the critical applications running on the edge tier with defined criticality and QoS requirements.

1.3 Thesis Outline

The thesis is organised according to the following layout and summarised in Table 1.1.

Chapter 2 is based on the paper "Citizen-to-City Sensing and the Role of Emerging Sociotechnical Infrastructures" submitted to the IET Smart Cities journal and is under review. It illustrates the sensing capabilities in urban environments on the citizen-to-city scale and how sensing at different levels is interlinked.

Chapter 3 is based on the paper "Challenges in the Design and Implementation of IoT Testbeds in Smart-Cities: A Systematic Review" submitted to IEEE Communications and Survey journal and is under review. It presents a systematic study of the challenges of developing, deploying and managing the IoT infrastructure deployed in smart city research projects.

Chapter 4 presents the paper "SMARF: A Smart-City Research Project Framework for Sharing IoT Infrastructure among Research Projects from different Organisations" submitted to IEEE Communications Surveys and Tutorials journal and is currently being reviewed. It offers an implementable smart-city framework with an open-source, reproducible, 3-tier architecture. It explores the possibility of sharing the IoT infrastructure between multiple smart-cities research projects at the cloud and edge tier.

Chapter 5 is based on the paper "Resilient Edge: Building an adaptive and resilient multi-communication network for the IoT edge using LPWAN and Wi-Fi" published in IEEE Transactions on Network and Service Management (IEEE TNSM) journal. It systematically investigates

how to meet IoT applications' mixed-criticality QoS requirements in multi-communication networks.

Chapter 6 concludes the dissertation and provides possible future research directions.

CHAPTER 2

CITIZEN-TO-CITY SENSING AND THE ROLE OF EMERGING SOCIOTECHNICAL INFRASTRUCTURES[1]

The concept of smart cities has been present for decades; however, it has yet to reach its full potential. To create a foundation for a city to be intelligent, we must first understand the length and breadth of urban data that can be collected and analysed and the benefits that can be derived from it using data curation; otherwise, it is challenging to understand what can be achieved and advance state of the art. Therefore, the motivation of this chapter is to understand the sensing capabilities in the physical environment from a citizen scale to a city scale and how they are interlinked at various levels and provide information on the data that can be collected, analysed, curated and how it increases productivity and reduces costs for citizens and city councils, creating a foundation path for a smart city.

Urbanisation is one of the four demographic megatrends that reshape the human community in addition to population growth, ageing, and international migration [21]. As cities grow, the logistics required to ensure essential services become more challenging for city councils [21]. The challenges include increased demand for natural resources (water and energy), increased pollution, and environmental impacts. The challenges urban communities face today incorporate the accompanying challenges such as emerging markets, migration, an ageing population, persistent inequality, and ageing infrastructure [22]. In confronting urban challenges, cities embrace technology to expand their services and allow citizens to share the benefits. Emerging technology, e.g., IoT, provides the ability to understand the physical environment with granular data, allowing better decisions.

[1]This work has been submitted to IET Smart Cities entitled "Citizen-to-City Sensing and the Role of Emerging Sociotechnical Infrastructures. The first author wrote the abovementioned paper and proposed the ideas/approaches, design, and experiments. The other authors provided their valuable reviews and suggestions to improve the paper. Sam Gunner and Patrick Tully guided the § 2.3.5 and § 2.4.3 respectively.

The IoT is a growing system of billions of devices connected to the Internet and each other through most wireless networks. It has become increasingly relevant to society and is integral to the lives of citizens. Similarly, digitalisation exploits digital technologies to transform business processes and workflows to improve business models carving the path for digital transformation. We designed our research questions to understand the landscape and impact of IoT and digitalisation and how it can help citizens and city councils.

The chapter is organised as follows: § 2.1 provides the research questions and approach. § 2.2 provides a brief background. § 2.3 provides IoT capabilities at the personal, building, district, and urban levels and how the data is interlinked at all levels. § 2.4 presents an introduction to the urban data challenges, different initiatives, and the role of UO in approaching these challenges. § 2.5 concludes the chapter.

2.1 Research Questions and Approach

Research Questions

This chapter aims to understand how IoT and digitisation help citizens and city councils reduce costs and improve productivity and our knowledge of IoT use by answering the following research questions.

RQ2.1 How does the use of IoT and the digitalisation process affect both citizens and local authorities, particularly the government, within the remit of smart cities and IoT?

RQ2.2 What are the key parameters (activities, entities, benefits derived) in the urban context and its governance, e.g., from well-being monitoring to environmental pollution levels and rubbish collection?

RQ2.3 What could be the initiatives that enable the curation of actionable data and help better understand the different phenomena (such as environmental and climate change)?

Research Approach

To answer these research questions, the author first examines priorities from a citizen's perspective, such as personal health, mental well-being, home, office, and community. The author then explore the various services provided to the citizen (e.g., environmental data, recycling services) and their dependence on the information. Further, the author reviewed multiple smart cities' digitalisation strategies and analysed the steps taken by various city councils to improve city services. Lastly, the author reviews research papers from collaborative projects such as Organicity [23], CityVerve [24], MKSmart [25], REPLICATE [26] and other UOs to understand innovative real-life examples of IoT and digitalisation.

2.2 Background

IoT has been helping to reduce human workload using automation [27, 28], increasing efficiency [29], and improving Quality of Life (QoL) [30]. The city council measures the progress of the city using the data generated by the IoT devices such as Key Performance Indicators (KPI). KPI parameters can be air quality, the percentage of reused, recycled, or mixed household waste, and the proportion of energy-efficient homes. For example, Belfast and Bristol City Council (BCC) have released their 'One City Belfast' agenda [31] and the One City Plan [32] respectively, where they have identified indicators that measure city progress. Many city authorities have been active in this space, collecting and making available city information. For example, the City of Chicago was one of the pioneers in opening urban data sets for public use through its data portal [33] and embraced urban sensing through collaborations with researchers and innovators on projects such as Array of Things (AoT) [19, 34].

Other initiatives in cities are driven by non-government stakeholders, such as in the US, the University of Michigan Urban Collaboratory [35] provides an opportunity for interdisciplinary faculty and students to work with city stakeholders. Together, they identify community challenges and implement solutions using innovative city technology and novel urban design methods.

The reach of IoT is not limited to devices that urban dwellers can use to improve their daily lives but could be scaled up to the city government level to improve and optimise cities and allow all citizens to experience a high quality of life. The opportunities offered through IoT implementation are countless but simultaneously challenging because of the complexity of managing the data, and devices with different technologies, understanding citizen concerns and resolving them.

2.3 Generating Data at Multiple Scales: Sensors, IoT, and Big Data

The use of IoT and digitalisation produces data that can be logically viewed as falling within four levels based on the location of the generation, the use and how data collectors and users change at scale: personal, building, district and urban. The data generated at the different levels (personal, building, district and urban on the individual scale) are intrinsically linked, and data at one level can be used at a different level. The author review a wide range of IoT, from the personal to the urban level. The data is generated at different levels and different maturity. It can come from well-established applications that people have used for years or pilot projects that may contribute to the established technology of the next ten years. It is important to highlight that the technologies mentioned (§ 2.3) can be aspirational or proof of concept, piloted on a small scale or proposed only in the laboratory environment; only a few technologies become successful based on financial funding and model. The author attempt to map the entire landscape to understand the potential for IoT and digitalisation.

Figure 2.1: Data collection and its benefits at personal level

A level classification is provided as a large amount of data is generated at each level. Understanding and differentiating between different levels of data generation (and essential parameters such as granularity and frequency) facilitates data curation. Such descriptions improve citizens' and city officials' understanding of what data is being collected, stored and increase transparency and trust in data collection schemes from an operational perspective. Technically, it simplifies data reuse by application developers, as sensor data feed creators are expected to describe the produced data harmoniously. The author present the data collection and use at personal (§ 2.3.1), built (§ 2.3.2), district (§ 2.3.3) and urban level (§ 2.3.4) and how it is intrinsically linked (§ 2.3.5).

2.3.1 Data Collection and its Benefits at Personal Level

The main concerns of a person are health, mental well-being, and life-threatening circumstances such as a stroke. Table 2.1 and Fig. 2.1 summarise the usage of IoT on a personal level. Wearable devices with sensors, such as a Fitbit or an Apple Watch [36], measure different physical activities such as step counts, calories burnt, sleep tracking, and floors climbed, which helps keep track of the individual's health. Wearable sensors also help identify sharp activity for stroke and Traumatic Brain Injury (TBI) [37] in patients. They have also been extensively used for the localisation and recognition of movements of day-to-day living [38, 39]. For people (kids or older)

Table 2.1
Data collection and its benefits at a personal level

Activity	**Source**	**Items Measured**	**Benefits derived**
Exercises/Sleeping	Smart watch/Fitness tracker	Heart rate, calories burnt, steps taken, floor climbed	Identify sharp activities for stroke and TBI. Self-monitoring of Health.
Activities in bathroom	Electrocardiogram, photoplethysmogram, ballistocardiogram	Heart rate, blood pressure	Health and well being of an individual
Sleeping	Pressure sensors in or under the bed/body-worn altimeter/tilt sensor	Heart rate, respiratory signal, sleep posture, movement activity and quality of sleep	To improve sleep analysis and detection of sleep problems; Detects and alerts when a person leaves the bed.
Sitting	Pressure or force sensors	Body posture while sitting on the chair or sofa	To avoid the adverse effect of poor sitting behaviour and posture that can be linked to pain and other complications.
Water intake	Water flow sensors	Hot and cold water usage	Monitoring water usage in kitchen/bathroom to determine if elderly people are drinking water regularly.
Mental well-being/Social interaction/Self journal	Mobile application	Occasions where people showed gratitude; social application (Facebook/WhatsApp) usage	Monitor mental and emotional well-being for vulnerable people. Depending on multiple parameters and the worrying situation, caregivers can be informed.

who may not prefer wearable devices, toilet seats [40] in the bathroom are equipped with sensors that can measure heart rate and blood pressure and keep track of the health and well-being of an individual. Sleeping beds have been fitted with sensors to monitor heart rate, respiratory signal, movement activity of sleep posture, and sleep quality, providing sleep analysis and detection of sleep-related problems [41]. Pressure or force sensors can be integrated into sofas or chairs to detect and avoid poor body postures that harm health and cause pain and other complications [42]. Hot and cold water flow is measured in the bathroom/kitchen to monitor the water used during a bath and usage of drinking water [43, 44], which helps to monitor the health of older people. Personal health applications provide individuals with information on their health and wellness and allow them to follow their medical providers' and personal health goals and care. The individual can use personal health applications to exercise regularly, enhance cardio fitness, move throughout the day, reach rest goals, track their menstrual cycle, or practise mindfulness with breathing workouts to remain calm and attentive. Applications [36] use sensors and artificial intelligence to help patients with post-traumatic stress disorder (PTSD) and trauma improve sleep quality or measure and record tremors and dyskinetic signs in patients with Parkinson's condition. Health applications help users understand their health and well-being. The health insights can motivate the citizen leading to a change in behaviours. In addition to walking, citizens also cycle or use e-bikes to travel and may want to know their cycling statistics.

In addition to "physical health measures", "mental well-being measures" are metrics for assessing the well-being of citizens. Mental health applications provide regular meditation

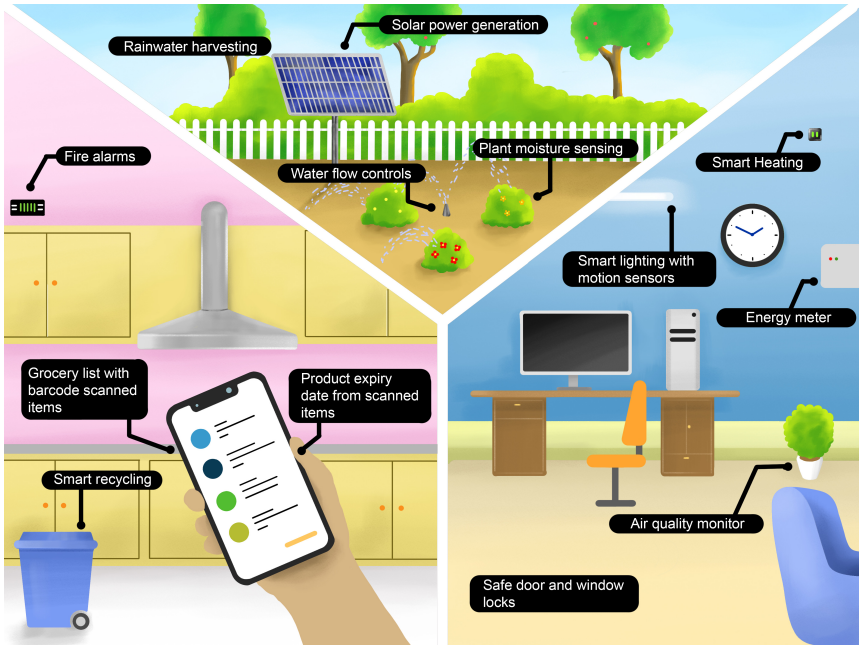

Figure 2.2: Data collection and its benefits at built environment level

routines, deep breathing exercises aided by visuals and haptic feedback, and request a moment to halt and contemplate a thought or an action. Citizens can use applications to remind them to breathe or reflect throughout the day. The user also uses applications to concentrate on a specific activity by turning on digital wellness applications and having focus time for career, private time, rest, wellness, mindfulness, gaming, reading or driving. Citizens can also decrease device distractions by permitting only notifications from individuals and applications they prefer [36]. Some applications enable users to feel grateful or thanked, help feeling good, and stimulate positive feelings about life. Mobile applications promoting gratitude [45] have been created to foster gratitude and assess social connectedness. Community initiatives such as Action for Happiness [46] (a movement of people committed to building a happier and more caring society) help support mental health.

2.3.2 Data Collection and its Benefits at Built Environment Level

People spend approximately 15.7 hours per day [47] at home and the rest at work or outside. The number of hours at home has increased further during the pandemic (2020/21) when people visit outside only for essential services. Homes must be safe and healthy places to live, and the IoT can ensure this. Data can be collected by the citizen or the agency that maintains the home/office. Table 2.2 and Fig. 2.2 summarise the usage of IoT at the building level. At a building level, we examine the use of IoT from the perspective of efficiency, security, environment, and usability outside the home.

Efficiency Perspective

From the efficiency perspective, citizens can optimise the lighting and heating around homes/offices. A citizen can use a motion sensor to turn on and off lights based on the presence of humans, saving energy and reducing operational costs [48]. They can use thermostats to automatically set room temperature based on room occupancy to save energy. MIT Senseable Team created a system "Local Warming" [49], which directs the heat to where people are present rather than heating the whole room, resulting in energy savings. Humidity sensors in the bathroom can automatically turn on the exhaust fan. Citizens can monitor the energy usage of the home/office space with smart plugs (connected to household appliances such as kettles and televisions) or electric metres attached to the main inlet for electricity. Electricity companies have begun to install smart metres that allow citizens to monitor their energy consumption half-hourly, daily, monthly and yearly. The electricity usage data help to understand and break down the electricity charges for the user, understand what the appliance uses most of the energy, and how to reduce consumption/electric charges. 3e-HOUSES [50] and Twinergy [51] implemented IoT technologies in social housing and provided an innovative set of services for energy efficiency. Such projects incorporate digital intelligence to allow citizens to actively adapt their consumption to market fluctuations with the help of data and automation. IoT technologies provide real-time monitoring and management of energy consumption, integrate renewable energy, and create resources to reduce energy consumption. Such projects and IoT technologies enable citizens to change their consumption behaviour, reduce costs, and improve QoL. Citizens also use e-bikes and Electric Vehicle (EV) to travel and charge them regularly; it is beneficial for energy companies to know the user requirements for charging depending on a vehicle type.

Researchers can further correlate information on Indoor Air Quality (IAQ), energy monitoring, and building materials used in the home/office to understand and determine the efficiency of building materials used [52].

In the kitchen, IoT devices are handy for creating a grocery list, recycling, tracking expiration dates of perishable items, and detecting smoke. Gas/smoke detectors are a legal requirement in the UK and, therefore, an integral part of a home that can detect Carbon Monoxide (CO) or natural gas and help prevent a malfunction in the event of a leaky or forgotten stove and save lives. Smart bins can quickly scan item barcodes, create a grocery list [53] for instant order and inform the user of the recyclability of the product [54] for proper disposal. Phone applications [55, 56] keep track of expiry dates to prioritise the use of ingredients and help reduce food waste.

Security Perspective

Citizens can monitor doors and windows (open or closed) using IoT sensors, helping regulate temperature. The number of occupants in the buildings can be determined using mobile phones and Wi-Fi signals [57, 58]. Depending on the occupant, the room door can be opened or closed.

This helps during guest or professional visits, e.g., opening the door remotely to a plumber who visits to fix a pipe. IoT devices can monitor babies or the elderly in the family or if someone is sick. Video-based systems (depth-based cameras) are used to analyse functional movements of a child or a patient [59], alert systems to detect dangerous events (e.g., a person about to fall [60], walking movements (abnormal gait pattern [43]), and identify specific types of dementia [61]. Parking sensors are installed at home to determine whether the car has been parked correctly. Additionally, it helps detect the vehicle (e.g., the car moved from its position during an odd hour). It also helps to detect recent key-relay attacks [62] on vehicles such as Tesla. In addition, innovative home products such as Amazon Alexa and Google Home provide multiple functions to protect the home using indoor and outdoor cameras, smart doors, and efficient heating and lighting using motion sensors and learning user preferences.

Indoor Environment

From an environmental point of view, citizens can use IoT devices (such as Smart Citizen Kit (SCK) [63], Luftdaten [64], Atmotube [65]) to measure environmental parameters (IAQ) around the house and install air filters to purify the air. Bespoke environmental sensors developed by research projects, such as the SPHERE environmental sensor [18], are also used to monitor IAQ. Bad IAQ irritate the eyes, nose and throat, causing headaches, dizziness, fatigue (immediate short-term effects) or respiratory and heart diseases, and cancer (long-term effects) [66]. There are multiple indoor air pollutants such as asbestos, biological contaminants, CO, lead, nitrogen dioxide, pesticides, radon, indoor particulate matter, second-hand smoke/environmental tobacco smoke, formaldehyde/pressed wood products, heaters, stoves, fireplaces and chimneys, VOCS [67]. Measuring IAQ over time helps to collect data and answer questions such as the effect of burning candles or cooking on IAQ; opening windows/doors for cross ventilation to improve IAQ. With growing awareness of air pollution worldwide, air filters and better indoor quality are a top priority for most urban cities. In addition to IAQ, citizens can measure indoor environmental conditions such as temperature, humidity, barometric pressure, and luminosity. The data help answer questions on the impact of indoor environmental conditions on the room occupants' health; the impact of humidity causing the room's walls' dampness in the house. SPHERE [68] developed home sensors to analyse and assist in managing health and well-being conditions, allowing early diagnosis, lifestyle shifts, and the suitability of patients to live at the residence.

Usability outside the home

Outdoors, IoT devices can benefit solar power generation, moisture detection, parking sensors and rainwater harvesting. Citizens can generate renewable energy using solar panels or smart-flower [69] to generate electricity and reduce dependence on brown energy [70, 71]. Citizens can schedule the usage of washing machines and dryers, charging EV to use renewable energy (during the day when solar energy is generated) [72]. Furthermore, citizens can sell surplus

Table 2.2
Data collection and its benefits at a built environmental level - home/office

Entity	Activity	Sources	Items Measured	Benefits derived
Room (Living/ Dining/ Bedroom); Office space	Lighting	Motion sensors	Human presence	To switch lights on and off to save energy costs.
	Heating	Motion sensors	Human presence	To set the heating based on the presence of people and direct the heat to people rather than heating the entire room.
	Safety check	Door and window sensors	Doors/windows (open/closed status)	To ensure doors/windows are closed for safety and heating/cooling efficiency.
	Movement monitoring	Video-based system/depth-based cameras	Human functional movements	To detect dangerous events such as a person falling or abnormal gait patterns.
	IAQ monitoring	Air pollution sensors	Air pollutants such as asbestos, biological pollutants, CO, CO2, sulphur oxides, Volatile Organic Compounds (VOCS)	To understand the cause and ease irritation of the eyes, nose and throat, headaches, dizziness and tiredness or respiratory conditions, heart disease and cancer.
	Energy monitoring	Smart plugs/Energy monitors	Energy usage	To determine energy usage by different appliances and co-relate based on the occupancy of the building space and determine the efficiency of the building materials.
Kitchen	Grocery list	Bar code scanner	Smart bins scan items bar code	To automatically create a grocery list, sync it to the phone application and order automatically.
	Recycling	Bar code scanner	Smart bins scan items bar code	To inform the user whether the product could be recycled or not and how it can be recycled.
	Product expiry date	Bar code scanner	Date of manufacture, best use before and expiry date	To track expiry date, optimise usage of ingredients and help reduce food waste.
	Gas/smoke monitors	Natural gas and CO monitor	Levels of natural gas and CO in the air	To detect gas buildup from a leaky or forgotten stove and avoid accidents.
Other Area (Garden/ parking/ utility)	Solar power generation	Solar panels/smart flower panels	Energy generated, usable and extra energy	Scheduling charging of EV, washing machines or dryers to utilise renewable energy and send the extra energy generated back to the electric grid.
	Plant moisture sensing	Moisture sensors	Moisture level	Identifying when the plant needs watering based on the moisture levels, the gardens and plants can be watered automatically using stored rainwater.
	Parking information	Parking sensors	Proper or improper car parking	Identify whether the car has been parked correctly. It can also help detect theft (modern key-relay attacks).
	Water flows	Temperature/ volumetric sensors	Temperature and usage of water	Optimise hot/cold water usage and reduce water wastage.
	Rainwater harvesting	Rain sensor (pluviometer)	amount of rain	To harvest rainwater for flushing the toilet, washing clothes and gardening watering; also reduce the probability of urban flooding by managing stormwater discharge during heavy rainfall.

Figure 2.3: Data collection and its benefits at district level

electricity generated back to the community (electric grid) using feed-in tariffs [73]. Water supply interruptions cause significant problems for customers [74]. IoT helps the practice of rainwater harvesting [75] to flush the toilet, wash clothes, and water the garden by storing the rainwater in underground tanks. The practice of saving rainwater improves resilience in two ways: first, it helps reduce the amount of mains water required; second, it can be used when the regular water supply is interrupted [76]. It also helps to reduce stormwater discharge from heavy rainfall, reducing the probability of urban flooding and pollution [77]. IoT also plays an essential role in maintaining gardens by sensing the moisture in plants/gardens to determine when plants need water and can be automatically watered using stored rainwater.

2.3.3 Data Collection and its Benefits at District Level

The district level includes a neighbourhood and the surrounding area where a citizen resides. The author examine the use of IoT/digitalisation from a district level's mobility, liveability, and community perspective. Most of the district-level parameters are collected by the government and are available to citizens for use. Table 2.3 and Fig. 2.3 summarise the usage of IoT at the district level.

Mobility

Mobility and accessibility are significant issues in urban life. Mobility refers to the physical movement measured by trips, distance, and travel time. In contrast, accessibility refers to the ability to reach preferred goods, services, activities, and destinations [78]. In addition to distance

Table 2.3
Data collection and its benefits at a district level - useful information for citizens in the neighbourhood and surrounding areas

Entity	Source	Items Measured	Benefits derived
Mobility information	Google maps/Bristol open data	Air quality, temperature, wind speed/direction, traffic delays Empty parking and charging points Cycling route, public bike pumps, cycle shops and repairs	To plan the journey and reroute traffic to manage pollution. Real-time parking spot availability near the citizen and EV charging points. Help citizens explore the city by using bicycles.
Neighbourhood information via Open data	Bristol open data/City council website	1. Trees/parks and green spaces 2. Air and surface water quality 3. Recycling banks 4. QoL such as to measure of inequality in health, lifestyle, community, local services, and public perception of living in the city. 5. Crime map 6. Children's school-specific data	1. To raise awareness of the environmental and monetary benefits of trees. Also, make available information about the different parks and green spaces available. 2. Determine the liveability of the neighbourhood 3. Provide information about recycling points in the city and what kind of waste is recycled 4. Access and improve QoL for citizens. 5. Provide details on the different types of crimes that occur in the neighbourhood and what actions the authorities take, including progress in different cases. 6. Access the quality of education in schools and improve it.
Community efforts through social networking websites	Mobile applications	1. Descriptions and addresses of local food outlets with discounts for locals 2. Items purchased, use, and waste 3. Rental resources shared between residents 4. Neighbourhood competitions 5. Skill set of the local population 6. People willing to share food with those in need. 7. Mental and physical health of residents 8. Lactating moms and breastfeeding hubs	1. Promote local food, reduce food waste, and promote relationships and goodwill. 2. To reduce household waste, provide reusable containers to reduce the waste of non-recyclable packaging. 3. To allow people in the neighbourhood to request and share items to facilitate efficient use of resources. 4. Organise events such as singing and running to create more gracious and happening hoods. 5. If someone needs to learn a new skill, they can find someone in the neighbourhood who can teach that skill set. 6. Match older people who need food with those who want to share their meals or share excess food voluntarily. 7. Reduce loneliness by organising social gatherings for those who benefit from it 8. To promote breastfeeding within the neighbourhood with a map detailing the best locations, provide information about why you should breastfeed, how to breastfeed, and find help in the local area.

and travel directions, knowing parameters such as environment (air quality/temperature, wind speed/direction, traffic delays), location of EV charging points, car parking spots, and availability would be beneficial from the traveller's perspective. For cyclists, information such as proposed cycle routes, cycling leisure/strategic routes, cycle shop/repairs, and public bike pumps can help navigate the city using bicycles. From an accessibility perspective, knowing the options available for disabled parking and disabled access to the building and stores is also beneficial. Cycle routes used by the citizens can help transport planners to plan cycle routes and help health officials understand the long-term impacts of cycling on citizens.

Liveability

House prices depend on QoL in the area and near essential services (metro rail/bus station/ hospitals/schools/ shopping malls); IoT and digitalisation help narrow the options for the potential home buyer. Digitalisation helps owners predict the rise/fall of house prices based on future

construction in the neighbourhood. The safety, leisure and accessibility of essential services are crucial to a citizen. Regarding safety, crime maps provide detailed information on crime rates and various types of crimes. From a health perspective, historical and real-time air quality data and surface water quality help citizens understand the liveability of the neighbourhood. Information regarding schools, such as the number of students attending the classes and the student's performance, can help parents choose a suitable school for their children. Knowing the location of parks and green spaces is essential for relaxation purposes. Trees Near You [79] provides information about different species of trees and their environmental benefits and allows citizens to connect with nature. Information such as recycling bank locations and the type of waste recycled is useful. In addition, comprehensive information on QoL indicates the happiness of citizens. It is measured by multiple indicators, such as inequality in health, lifestyle, community, local services, and public perception of living. QoL indicators are obtained from an annual survey of neighbourhood residents. Data of this nature is commonly made available by the local authority, open data platforms by the government or the city council (e.g., Open Data Bristol [80] or London Data Store [81]).

Community

A neighbourhood often contains citizens of all age groups. Sharing resources becomes essential when the focus moves from a single individual to the community. People can share the vehicle and ride with a stranger by carpooling, borrowing a power tool, and more, all with the click of a button. A good example of sharing resources is Peerby [82], sharing knowledge is Konnektid [83], sharing food is Shareyourmeal [84], Olio [85] and 'Too Good To Go' [86]. Helpfulpeeps [87] is a social marketplace for local help and allows people to ask for help and offer help. The author saved 365£ using Olio and participated in sharing food ingredients (sandwiches, staples) and non-food items (lamps, monitors, television, plants) with 36 people over two years. Similarly, the author saved 88£, 30kg of CO2 impact by using the 'Too Good To Go app' and buying food at a reduced price.

Restaurant owners can use a mobile application to promote local food within the community, containing descriptions and addresses of local food outlets with discounts for locals, followed by a local food fair. Customers can be credited with a stamp (food outlet logo) to promote the relationship and the goodwill factor (e.g., Milton Keynes organised the MK Food Revolution [88] to promote local food). To reduce household plastic waste, shops can allow residents to buy dry bulk goods such as pasta, rice, flour, and beans in reusable containers, thereby reducing non-recyclable packaging waste (e.g., MKSmart launched refill shops with zero food waste agenda). Communities also organise neighbourhood competitions to organise events or healthy competitions (such as singing and running). HoodChampions [89] promotes citizens to contribute to their neighbourhood, help each other, and create more welcoming and happening neighbourhoods in Singapore. Strava [90] tracks human exercise, mainly cycling and running, using GPS data and

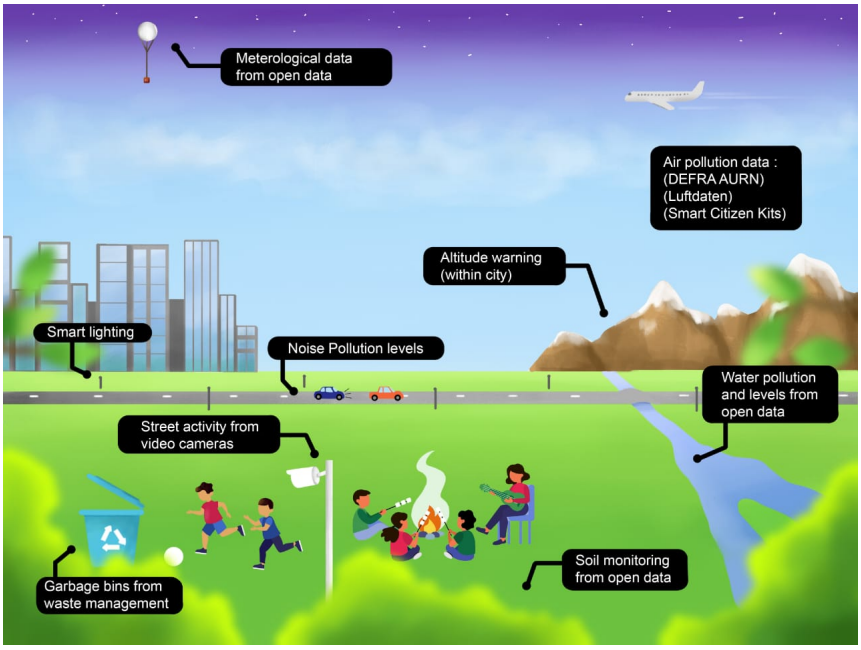

Figure 2.4: Data collection and its benefits at urban level

incorporating features of social networks.

Promoting wellness can be performed regularly to improve citizens' mental and physical health. The initiative helps reduce the feeling of loneliness among people through social gatherings and challenges. Walkers Group Santander created a mobile application for residents to schedule sociable walking events [91] and Walk in the city (another mobile application) challenged seniors through the use of gamification [92]. Breastfeeding is also promoted within the neighbourhood, with a map detailing the best breastfeeding locations to encourage young mothers. The Breastfeeding Hub [93] created a mobile application to promote breastfeeding in Milton Keynes with information on why mothers should breastfeed, how to breastfeed, and details of who can help in the local area.

2.3.4 Data Collection and its Benefits at Urban Level

The government and city councils must ensure that the city is livable and provide details about the steps they take to create a better life for citizens. The parameters mentioned in this section are collected by the government for official purposes and are provided to the citizen through an open data platform or used for urban planning/to meet specific regulatory standards. For this, city councils monitor multiple parameters such as environmental, localised information, transport/mobility, and crowd-sourced data. Fig. 2.4 summarises the usage of IoT on a urban level. Table 2.4 and Table 2.5 summarise the use of IoT at the urban level (the environmental factors the government monitors and the citizen services it provides to its citizens, respectively).

Environmental Factors

The government collects information on air quality, temperature, humidity, and barometric pressure of the city and surrounding areas by installing high-quality (high-precision) sensors. For example, Automatic Urban and Rural Network (AURN) is the UK's largest automatic monitoring network that reports compliance with ambient air quality directives. City councils often measure air quality using sample diffusion tubes placed at monitoring locations and collected over a few months making limited observation points for effective policy-making around pollution risk mitigation. To circumvent that, IoT technologies with environmental sensors can send the reading periodically, reducing cost and increasing productivity for city councils. UMBRELLA project has deployed more than 200 nodes spread throughout the South Gloucestershire region [94] and a few nodes at Cardiff University [95] that contain air quality sensors. Similarly, Project Eclipse [96] installed 115 low-cost solar-powered urban environmental sensors connected using a cellular network in Chicago to observe pollution at acceptable spatial and temporal resolutions [96]. Furthermore, the government encourages citizens to install low-cost sensors (such as SCK and Luftdaten) and provide the data to the city council. Air quality (PM) data sensors are installed in public buses or bin lorries (a low-cost solution to collect data for the whole city regularly) [97]. City councils can monitor air pollution sources such as power plants, road transport, home heating, farming, and industrial operations. The air pollution data helps to understand its source (construction/wood burning/traffic conditions/fireworks on Diwali and New Year nights). Combining air pollution with meteorological data allows air quality predictions to be made, to understand how it is affected by different seasons (e.g., in winter, people might burn more wood to keep rooms warm; rain reduces air pollution), and to inform policy on what measures can be taken to reduce its impact. Measures such as motivating parents to choose more sustainable transport by presenting Air Quality Index (AQI) around schools during picking up and dropping children; diverting traffic temporarily [98] can improve air quality. The knowledge gained can provide inferences from the above data that can help predict air quality in the region and take action accordingly.

City councils also measure the quantitative (i.e., volume) and categorised data (i.e., the type of noise such as traffic, construction noise, and birds sound) in the city [99]. The noise map can serve as one of the QoL indicators and can influence citizen's health condition; helpful to citizens when planning to move to a new location to prefer an area with less noise pollution [99, 100]. In addition, noise maps help construction companies fine-grain and optimise the materials they use to reduce noise pollution. Sensitive microphone devices capture brief clips of real-time ultrasonic and audible noise from birds, bats and wildlife, traffic, and human activity. Data are used to analyse the effect of noise and pollution on the animal's [101] behaviour; determine the activities of the bat population around the meadows; the pattern of human activity during various seasons of the year; source of the noise generated (e.g., cars, trucks, people). Sound sensors are attached to smart street lamps in public spaces to detect specific sounds such as gunshots, car

alarms, screams, fighting, and sound levels in bars and cafes. This system helps law enforcement authorities detect and prevent incidents.

Apart from air and noise pollution, the city council can measure water quality data (e.g. pH, ammonia) and water flows in surface water bodies (e.g., water depth, speed). Water quality is essential in the water treatment plant (provided to city homes for drinking) and water bodies for water sports. Monitoring and managing water levels in ports/water bodies can save energy (pumping water to maintain water levels) or avoid flooding. Sensors are used to monitor soil moisture and vibrations to detect a dangerous pattern of land conditions to warn about the risk of floods, landslides, and other natural hazards. The city council also records meteorological data, such as the amount of rainfall and solar energy, to estimate the energy demand of households [102].

Other information, such as wind speed and direction, is measured in different parts of the city. Cyclists can use this information to alter their route and avoid severely windy roads, or bridge owners can preventively close bridges. It helps to plan where to place domestic wind turbines and predict the amount of energy generated. Wind speed and direction combined with the altitude of different city roads help pedestrians, cyclists, motorists, and disabled people better plan their trips.

City councils use sensors to determine street activities, such as the number of people (pedestrians/cyclists) crossing an intersection or using a park with cameras, a microphone, and edge computing [19]. The data provide insight to city authorities for improving citizens' services or taking appropriate action to manage crowds in the city centre (e.g., informing law enforcement if necessary). Thermal cameras are also installed near the harbour to detect people who may fall into the water [103].

The city council provides street lighting to promote urban security and make roads and pathways safer. The city council must resolve street lighting problems as soon as possible to avoid a potential accident. Streetlight teams run periodic manual checks by driving along stretches of road to check if street lights are showing normal behaviour, turning off and on when they are supposed to, which is costly and requires traffic disruption. UMBRELLA project monitors street lights using a camera and a machine learning model, allowing the streetlight team to monitor the street lights [104] remotely. Suppose that a street light needs to be fixed as intended. In that case, the system sends an alert to the city council streetlight team providing speedy resolution, increasing council productivity, and reducing costs. The city council provides better street bright lighting by moving towards Light Emitting Diode (LED) lighting, which saves electrical power. They can also further improve the savings by combining lighting with sensors to detect movement [105], so the lights may be dimmed when no one is in its vicinity. A city council can use smart bins [106], which can compress garbage and inform authorities when they are full. This system provides authorities with information to send their staff only when required.

Table 2.4
Data collection and its benefits at an urban level - environmental parameters collected by the government and analytics obtained

Entity	Source	Items Measured	Benefits derived
Air pollution	DEFRA AURN/ Luftdaten/ SCK	Air temperature (°C) Relative humidity (%rh) Ambient light (lux) Barometric pressure (kPa) Particulate matter (PM1/2.5/10) (ug/m3) Equivalent carbon dioxide (ppm) VOCS (ppb) Ozone (ppm) Nitrogen Dioxide (ppm)	To understand 1. The impact transport mode selection has on air quality, with specific attention given to the areas around schools. 2. What are the peaks of air pollution at schools/train stations/coach stations/airports at the time of arrival and departure of students/trains/coaches/aeroplanes respectively? 3. The impact of weather (e.g. rain) and different seasons on air pollution. 4. Whether the air pollution can be reduced by temporarily diverting the traffic around the area of concern. 5. Combine with wind velocity data to identify the impact of specific pollution sources. 6. Allows comparison between the pollution contributions of 'stop/start' and 'steady flow' traffic. 7. Which days/weeks are good and worst for air pollution? 8. Predict air quality in the region based on the above parameters.
Water pollution/Levels	Bristol open data	pH, Dissolved oxygen, temperature, turbidity, electrical conductivity, chlorophyll, Nitrate, Ammonia, Chloride, Rhodamine, Hydrocarbons and water levels, water speed in surface water bodies	1. To determine water quality and find ways to contain or improve the current situation. 2. To monitor and reduce flooding.
Soil monitoring	Cosmic-ray soil moisture monitoring network	Soil moisture, vibrations	To detect dangerous patterns in land conditions and warn about floods, landslides and avalanches.
Meteorological data	Bristol Open Data, Wunder-Ground, Met-Office	Solar energy, rainfall, wind speed/direction	To predict rainfall, hurricanes and solar energy generation capacity.
Altitudes (within the city)	Ordnance Survey, Google maps	The elevation of different roads in the town	1. To help pedestrians, cyclists, motorists and disabled people to plan trips accordingly. 2. Useful for predicting surface water flooding.
Noise pollution	SCK	Noise volume (dB) and type of noise	1. To utilise the noise map in determining life quality and its effect on citizens health. 2. To help construction companies fine grain and optimise the building materials to reduce noise more effectively. 3. To capture ultrasonic and audible sounds of wildlife (bats, birds), traffic, human activity in real-time to determine: a. How active is the bat population in the area? b. Does traffic noise change animal behaviour over a day? c. What is the pattern of human activity during different seasons of the year? d. What are the sources of the noise generated?
Sound patterns	Noise sensors	Specific sounds such as gunshots, car alarms, screams, fighting and sound levels in bars and cafes.	To help law enforcement authorities in detecting and preventing incidents.
Street activity	Video cameras	Number of people crossing an intersection, or utilising a park	1. To provide insights to city authorities on improving services or take appropriate actions to manage crowds. 2. CCTV is also used for law enforcement and traffic management.
Smart lighting	Street lighting organisations	Approaching or going pedestrian or traffic	To save energy by dimming lights when no one is around and brighten it when people/traffic is approaching it.
Rubbish bins	Waste management - city council	When the garbage bins are full and need to be replaced	To optimise the collection of garbage and provide city authorities in utilising the staff more efficiently.

Table 2.5
Data collection and its benefits at an urban level - parameters measured by the government to improve citizen services

Entity	Source	Items Measured	Benefits derived
Citizen statistics	Bristol open data	Monitoring localised information. Statistics on population growth, crime, marital status, religion and employment Monitoring parking spaces.	To : 1. Provide conversational interfaces to make urban data more accessible and engaging to citizens. 2. Provide the real-time location of public vehicles such as emergency vehicles (police, ambulances, fire), buses, public taxis, trains, waste bin trucks, etc., to address critical situations quickly. 3. To help locate the nearest parking space and pay online, reducing time in finding a parking spot and CO emissions, less congestion.
Citizen mobility	travel services operator	monitoring vehicles mobility People travelling patterns	To deduce the people travelling patterns via trains, buses, boats and taxi's to figure out the most crowded stations/routes and how to improve services for people, e.g. support faster movement or add more trains/buses. To help predict the impact on stations (if any planned/unplanned work comes up) for which customer communications and operational plans need to be implemented. To understand: 1. Which route or platforms do people use? 2. Do they take the first train or wait for a less crowded one? 3. Do people choose the fastest route or the most comfortable one? 4. How do customers move around sophisticated stations? 5. To track from which and what regions vehicles are entering and exiting to and from the city?
Citizen mobility	Open data	(Anonymous) Real-time mobile locations of the citizens	To understand: 1. How are the neighbourhoods used during the day? 2. How does the allocation of buses and taxicabs associate with the people density? 3. How are goods and services disseminated in the city? 4. How do diverse sociable crowds, such as tourists and residents, inhabit the city?
Feedback Data	Crowd-sourcing platforms	Crowd-sourcing data - citizen feedback	To allow citizens to provide information about the city and help the government provide better services. The citizens can: 1. Install sensor kits to measure air pollution, humidity, and temperature; locate and report potholes and upload data to crowd-sourced maps. 2. Report flooding/broken streetlights/potholes to the crowd-sourced maps. 3. Use the website to propose, debate, and vote on ideas for improving the city.

Localised Information

A country is usually divided into counties, and a county is divided into wards. City councils keep information on population growth, crime, marital status, religion, and employment. Eventually, they can provide an interactive platform for citizens that provides conversational interfaces (powered by chatbots), making urban/city data more accessible and engaging to citizens, replacing current visual-based interfaces [107]. The real-time location of public vehicles such as emergency vehicles (police, ambulances, fire), buses, public taxis, trains, and waste trucks is also recorded and provided to address any critical situation in the shortest time possible. Cities with city

operations centres can coordinate emergency city services such as hospitals (ambulance), fire, police, transport providers (bus, rail, airport), and mental health services in the community. They also monitor different CCTV cameras and traffic junctions in the city. Having a coordinated response between various agencies, they would respond quickly and prevent emergencies [108].

Transport - Mobility

City councils and governments can collect data on the purchase of cars, trains, buses and tickets to determine the mobility patterns of citizens/vehicles. Transport authorities can deduce people's travelling patterns from taxis, busses, trains and boat ticketing to figure out the most crowded stations and how to improve services for people (e.g., adding more trains based on the number of people using the service). The insight also helps predict the impact on stations (if any planned/unplanned work comes up) for which customer communications and operational plans need to be implemented. Transport for London (TfL) uses their knowledge [109] about commuter routes to understand the choices made by commuters, such as the routes/platforms used, the trains they prefer (congested train/earliest train/fastest route/most comfortable), their movement around complex stations. Additionally, TfL encourages people to avoid busy stations by making travel cheaper outside the city rather than through the middle. The government working with mobile phone companies can also create real-time visualisations utilising mobile phone signal data to reveal the dynamics of the modern city (i.e. anonymous movement patterns of people and transportation systems determine the common usage of streets and neighbourhoods [110]). By anonymously combining the mobile location of citizens with the movement of public transit, pedestrians, and vehicular traffic and overlaying the data on the city map, governments can generate real-time maps that help understand the use of neighbourhoods during the day. Data help to understand the correlation between the distribution of buses and taxis with the densities of people, the distribution pattern of goods and services in the city, and the travel patterns of different social groups, such as tourists and residents. The knowledge gained helps local transport authorities optimise transport services. With the help of APNR, the government can analyse vehicle movements across its border and the number of vehicles with petrol, diesel, hybrid and electric vehicles. The authorities can compare the number of vehicle types over time to understand how air pollution depends on the kind of vehicles and the acceptance of electric cars by the citizens. City authorities can also combine vehicle mobility data with asphalt-embedded parking space sensors, allowing the system to direct drivers to the nearest available parking space. For example, Barcelona implemented a parking application, "ApparkB", to locate the closest parking space and pay online [111]. The benefits are faster parking, fewer CO emissions, and happier citizens. The city council is also responsible for road infrastructure, markings, and signs to guide drivers. However, faded lane markings and graffiti street signs make it difficult for drivers to follow the road signs and potentially endanger themselves or others. BigClout project [112] used edge machine learning to implement a roadway damage detection application

to observe the transportation infrastructure, resulting in increased productivity and reduced costs for the council.

Crowd-sourcing Data

Citizens have been working with governments for centuries to provide complaints and suggestions to improve the working of city council and government. Allowing citizens to provide information about the city helps the government provide a better feedback channel. Digitalisation has improved these processes and improved accessibility, ease, and resolutions. The system uses an open data platform (for example, on a map) where citizens can use low-cost environmental sensing kits to measure air pollution, humidity, temperature, bump sensors (on bicycles) [113] to measure bumpiness on the road and upload data to crowd-sourced maps. They can also report flooding/broken streetlights/potholes to crowd-sourced maps and open up decision-making via proper channels such as the website. Each month, the city council can select the most popular ideas from the website and implement financially viable projects proposed by citizens. For example, initiatives such as Paris' Madame Mayor, I have an idea' [114], Iceland's 'Better Reykjavik website' [115] and French platform Carticipe [116] allow citizens to propose the improvements in the city on a map, debate issues, and vote for their favourite ideas. Platforms such as Ushahidi [117], OpenStreetMap [118] and Shareabouts [119] are the mapping application for crowd-sourced information gathering which can be used. CycleStreets [120] is a crowd-sourced cycling-specific travel planner.

2.3.5 Data Interconnection at Different Levels

Data at all levels (personal, building, district and urban) are interconnected and can be used at different levels based on different stakeholders. We use the electric bicycle (e-bike) case study, a part of REPLICATE (Renaissance of Places with Innovative Citizenship and Technology), to illustrate the interlinked data at different levels. The e-bike intervention was designed to support the sustainable mobility of healthcare workers in the city. Also, to understand aspects of the transition to more sustainable transit in the form of shared e-bike schemes, which were deemed a potential future option due to Bristol's hilly topography. Community caregivers used e-bikes to cycle through the hills of Bristol to visit senior care homes/new mothers. The hilly area also encouraged people to take e-bikes better than regular bikes. Data from the e-bike trip have been accumulated and can be analysed differently. For example, evaluating the amount of carbon emission saved by using e-bikes; the effect on citizen health by using the e-bike instead of cars; whether they enjoy riding e-bikes more than driving because of less traffic and faster travel time; the usage of electric mode on a hilly area or average elevation; the preference of travelling routes such as cycle routes, roads, traffic-free streets.

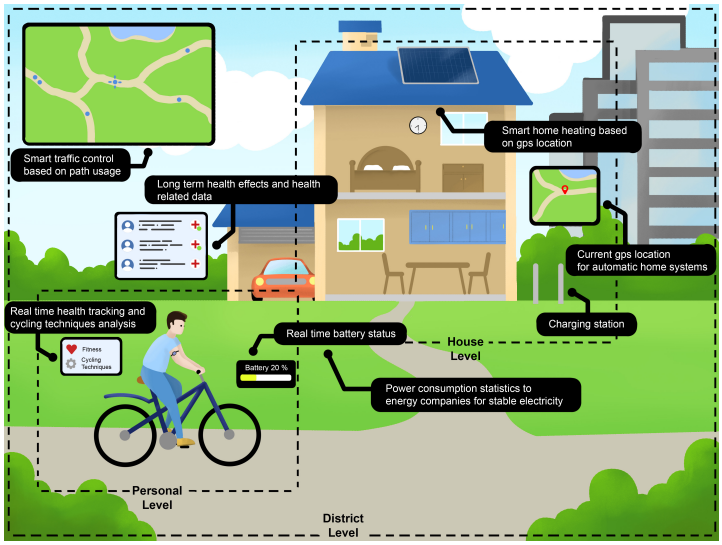

Figure 2.5: Data interconnection at different levels

Data and its value at Different Levels

The e-bike case study generated an extremely rich dataset, providing information on many aspects of e-bike usage. This demonstrates how one type of monitoring device (deployed on multiple bikes) can provide data that is useful to each of the different levels (i.e. personal, building, district, and urban) (Fig. 2.5).

On a personal level, the devices provided the e-bike riders with a detailed account of the exercise they had performed. The data helped the end user understand their fitness levels and provides insights that could improve their cycling technique. The GPS on board records a complete route trace from start to finish [121].

At a building level, the trajectory data could be used to alert an intelligent home heating system that an inhabitant is on its way and trigger the start of the heating system. Giving an intelligent metering system visibility of the e-bike's battery status also enhances electricity demand-side management opportunities, especially if historical journey data can be used to predict when the e-bike is likely to be needed next.

The generated data could help the e-bike rental scheme at the district level, providing details on e-bike maintenance requirements and giving some anti-theft functionality, potentially allowing the bike to be tracked should it be stolen. Citizens also share information about e-bikes with friends to promote cycling and community sharing.

At the urban level, the data generated can inform a wide range of different operational and strategic decisions. Personal trips can be inspected to understand the route preference that an e-bike rider has made, for example, revealing circumstances where a traffic-free passage has been selected despite this adding distance to a trip. In the short term, city authorities can use cyclist locations to optimise traffic signal control. If e-bikes are part of a bike rental scheme, the battery

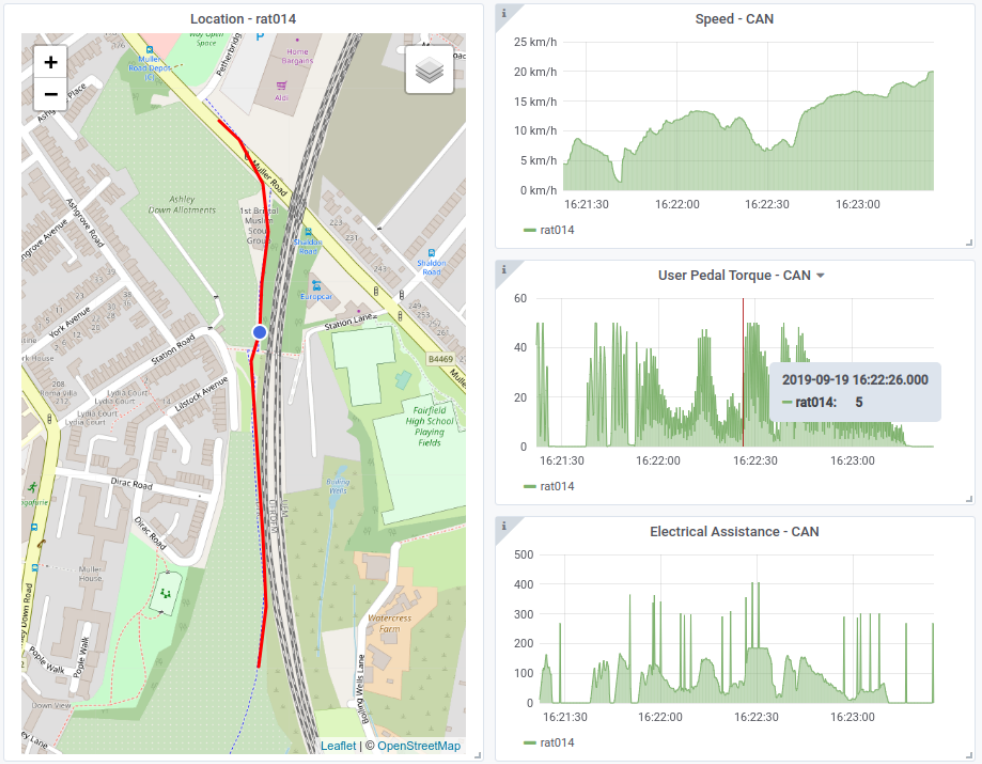

Figure 2.6: Personal-level data showing a user's cycling activity [17]

status will improve the implementation and security logistics of the scheme. In the medium term, aggregated data on e-bike trajectory can justify investing in new cycle infrastructure, such as bike paths. In the long term, health professionals can use the data to understand the long-term health impacts of adopting active modes such as cycling. BIC is already collaborating with health researchers exploring how cycling impacts type 2 diabetes.

Presenting multi-level data to different stakeholders

The data is consumed at different levels in different ways. Therefore, data must be processed and visualised differently depending on the target audience.

The e-bike case study shared personal data with the user. When the user's information is presented to the user in question, anonymisation is not needed. Several different datasets can be amalgamated to increase the insight the monitoring system can deliver. In the example of tracking the electronic bicycle shown in Fig. 2.6, GPS data and pedal torque were overlayed on each other to provide the user with detailed information about the effort they put into their cycling at each moment of the ride. Raw data might be desirable, but a simplified user interface might be more appropriate. Similarly, gamification might incentivise them to pedal harder or reduce the electrical assistance they use on a journey.

When presenting data to building-level services, the system must more carefully decide what

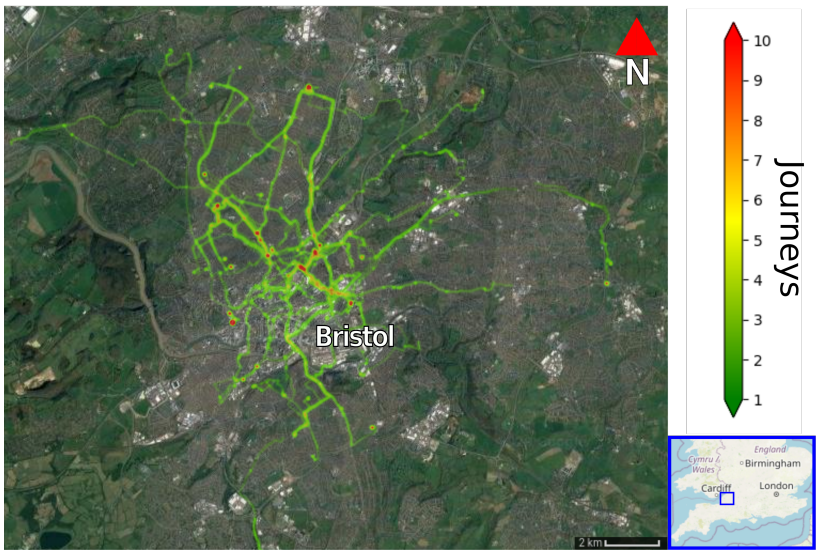

Figure 2.7: City level data, showing how often different parts of the city are visited by the monitored e-bikes (June 2019) [17].

information to expose and where to expose it. Multiple agencies will likely provide building-level services to different commercial suppliers, and each will have additional data requirements. Only some of these building-level services may be entirely trustworthy, and so only the minimum quantity of data required to accomplish their function should be exposed. In the application where the e-bike informs an intelligent metre of its energy requirements, this information alone should be provided without giving access to more personal data about where the e-bike has been or when the rider was out of the house.

With almost 3,000 journeys recorded by the e-bike monitoring system, the data built a picture of the popularity of various routes within the city of Bristol's road network (Fig. 2.7), displaying which areas cyclists regularly detour [121]. At the urban level, combined data provided significant value to planners, such as the e-bike usage heatmap (Fig. 2.7). Data from the e-bike case study has already been disseminated with the provincial authority to help them understand which city regions would benefit the most from enhanced cycling infrastructure. Centralised systems primarily collect the data, but presenting the data to that centralised system must not compromise user privacy. Knowing the travel trajectories to allow amalgamated statics about a specific route to be built up might be desirable. However, seeing a user's complete journey history provides many personal details such as home and work addresses, places of worship, family and friends' addresses, and much more. The e-bike case study has formed the foundation for further funding applications for several studies, focusing on e-cargo-bike logistics and how e-bike usage can help prevent early mortality in over 55s [121].

Data can be anonymised to resolve privacy challenges by obscuring the beginning and end of a journey or breaking the journey down into 'links' and only presenting those widely used links. Each has advantages and disadvantages, and a compromise must be reached to ensure that the

importance of the data is maintained without sacrificing data security. Each application will have its requirements and so will require its post-processing.

2.4 Responses to Urban Data Challenges

The author provided the use of the IoT and the digitalisation process (§ 2.3) at different levels (RQ2.1) and critical document parameters such as the entities, benefits derived (RQ2.2) in Table 2.1, Table 2.2, Table 2.3, Table 2.4, Table 2.5. However, data analysis and insights are only possible when the data is collected and made available to process and analyse securely and resiliently. There are challenges involved in data collection and processing, such as privacy, IoT infrastructure, and cost related to IoT (§ 2.4.1). Multiple organisations have tried to solve the challenges at different levels (§ 2.4.2). Bristol Infrastructure Collaboratory (§ 2.4.3) collects and works with citizens at different levels, fulfilling the UO's role (§ 2.4.4) in resolving a few urban data challenges.

2.4.1 Urban Data Challenges

IoT Infrastructure challenges

Smart city research projects often deploy IoT infrastructure, including cloud, edge and endpoint devices, to collect, analyse and visualise data at each level (personal, building, district, and urban). There are multiple challenges faced in the multiple phases of IoT infrastructure deployment. Challenges range from understanding project requirements to designing how to fulfil those requirements and setting up defined infrastructure to ensure that different infrastructure components work together and tested in the lab, small-scale deployment, and deployment in the real world.

Requirements analysis and deployment challenges are project-dependent and depend primarily on citizen preferences, communication between stakeholders, and expectations of different collaborators/partners. Designing the IoT infrastructure involves challenges in ensuring end-to-end security of the platform and having a resilient infrastructure in terms of network, device, thermal, and power. It also includes data-related challenges such as storage, reduction, access, integration, harmonisation, monetisation, curation, availability, and liability. It is also essential to understand how various users/devices/applications are authenticated and authorised, including how the credentials are stored securely. Implementing the IoT infrastructure presents challenges in providing IoT devices and ensuring their network connectivity is secure and reliable. Challenges include understanding how applications will be deployed on IoT devices and ensuring compatibility between hardware architectures. With the increasingly installed hardware devices and software, it is essential to perform accounting and monitoring to be aware of the IoT infrastructure. Additionally, there are challenges around the IoT infrastructure's scalability, modularity, extensibility, adaptability, and reproducibility. IoT deployments often rely on battery-

powered wireless embedded devices with severely limited (in terms of processing, storage, and networking capabilities). Such devices are generally difficult to manage and often suffer from multiple security vulnerabilities [122].

For example, citizens use electronic bikes to travel and might require cycling statistics on a personal level. At a building level, energy companies might benefit from the battery status of electric bikes. At the district/urban level, health officials might benefit from cycling data to understand the long-term impacts of cycling, and transport agencies might benefit from understanding the use of cycling routes. In this case, collecting data at all levels is complex and requires custom-designed hardware. If the citizen buys an electric bike, he may have access on a personal level. Still, it would be difficult for other stakeholders (energy companies, health officials, transport agencies) to get the data at a different level. BIC plays a role in solving such challenges by designing custom hardware and providing data at different levels.

Privacy challenges

With the ever-increasing number of IoT devices that capture almost every essential physical parameter available, the citizen's privacy is potentially at risk. Multiple physical parameters can be combined to understand an individual's habits, presence, or even activities at home [123]. For example, it is possible to track an individual's movements inside their home [124], correlate them with the metadata gathered by a Wi-Fi router, and draw a timeline of the activities of a person. In terms of government services, law enforcement agencies have piloted facial recognition cameras in public places [125] to detect people with a criminal history. China's government has developed a social credit system [126] to rate citizen trustworthiness by calculating their credit score based on financial activities, including shopping habits, traffic tickets, taxes payments, and leisure activities [127, 128].

Data challenges at the personal level are about the user's privacy and how the information is used by a company supporting personal monitoring infrastructure, such as, e.g. a tracking kit provider.

Building-level data challenges are related to user privacy, security, and activities. For example, data on lighting, heating, IAQ, and energy monitoring can inform on house occupancy and identify activities such as cooking, bathing, and watching television; information about recycling and shopping lists can be used to recognise a homeowner's eating patterns. Furthermore, using IoT at the building level could become a security risk. For example, CCTV being hacked, digital door locks malfunctioning or being hacked, leaving citizens outside their houses. At the building level, citizens have seen many products (smart plugs, washing machines, ovens, lighting, baby monitors); all appliances can now be purchased with IoT-enabled features. They improve QoL by allowing easier access and management, reducing costs, and increasing productivity [129–131]. On the other hand, it can make a home more vulnerable to cyber-attacks (as seen recently when baby monitors were hacked) [132].

Data at a district level is about the different services available to the citizens and the community network in the neighbourhood. It can contain data about the citizen, what item or information they shared, and with whom (applications that allow sharing food/items and others). Such applications often provide specific terms and conditions users must accept to use the application. However, as citizens (the person sharing and the person receiving) would be acquaintances or in agreement (terms and condition - acceptance), privacy implications depend on the application privacy agreements.

Data at the urban level provides information on city life (for example, vehicle movement patterns, number of people in tourist spots, and park use). They are generally anonymised because they do not relate to any specific individual [133, 134]. Governments also work with mobile phone operators to obtain data that can provide details of citizen habits and preferences, creating privacy concerns. Furthermore, the challenge of using these data can lead to skewed analysis if users turn off tracking/Bluetooth or have enabled privacy-preserving options. It is also vital to note that data access can create inequality (since more QoL parameters about neighbourhoods are known, wealthier people tend to have more ability to relocate than citizens who do not have awareness) [135, 136].

IoT Cost and Benefit challenges

There are capital and operational costs associated with the IoT infrastructure [137], and the city councils must understand the benefits and return of investment in sensor installation and maintenance. Costs include the requirement for regular upgrades and replacements of sensors, potentially due to natural hardware failure or vandalism. For example, in recent developments in a country, protesters vandalised [138] smart lamp posts [139]. Citizens should be involved in digitalisation in their city and work to raise their awareness [140] (what they measure and their visual appearance) of various devices installed in the city. Citizen awareness of their city-systems reduces the probability of misuse [140, 141], e.g., vandalising non-biometric devices such as CCTV. Another maintenance cost is the normal or extreme weather events that damage the IoT infrastructure.

Data accuracy depends on the sensor's quality (low-cost vs high-cost) and the location of the sensor from the user (near vs far). The city authorities can enhance the precision of the data by installing more sensors and increasing the granularity of the data. However, it also increases installation costs and presents challenges to data scalability. Challenges raise the question of whether the energy cost of running billions of IoT devices outweighs its benefits [142, 143].

A challenge also lies in determining the optimal placement of edge devices to minimize installation and maintenance costs. The city council faces the task of ensuring that these devices, along with their sensors, can offer a comprehensive view of the city or effectively gather the data needed for specific purposes. The placement of devices can be approached in different ways: through efficient modeling or in alignment with project requirements (strategically positioning

them near intersections, buildings, or industrial areas) and by taking into account input from citizens and the community.

Another bottleneck for smart city projects is achieving financial sustainability [144]. Initial funding often comes from the government, but this rarely covers long-term operational costs; the ideology of open data leaves few avenues for revenue generation. It has been suggested that, often, innovative city projects focus on technology deployment rather than application and result, making the transition to financial sustainability painful [144].

2.4.1.1 Data collection limitations

There are also some limitations to data collection. Data is often collected or maintained in a proprietary protocol or manner, such as personal data collected using personal devices like Apple Watch or FitBit, or building data collected by the building owner. In such cases, the data can be exported by the citizen/building owner and provided to researchers for analysis under data sharing agreements. There have been projects such as Databox [145] that aimed to enhance accountability and give individuals control over the use of their data by providing an open-source personal networked device that collates, curates, and mediates access to an individual's data by verified and audited third-party applications and services. In an IoT infrastructure consisting of cloud, edge, and endpoint devices with applications running on the edge, data can be collected from the applications based on the research agreements with the application owners.

Moreover, data collection typically originates from the endpoint, gets aggregated at the edge, and is subsequently transmitted to the cloud. However, there are scenarios in which data necessitates processing at the edge, followed by an action execution either at the endpoint or within the edge tier. Alternatively, it may require user notification for a specific task. In such instances, the application code running on the containers can be adapted to handle data processing and transmit notifications or action directives through various communication channels. For example, a application monitoring the elderly human fall detection using Wi-Fi signals [146] might want to notify relevant authorities.

Additionally, applications operating at the edge can be categorized based on their resource requirements, including RAM, CPU, network bandwidth, data processing capabilities, and the type of data necessary to execute specific actions. For instance, applications in smart cities often fall into the low-capacity category, as they primarily collect environmental sensor data. Conversely, high-capacity applications, such as those facilitating low-latency communication for dedicated virtual corridors for emergency vehicles, require robust resources. To manage resource allocation effectively, Kubernetes pods and containers offer the capability to configure rate limits for CPU, RAM, and network bandwidth (both incoming and outgoing) through resource management settings. Furthermore, to support critical functions like emergency response and data with low-latency requirements, efficient communication between the cloud and the edge or vice versa can be achieved by implementing multi-network protocols. This approach enhances

the resilience of communication networks in scenarios where timely data exchange is crucial.

2.4.2 Managing the Complexity of Urban Data

Different responses have been received from different entities on different scales (city/national/ public-private partnerships) to collect and manage the complexity of urban data.

City Response

Local authorities have launched several initiatives to improve their cities regarding fairness, health, sustainability, resilience, equality, diversity, environment, aspiration and success for everyone, based on quality data and IoT to tackle challenges caused by urbanisation. In Bristol, Bristol city council (BCC) worked with Bristol is Open [147] to deliver research and initiatives for developing a smart city, including a city-operating centre and providing citizen-centric solutions. From an environmental perspective, BCC has launched a climate emergency plan [148] and committed to being carbon neutral and climate resilient by 2030. They have identified ten fundamental scopes: transport, buildings, heat decarbonisation, electricity, consumption and waste, business and the economy, and others. BCC has launched the One City Plan [32], which sets the ambitious vision for the future of Bristol until 2050. The plan is built on six themes: connectivity, health and well-being, homes and communities, economy, environment, learning, and skills, supported by key enablers such as culture and technology.

Technology plays a role in developing solid evidence-based urban datasets to measure the city's progress and the success of the one-city and climate emergency plans.

Public and Private Partnerships

Multiple projects have enabled the collection and analysis of urban data. For example, the EU and the Cantabria government funded Smart Santander [97], a collaboration between 15 partners from the public sector, enterprises, universities, and research centres. Smart Santander deployed multiple IEEE 802.15.4 devices, General Packet Radio Service (GPRS) modules, and Radio-frequency identification (RFID) tag/ Quick Response code (QR) code labels deployed in both static sites (streetlights, facades, bus stops) and mobile vehicles (buses, taxis) for different smart city use cases.

UMBRELLA [20] is a joint project between South Gloucestershire Council and Toshiba, with the support of the West of England Combined Authority and the Local Enterprise Partnership. UMBRELLA project has deployed more than 200 nodes spread throughout the South Gloucestershire region [94] and a few nodes at Cardiff University [95] that contain air quality sensors. Similarly, in collaboration with JCDecaux, Microsoft implemented Project Eclipse [96] and installed 115 low-cost solar-powered urban environmental sensors connected by a cellular network in Chicago to observe pollution at satisfactory spatial and temporal resolutions [96].

National response: UK Collaboratorium for Research On Infrastructure and Cities (UKCRIC) and its Urban Observatories

In the UK, the UKCRIC project aims to provide a network of facilities and methodologies (including dedicated spaces, testbeds, methods and tools) for research, innovation, pedagogy, and collaboration. UKCRIC was established as a distributed research capability in response to the perceived need for investment and regeneration of the UK infrastructure [149]. It attempts to focus on problem-specific challenges such as climate change that cannot be solved by one organisation [150]. UKCRIC's vision describes the function of the observatories as entities that generate evidence concerning infrastructure development. In addition to large-scale laboratory facilities designed to meet the challenges related to physical infrastructure, a network of collaborators UO hosted by different universities was developed to increase understanding of how cities function and to support decision makers in managing city infrastructure through the use of IoT and digitalisation. UKCRIC's vision describes the function of the observatories as entities that generate evidence concerning infrastructure development.

An UO provides a platform for collecting, modelling, and analysing data to inform local, city, regional, and national decision-making. UOs help stakeholders understand the challenges and perform evidence-based interventions. UOs promote platforms as accelerators of change and innovation to support decision-making and develop infrastructure delivery and operational insights. They provide go-to places to help other cities establish observatories by sharing learning and best practices.

Six observatories have been set up in Newcastle, Bristol, Sheffield, Manchester, Birmingham, and Cranfield. The different UOs have different focus based on the expertise of the host university. However, they are aligned in fulfilling technical aspects, such as designing and deploying urban sensing networks, data curation, and providing analytics to turn data into information. The focus is on creating a co-learning and co-production environment that stimulates and supports full participation. This accelerates smart city planning and long-term, evidence-driven strategic and operational deployment (innovation and change).

UOs also foster collaboration between people and organisations interested in infrastructure. UKCRIC UOs will continue to address urban challenges, drawing on practical experience from the network to enable the next generation of infrastructure innovation [151]. Resources and data are shared and available for download, allowing researchers to access data from other cities. It also helps researchers work on problems that require access to multi-city data for insight development.

To date, much of the output of UO has been the development of methodologies for the deployment of low-cost sensor networks [152] and the analysis of data from these networks [153, 154] however; more work is required to understand how these data and analysis can be used to support decision-makers in implementing meaningful change [155, 156]. There has been a significant use of publicly available UO data throughout the COVID-19 pandemic [157].

Our work on the smart city framework developed (chapter 4) in this thesis is devised through the mechanism of UO, which could be one of the delivery stakeholders. The author demonstrates a way to design a smart city framework that can be deployed in the lab, real world, or city level. The different activities of the UO (§ 2.4.3) help the author to understand the requirements of a smart city framework. The UO supports the author by providing the opportunity and funding to enable work on the smart city framework.

The smart city framework and architecture can be developed in several ways, such as observatories and private organisations. The observatory is often more open to innovation with multiple engagements, whereas the private ones might be closed sources and focus more on establishing commercial viability.

2.4.3 The Bristol Infrastructure Collaboratory

The UO in Bristol (BIC) originated from people working in various infrastructure sectors, fostering cross-discipline collaboration across the university. A network of stakeholders has formed a portfolio of relevant themes, including asset monitoring for infrastructure, energy systems, mobility, people-space interaction, water quality, citizen detection, and mobile use using digitalisation and IoT.

BIC aims to support researchers/organisations by collecting urban data through observatories that researchers can use for analysis and collaboration. BIC believes in collaboration, enabling the creation and implementation of the technology required for the research project. BIC is also interested in developing soft infrastructure in the form of people, networks, relationships, and processes to enable technology to work well. Soft infrastructure is a conduit for brokering meaningful collaboration between local/international stakeholders aimed at urban innovation to deliver next-generation infrastructure systems. BIC works across all different levels of IoT, both for data collection and usage.

Personal and Building Level

BIC does not collect data at a personal level at present; however, data collected at other levels related to the city can be presented to the citizen.

BIC works in the people-space interaction area at a building level. ”Urban Vision” [158] aims to understand the impact of different visual illusions/patterns in an environment on health and well-being, such as people’s mood, behaviour, and gait. BIC set up an interactive travel exhibition [159] consisting of a walkable corridor installation. They monitored the participant’s movements (§ 2.3.2 - Security), visual cognitive processing load, reaction times, and gait kinematics using retro-reflective markers attached to different body parts, detected by a motion capture system and additional cameras.

BIC works in microgeneration, peer-to-peer transactions, consumer behaviour, and energy performance monitoring (§ 2.3.3 - Efficiency) in the energy area. BIC works as a living lab for

open data energy management and collaborates between campuses and communities to reduce demand and increase renewable energy usage. BIC has installed sensors (MCU520 [160]) in several campus buildings to measure the condition of the electric grid (e.g., reactive energy, mean voltage), improve energy use, and identify where improvements can be made. BIC collaborated with the university sustainability services responsible for energy consumption across campus and is forming a smart sensing trial to understand the impact on the university's energy consumption. Alongside this, BIC has installed several smart energy systems, including local energy storage systems, in single- and shared-owner homes, where data is collected to understand the impact of these systems and how energy savings brokerage can be implemented. BIC participates in the TwinERGY [161] project, a European project that will design, configure, and combine a creative suite of tools, services, and applications for energy customers. The project aims to empower citizens to track their energy use, actively participate in the market, increase understanding and knowledge about consumption habits and energy behaviours, increase local intelligence and participation of consumers through the Digital Twin mechanism, and encourage green and sustainable ecosystems.

In monitoring infrastructure assets, BIC collaborated with the Clifton Suspension Bridge Trust [162] to collect structural health monitoring data from the Clifton Suspension Bridge, identifying new methods to characterise pedestrian and vehicle traffic [152, 154] to improve bridge security and optimise maintenance.

Working collaboratively with city stakeholders (Knowle West Media Centre (KWMC)), BIC developed and deployed the FrogBox urban sensor (measures temperature and humidity) through co-creating 'Bristol Approach' [163]. BIC used FrogBox in several pilots to address the problem of residential dampness in student housing [164]. Continuing citizen-led innovation in urban sensing, BIC has deployed several pollution monitors (§ 2.3.2 - Indoor Environment) in schools and community centres to use air pollution data to encourage parents to adopt less polluting transport modes. Continuing to work with the citizens, Cotham Hill residents were concerned about the pedestrianisation scheme that the council had introduced in the area. There was concern that this would lead to increased noise pollution from drunk university students and restaurant activities. Citizens sought to collect data on the scheme's performance and have the requisite evidence to feed into the council's consultation on plans for this scheme to become permanent. BIC surveyed residents and provided citizen sensors (SCK) that collect urban data using automated measurements.

District and Urban Level

BIC organises events for Science, Technology, Engineering and Mathematics (STEM) outreach, training, citizen-driven urban innovation and co-creation at the district level. BIC collaborated with KWMC to implement a citizen maker programme (§ 2.3.3 - Community) enabling engagement with citizens, understanding their challenges, and developing IoT sensors (measurement

of temperature, humidity, air pollution) to solve these challenges. To showcase BIC work to the public, Bristol-based musicians and sound artists collaborated with the University of Bristol's Jean Golding Institute, Music and Engineering departments to construct a harp structure [165] that musically depicts the structural health data from the Clifton Suspension Bridge.

BIC works on intelligent transport systems, EV, multimodal mobility, and people's perception of space on mobility and people-space interaction at an urban level. For the REPLICATE project, data such as battery voltage EV, battery usage patterns, and cycle peddle torque were collected using speed, rotation, and other sensors. BIC investigates interactions between complex transport infrastructures, people in shared spaces [166] and the sensory impact of city infrastructure on the health and well-being of citizens. It aims to create shared spaces that minimise conflict between pedestrians, cyclists, and car drivers, improve people's experiences of shared space, and encourage effective use of shared public infrastructures. Furthermore, BIC explored the role of sensory information to encourage people to travel more actively, aiming to identify how this affects people's walking behaviour and use of public transport.

BIC implemented and deployed custom hardware to monitor a fleet of electric bicycles (§ 2.3.5) and understand the aspects of mobility (use, multimodal aspects), energy demand, and associated issues for supporting infrastructure for mobility (§ 2.3.4 - Transport mobility). A different collaboration with the School of the Art Institute of Chicago, examined object mobility using custom-made data collection tools. The project named "ofThings: Experiments in Object Mobility" [167] aims to engage with the citizen of Bristol and explore how citizens could engage in shared use mobility and active transportation modes to mobilise objects. The emphasis is on pedestrian movement and objects portage without dedicated motorised fleets.

BIC is developing a series of smart infrastructure testbeds to understand the deployment and use of affordable IoT sensors (capable of measuring characteristics such as pH, dissolved oxygen, temperature, turbidity, and electrical conductivity) to comprehend the influence of the built surroundings on the condition and quality of water [153] and provides early warning of floods. A multi-parameter water quality monitoring system [168] has been deployed in Bristol's floating harbour (§ 2.3.4 - Environmental Factors). It demonstrates the feasibility of collecting and presenting high-frequency real-time water quality data. The study aims to create a model for predicting water quality established on real-time data to assist policymakers and decision-makers.

BIC deploys suitable urban technologies such as AoT and SCK to collect urban sensing data (temperature, humidity, air pollution) to support decision-making related to air pollution challenges. Furthermore, BIC developed a portal [169] to provide air pollution data (§ 2.3.4 - Environmental Factors) for any location in Bristol by collecting data from devices around that location. It holds air pollution data from different sources, such as the SCK project, Luftdaten, Open Bristol Data, and DEFRA AURN.

2.4.4 The Emerging Role of Urban Observatories

UO aims to work with citizens, city councils, and organisations to solve urban challenges. Data collection often requires an IoT infrastructure to collect, analyse, and visualise data. The infrastructure must be designed, implemented, and deployed in public or public spaces for citizens. An entity or organisation such as a non-profit organisation, university, industry partner, private institution, government, or a collaboration between multiple organisations must take ownership of design, implementation, and deployment. In our case, BIC integrates multiple partners' requirements to design and deploy such an architecture. UO helps manage the complexity of sensing projects. It can support the design process, develop an architecture, platform, and visualisation for the end user, and deliver a solution by assembling technology skills as an external stakeholder and entity. UO provides the ability to test at scale in the real world, outside of lab or simulation, and understand the implications of deployment and other hidden overheads such as value vs cost. For example, the e-bike monitoring device was prototyped in the lab and later deployed in Bristol. BIC provided hardware sensors and infrastructure for data storage, analysis, and visualisation. To support activities that generate data at different levels (from personal to urban), BIC provides the collection, storage, and analysis of city-related data.

UO enables secure and transparent access by ensuring that citizens participating in the case study understand the ethical process of signing an agreement detailing the rights and purposes of data collection and processing.

BIC's position in academia does provide some benefits, such as a reduced requirement for product maturity compared to our industrial counterparts. As academic research engineers, BIC can experiment with novel technologies, even if they do not work 100% of the time. BIC can learn a lot about the practicality of these novel technologies for urban management applications. Academia is perceived as more 'trustworthy' than industry [170], which could increase the willingness of the operational staff to work with us. However, perceptions of academia are not entirely positive: some people believe that academics only want to observe a situation [171] and not 'get our hands dirty'. There is also the possibility that BIC (academia) will be perceived as naive to the complexities of the problem situation it engages with [172] and can be perceived as being an 'outsider' [173], which will have an impact on how those within the urban management soft system interact with the BIC and any projects in which it is involved.

A UO can be a resource to collect, store, and analyse city-related data. A UO provides a focal point for researchers, city leaders, and other city stakeholders to understand where additional sensing would benefit and appropriately target activities.

BIC brings people with different skill sets and relevant backgrounds to analyse the collected data and survey questionnaires and respond to the correct stakeholders. The bringing of people together and the production of the desired results is very difficult for individual entities such as citizens, councils, or anyone who lives between layers and is trying to join the dots. For the e-bike intervention, the BIC connected researchers with the skills to deploy the appropriate sensors and

Table 2.6
A summary of Urban Observatory and their features

Aspect	**Description**
Purpose	Conduct research and gather data to understand urban dynamics, inform decision-making, and improve city living.
Scope	Monitor various aspects of urban life, including transportation, air quality, energy consumption, and social behaviors.
Data Collection	Utilize advanced technologies like IoT, sensors, and GIS for real-time data collection from diverse sources.
Collaboration	Involve collaboration between cities, universities, research institutions, and sometimes private sector partners.
Focus Areas	Address challenges related to sustainability, resilience, urban planning, and policy development.
Technology Integration	Leverage data analytics, machine learning, and other technologies to analyze and derive insights from collected data.
Smart City Initiatives	Contribute to the development of smart cities by utilizing technology to enhance efficiency and well-being.
Decision Support	Provide data-driven insights to support decision-making in areas such as infrastructure development and resource management.
Public Engagement	Foster public awareness and participation through data visualization, open data initiatives, and citizen engagement.

modifications. The data collected provided information on how e-bikes could improve community nurses' activities to support old care homes. The project was carried out in partnership with the hospital and the council, providing bikes under a different scheme. BIC generated meaningful data on bicycle use by continuously measuring data and providing quantitative metrics combined with qualitative data from interviews and questionnaires to determine the impact of e-bikes on community nurses' activities. The BIC provided the technical capabilities to collect and analyse the data.

The emerging UO concept facilitates the action from the collected data. However, it may have little impact, as the data might require other stakeholders to perform the analysis or provide it to someone else. For example, health professionals or mobility models can collect and combine the data with their internal data to perform the analysis. Cities can institutionalise infrastructures such as urban observatories to facilitate data collection, storage, and sharing to citizens, researchers, and innovators to solve the city's challenges. In summary, BIC solves urban challenges by designing, implementing, and providing IoT infrastructure to collect urban data and bringing researchers and organisations with different skills to solve complex urban challenges. Table 2.6 provides a summary of urban observatories and its features.

2.5 Conclusion

This chapter attempts to contextualise urban sensing and its resulting data for city governance and explore its potential and challenges. The work makes a strong case for the ubiquitous nature of IoT and digitalisation. Technology classification creates an exciting way to digest and conceptualise how this technology can be interlinked. The work presents different situations in which city councils can use IoT data to optimise citizens' and cities' quality of life. The situations are catalogued as personal, building, district, and urban levels according to where or at which scope data are generated.

We started by identifying and categorising what types of data can be collected from a citizen's perspective using readily available and accessible IoT devices and examined how this information could be used in city decision-making and management - with all its caveats. More specifically, we discuss the role of IoT and digitalisation in improving urban society at various levels and helping

cities become more innovative, sustainable, and resilient. We subsequently looked at the role of citizen-led action and the engagement mechanisms that could lead to leveraging individual data contributions at the city scale, such as community efforts and Open Data initiatives. We discussed the emerging concept of an Urban Observatory and its role in curating the various IoT data streams effectively and in a trustworthy manner. We demonstrate how the latter could leverage data collection hierarchies to contribute to providing quality data for city management. Furthermore, urban observatories connect citizens, academics, industry, and government to solve community and city challenges and work toward designing and deploying urban-wide sensing networks, data curation, and analytics to turn data into information. However, data analysis and reaping its benefits is only possible when the data is collected and available for analysis. There are multiple challenges and risks associated with data collection. Additionally, challenges are associated with developing, deploying, and maintaining the infrastructure required to collect, process, and analyse data and provide information to citizens, policymakers, and governments. The next chapter (chapter 3) will explore the challenges.

The author provided the use of IoT and the digitalisation process at different levels (RQ2.1 - § 2.3) and critical document parameters such as the benefits derived (RQ2.2 - Table 2.1, Table 2.2, Table 2.3, Table 2.4, Table 2.5). The author provided the different initiatives that enable the curation of actionable data and help to better understand the different phenomena (such as environmental and climate change) (RQ2.3 - § 2.4.2). Contribution to knowledge in this chapter (C1) is the list and classification of IoT data collected at different levels (personal, building, district, and urban), the benefits derived from them, how they are interconnected, and the different initiatives that enable the curation of actionable data in solving urban data challenges.

CHAPTER 3

Challenges in the Design and Implementation of IoT Infrastructure in Smart-Cities: A Systematic Review[1]

The second chapter focused on identifying data that can be collected at different levels and the analysis performed, and how IoT and digitisation can reduce costs and increase productivity. The next natural question is how we collect data in a secure, resilient, and reliable way and provide it to data scientists for data analysis without compromising security and data security regulations. Moreover, how do we share the insights with the broader public, such as citizens and policymakers, leading to an increased quality of life for citizens and improving city council services?. The motivation for this chapter is to understand the challenges faced in collecting, analysing the data and sharing insights with the broader public.

City councils, in association with researchers and universities, participate in multiple research projects such as SPHERE [174, 175], REPLICATE [176] and Twinergy [177] that aim to improve energy use, mobility, human well-being and productivity, decrease the energy footprint, and improve city resilience and sustainability [19]. The AoT team [19] has conducted various workshops with multidisciplinary academics and citizen communities to understand how IoT technology that comprises sensors, cameras, and computing 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 apparatus for research and development [19], simultaneously testing a new sensor, communication and computation

[1]This work has been submitted to IEEE Communications and Survey entitled "Challenges in the Design and Implementation of IoT Testbeds in Smart-Cities: A Systematic Review". The first author wrote the abovementioned paper and proposed the ideas/approaches, design. The other authors provided their valuable reviews and suggestions to improve the paper. Sam Gunner and Theodoros Spyridopoulos guided the § 3.2.4. Antonis Vafeas guided the Fig. 3.5, Fig. 3.6 and Fig. 3.4.

devices.

Advances in wireless communication and 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 deploy proprietary IoT testbeds for urban data collection at the personal, building, district, and urban levels. Managing the operation of the IoT infrastructure while considering the emerging security and privacy challenges, such as data privacy controls, network security, and device security updates, is challenging.

In this chapter, we focus on the challenges faced by multiple smart city research projects that aim to collect urban data, provide data to researchers for analysis, and provide information to citizens, policymakers, and city councils.

The chapter is organised as follows: § 3.1 provides the research questions and our approach. § 3.2 provides the background knowledge of smart city research projects, testbed and monitoring architecture, and the V-model. § 3.3 provides the challenges faced by the research projects mapped to the V-model (systems development process model). § 3.4 concludes the chapter.

3.1 Research Questions and Approach

Research Questions

The chapter 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. It identifies the various projects' challenges, organises them under the V-model's development lifecycle levels, and provides a reference guide for future projects. Understanding the challenges will facilitate current and future smart-cities IoT research projects to reduce implementation time and deliver secure and resilient testbeds.

This chapter attempts to investigate the following research questions:

RQ3.1 What are the challenges that a smart city research project faces in deploying IoT infrastructure that collects urban data?

RQ3.2 Can we classify the above challenges in different phases of research projects to help future smart city projects?

Research Approach

To answer the research questions, the author performed a systematic analysis of the challenges in developing urban monitoring IoT testbeds, relying on the experiences of the authors of relevant UO projects, innovative city projects and the study of relevant literature. These projects include Harbourside water quality monitoring [178], Clifton Suspension Bridge [179], electronic bike monitoring [121], damp residential detection [180] and SCK deployment in the Cotham Hill

Table 3.1
Smart city 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 3.2
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 [19]	130 EN (streetlamps) with on-board EP	Environmental, Camera	EP (I2C/SPI)	→	EN (4G)	→	CS
e-Agriculture [182]	EN (Lab deployment)	Light, temperature, soil pH and humidity	EP (Analogue)	→	EN		
Living Labs [183]	150 EN, 800 EP (120 location)	Air quality, microclimating, bat monitoring	EP (RPL)	→	EN (2G)	→	CS
Connected Vehicle Testbed [184]	3 Fixed EN (FEN), 2 Mobile EN (MEN)	Vehicle position data	MEN (wireless)	→	FEN (wired)	→	CS
Wireless Environmental Sensors [185]	1 EN, 7 EP (Lab deployment)	Environmental	EP (Bluetooth)	→	EN (LAN)	→	CS
Solar-powered WSN [186]	82 EP (real-world deployment)	Temperature, RSSI, battery level	EP (WSN) w/ sink	→			
Community Elderly Care [187]	EN, EP (70 elderly homes)	Motion, door contact	EP (Z-wave)	→	EN (cellular)	→	CS
IEEE802.15.4 Connectivity Traces [188]	350 EP (Office environment)	RSSI, PDR	EP (802.15.4) w/ sink	→			
LOFAR-Agro [189]	109 EP, 3 EN, (real-world deployment)	Temperature, humidity	EP (WSN) w/ sink	→	EN (WiFi)	→	CS
3E Houses [190]	(100 homes)(6 EP/ 1 EN)/home	Energy consumption data	EP (Zigbee)	→	EN (WiFi)	→	CS
New York Noise sensor network [191]	55 Nodes (1 EP and 1 EN)/node	Noise data	EP (USB)	→	EN (WiFi)	→	CS
Padova Smart City [192]	1 EN, 8 EP	Temperature, humidity, benzene	EP (802.15.4)	→	EN (WiFi)	→	CS
Flash Flood Monitoring [193]	3 iter. of IoT device deployed; EN, EP	Water levels	EP (USB)	→	EN (cellular)	→	CS
Smart Santander [194]	50+ EN, 700+ EP	Environmental	EP (802.15.4)	→	EN (Wired/Wireless)	→	CS
City of Things [195]	No details?	Air quality, traffic monitoring, parking	EP (WSN)	→	EN (multi-radio)	→	CS
SADMote [196]	5 EN, 12 EP	Environmental	EP (WSN)	→	EN (WiFi)	→	CS
SensorScope [197]	≈6EN, each serving ≈100 EP	Environmental	EP (WSN)	→	EN (GPRS)	→	CS
EpiFi [198]	≈ 18 locations (2 EP, 1EN)/location	Environmental	EP (WSN/WiFi)	→	EN (WiFi)	→	CS
Parking System [42]	2 EP, 3 EN	Parking, Light sensor	EP (Lora)	→	EN (Lora receiver)		
Residential Sensing [197]	≈ 20 homes ≈ 1200 EP	Temperature, light, door	EP (Z-wave)	→	EN (WiFi)	→	CS
Water consumption [199]	30 homes, 1EN, ≈ 4 EP	Water consumption	EP (433MHz)	→	EN (WiFi)	→	CS

Pedestrianisation Programme, as well as others [176, 177, 181]. Table 3.1 lists the different research projects in which the authors participated and provided their experiences. Table 3.2 lists similar intelligent city projects that the author referred to understand the challenges faced in the projects mentioned in the research publications.

The methodology the author used to understand the challenges faced in the smart cities research project is based on interviews and reviews of the literature. To collect the necessary data for our research, we conducted semi-structured interviews with system architects and the implementation team of European research projects on IoT platforms and testbeds for urban monitoring [176–181]. Table 3.1 lists the different research projects in which the authors participated and provides their experiences. The discussions focused on the challenges the participants faced during the development, implementation, and management of IoT infrastructure and testbeds. The author asked open-ended questions with a free-flowing approach by asking the interviewee questions, and the conversation 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 provide 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 author captured additional challenges based on their reflections on their experiences as members of smart-cities projects (Table 3.1). Furthermore, we thoroughly review the relevant literature on infrastructure deployment (Table 3.2).

To facilitate the exploitation of our work by future projects, we categorised the identified challenges based on the stage of the project lifecycle in which 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 (§ 3.3) in the various projects and organise them under the V-model's level (§ 3.2.4) to formalise the development process and provide a reference guide for future projects.

3.2 Background

This section briefly describes smart cities research projects, testbeds, three-tier architecture, and the V-model.

3.2.1 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 or structural health monitoring of buildings and bridges. Table 3.1 and Table 3.2 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, multiple projects such as Smart Santander, UMBRELLA [200], and AoT [19], whereas projects such as SPHERE [201] and REPLICATE [176] have deployed devices in citizen homes. Smart Santander deployed multiple IEEE 802.15.4 devices, GPRS modules and joint RFID tag/QR code labels deployed in 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 for testing wireless algorithms and smart city sensing (temperature, air quality, and noise). The AoT project deployed edge nodes in Chicago to gather real-time data on the city's surroundings, infrastructure, and activity for research and public use. SPHERE deployed a multimodal platform of non-medical residence sensors to operate as a prototype for future residential health 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.

3.2.2 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 [202]. 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). Managing such an infrastructure is challenging [189]. 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 [189]. 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 [203]. The author 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 [204], GENI [205], GRID5000 [206], FED4FIRE/FED4FIRE+ [207], FIT-Cloud [208], Emulab [209], PlanetLab [210], PRAGMA [211], DETER [212], NOR-NET Core [213], SAVI [214] **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 [215], SmartSantander [216], City of Things [217], UMBRELLA [200] **iii.** data-collecting research testbed that collects data from citizen house or public spaces, i.e., SPHERE [174, 175], REPLICATE [176], Twinergy [177], 3E Houses [218], SONYC [191], AoT [19], Scallop4SC [219], Padova [192].

3.2.3 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 [174, 175], REPLICATE [176], Clifton Suspension Bridge [179], AoT [19] and others [193, 220].

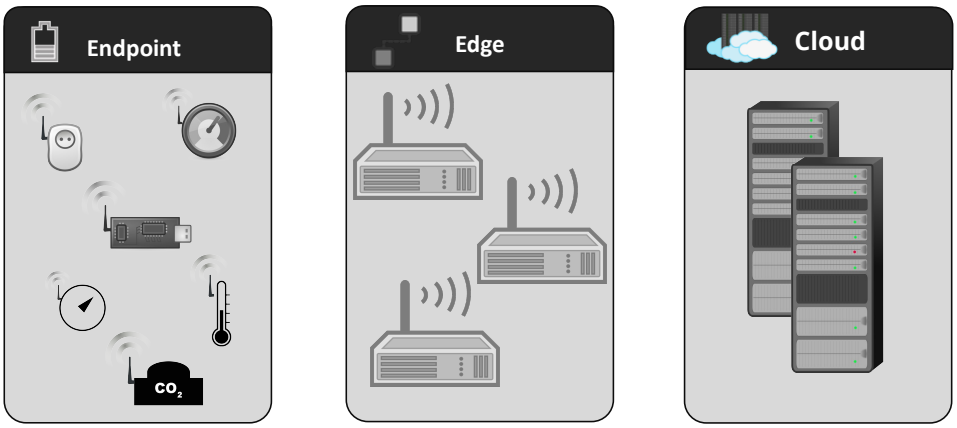

Figure 3.1: Typical three-tier architecture for a smart city

Fig. 3.1 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 [221]. 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.** 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 [222].

The edge tier collects the data sent by the endpoints and either processes it or sends it in a raw format to the cloud tier [223] 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 [224]. 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 generally connect 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 [195]. 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).

The edge hardware selection depends on both specific use cases and project prerequisites. In instances where the objective is as straightforward as gathering environmental parameters, a Raspberry Pi can adequately serve as the edge device. However, if the machine learning application demands GPU capabilities at the edge, a more advanced hardware setup may be necessary. Options in this category include the Jetson Nano, Jetson AGX, Coral TPU, or specialized hardware tailored to the task at hand. The applications can be run on suitable hardware using Kubernetes Node Feature and NVIDIA GPU discovery feature. By evaluating the unique requirements of each use case, it becomes possible to pinpoint the ideal edge devices for deploying applications. The selection of edge hardware demands careful consideration based on the use-cases, as its ramifications can extend to heightened network bandwidth demands, larger enclosure sizes, increased heat dissipation, and additional power requisites.

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 [223]. 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

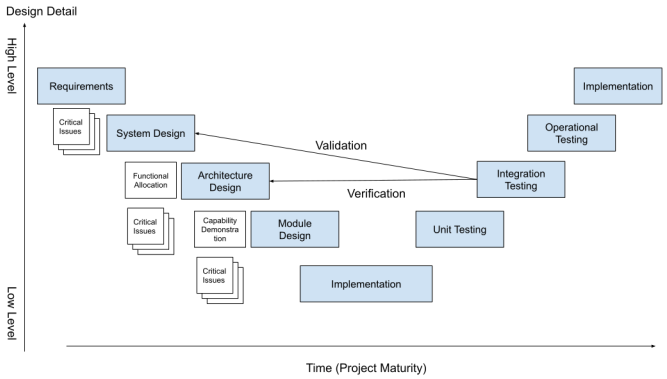

Figure 3.2: A graphical representation of the V-Model [17] (modified from [4])

chosen.

3.2.4 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 [225].

The V model [226] is based heavily on 'the waterfall model' [227] that preceded it but increased it by projecting the project cycle into a three-dimensional space. Fig. 3.2 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 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 [4, 17]. 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 [4] is the identification of 'critical issues'. Capability demonstrations are also important to demonstrate 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

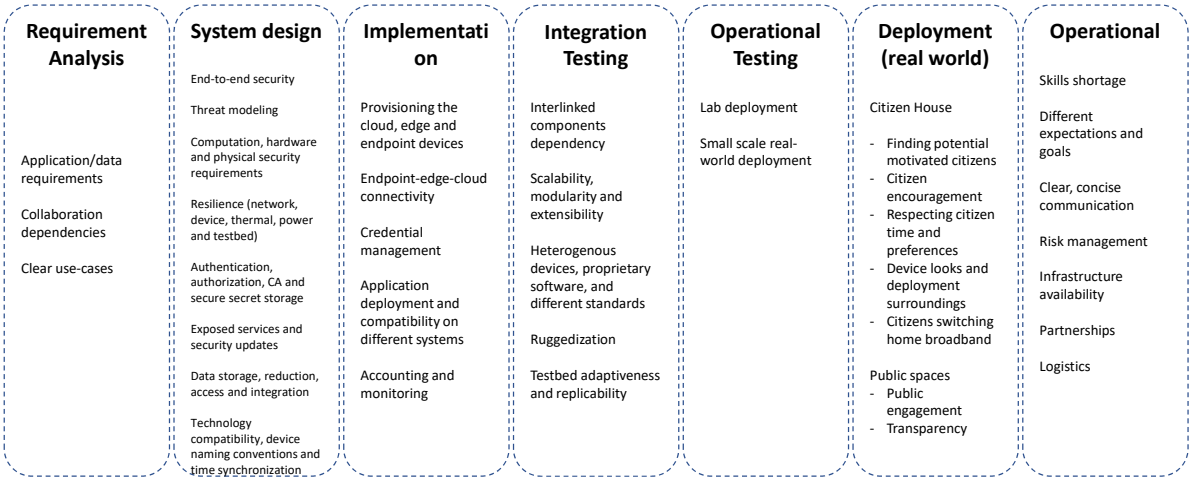

Figure 3.3: Summary of challenges in smart-cities research projects

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.

3.3 Challenges

This section discusses the challenges captured. We use the V-model to classify challenges and map various phases of the research project. Fig. 3.3 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.

3.3.1 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 [228]. 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 [229]. 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 [222].

Once the type of data is clarified, it is essential to consider the relevant General Data Protection Regulation (GDPR) [230] 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) [18].

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.

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 [231] and London Datastore [232]). 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 [222]. 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 [233]. 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

3.3.2 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 § 3.2 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 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 [234]. 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.

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 [235] that could compromise the security of the collected data or the testbed. 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 specify threats, attacks, vulnerabilities, and countermeasures that may impact the testbed infrastructure and its components. It can be performed in five significant threat

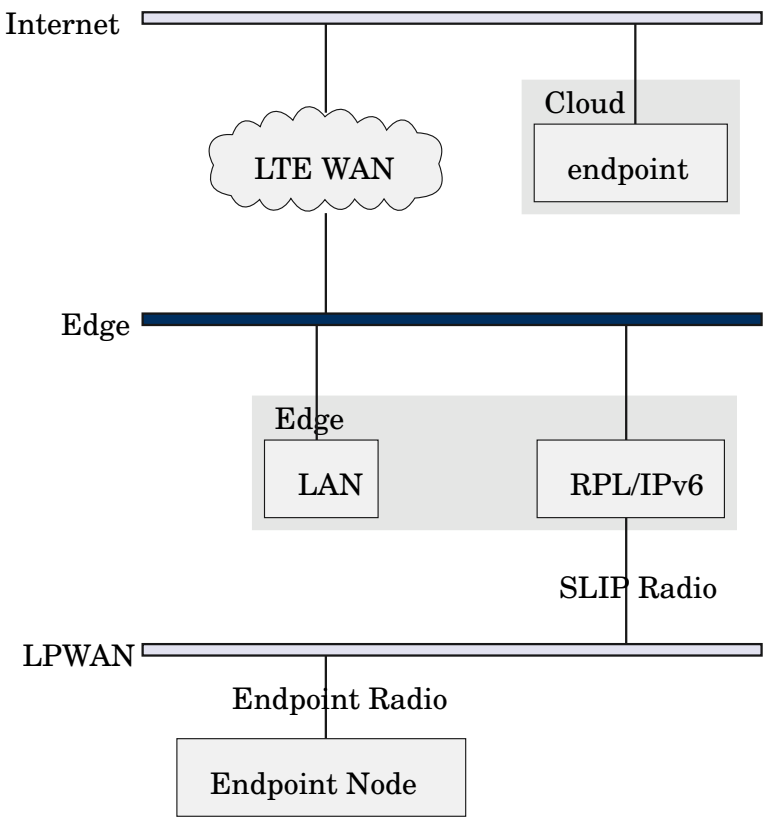

Figure 3.4: Local and remote threat models originate from the bottom up and up to bottom respectively

modelling steps: defining security requirements, creating an infrastructure testbed diagram, identifying threats, mitigating threats, and validating that threats have been mitigated [236–238]. Threat modelling will enable testbed administrators and architects to express the security design of the testbed, interpret those designs for likely security issues, and handle mitigations for security issues.

An example of architectural consideration of threat models for our three-tier approach is presented in Fig. 3.4. 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. 3.4 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 [239]. 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)) [188].

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 [240]. 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 [183, 198, 241]. Furthermore, applications also suffer from latency problems [242] 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 [19] 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 [243]. 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) [194]. 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 [19].

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 [244].

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 [245]. 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 [239], 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 [246]. 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 [191, 245]. 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 [247]. 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 [245]. 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 [186, 196, 248]. The inability to upgrade or configure the firmware remotely means that the code/firmware must be immaculate and comprehensively experimented with, 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 that disturb 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 [249]. 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 [234, 250]. Data reduction and compression methods, such as sending preprocessed data to the edge/cloud tier rather than raw data [185], 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 [250]. 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 [18]. 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 [251]. 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 [187]. 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 [190].

From a data integration perspective, the platform should be able to integrate data streams from multiple heterogeneous data sources [252–255]. Using similar data formats will allow better data interoperability [228, 242, 256]. 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 [18]. For instance, different vendors can have different ways of implementing applications, services, communication, and data streaming API [184] requiring common and standardised APIs [242, 257]. Further, the applications deployed on testbed infrastructure must enable the users to perform a simple, complex query and subscription [258]. Furthermore,

data transfer from the endpoint to the edge to the cloud should be reliable with minimal data loss [193, 198]. 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) [246].

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) [19, 182]. 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) [259].

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 [186, 258]. Also, all devices in each level (cloud, edge and endpoint) must be synchronised in time for data integrity and audit log purposes [201, 248].

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 [260], and how devices at the endpoint, edge, and cloud tier will be managed, provisioned, and communicate with each other [184].

3.3.3 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 [261], Luftdaten [262], SensorTag [263], and Smart Plugs [264]. 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) [265] 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 [189, 196, 229, 248] 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 [201] 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 [201, 229]. Secure transmission protects against packet sniffing, man-in-the-middle attacks, replay attacks, 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. 3.5. Both sensor LPWAN nodes and cloud addressable 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 [266]. Second, connect the device through Demilitarized Zone (DMZ) to the cloud using a VPN [191]. However, in the case of a cyber-incident where an attacker compromises one edge tier device, they can explore and

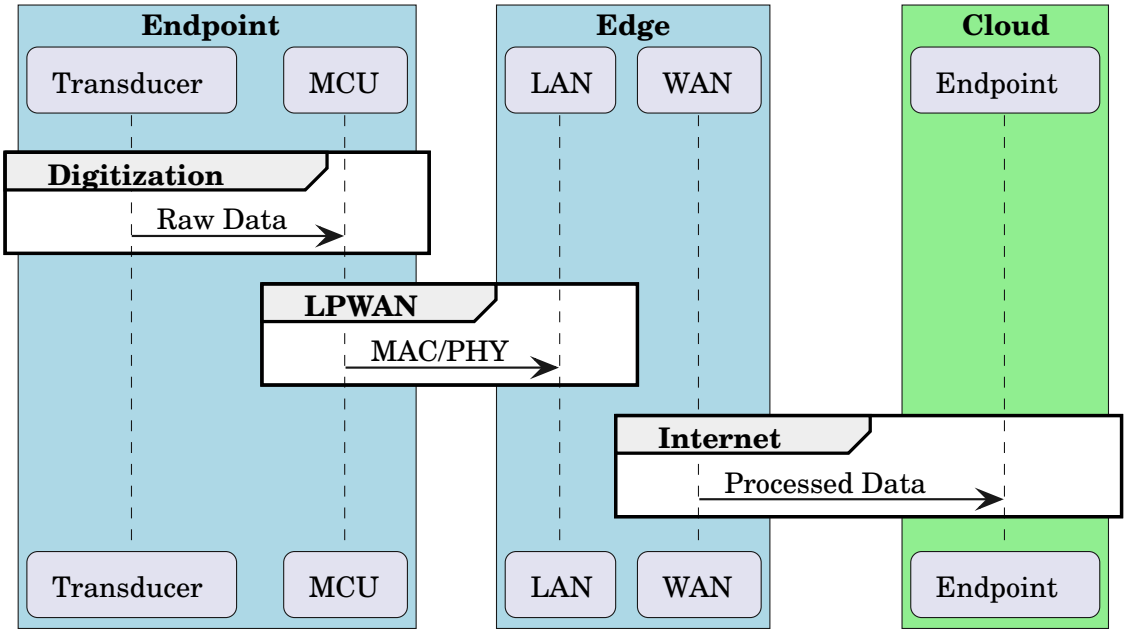

Figure 3.5: Connectivity points between the three tiers for a WSN use case

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 hides the application infrastructure from the Internet, evades network-based attacks (DDoS, ransomware, malware, server scanning) and reduces security risk. However, business 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 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 [228].

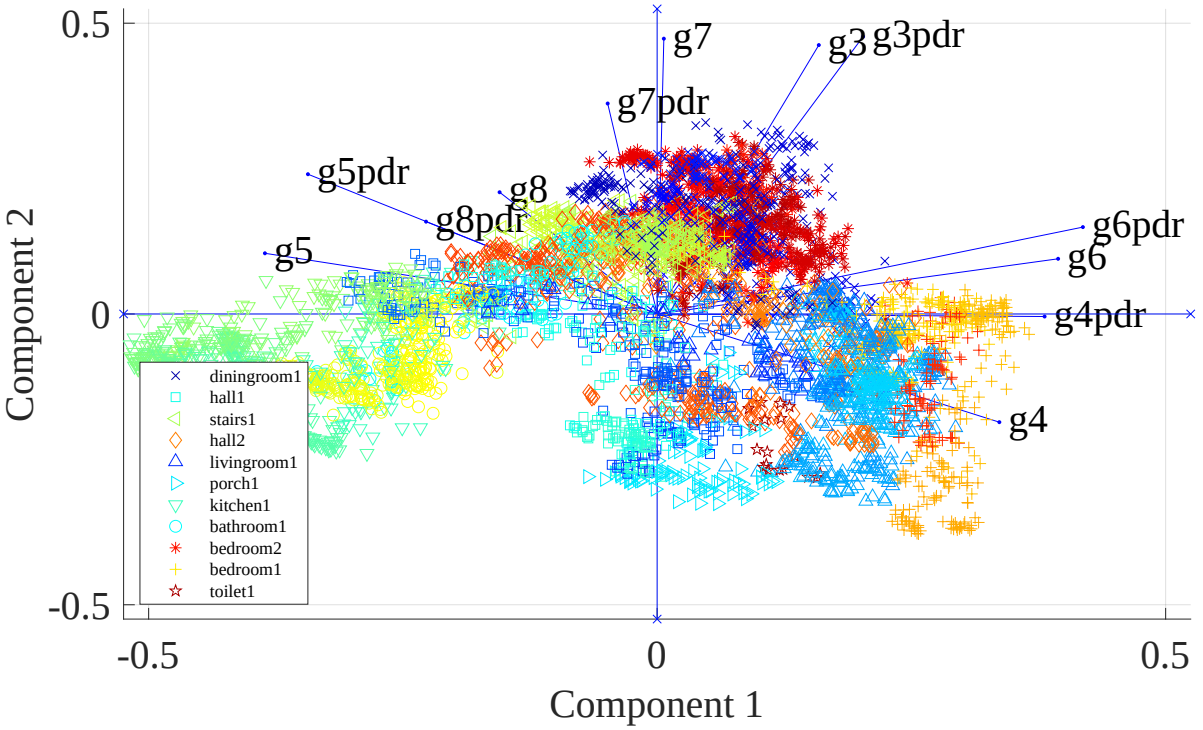

Figure 3.6: Mobility of BLE tags in a house, the association of the PDR and signal strength for eight listening gateways [268]

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 [18, 42]. 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) [188, 267]. 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. 3.6 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. 3.6, 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 [189]. 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 [269]. 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 [270]. Information for the particular network in its formation [247] can be inferred by sniffing those 10 ms timeslots where routing is established [271].

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 [191, 272]. 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 [266]. One preferred way of providing access is to store the administrator's public SSH keys[2] [273] 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

[2]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.

attackers compromised Florida City's water supply using remote access software (TeamViewer), which allowed staff to share screens and troubleshoot IT issues [274] by exploring systems from the Shodan search engine and outdated passwords.

Application deployment and compatibility on different systems

Research projects involve multiple researchers developing different applications (Python/R programs) [246] 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 [275, 276] 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 [198].

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) [251]. 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 [273]. 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 [18, 191, 194, 197, 201, 277]. 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). 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 [201, 248]. For example, SPHERE [18] monitored the status (reachability) of the deployed endpoints by regularly polling various devices within the home network based on Nagios.

3.3.4 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 [278]. 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 [189]. 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 [182]. 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 [186, 193]. For example, research projects that involve energy monitoring deploy multiple devices such as smart plugs [279], Tesla powerwall [280], OpenEnergyMonitoring [281]. 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 [187]. Debugging and finding the misbehaving piece takes considerable time and is challenging [186, 239, 282, 283]. 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 [284]. Additionally, components and their functionality must be well defined and have robustness and resilience built in, saving time for system administrators [18, 201, 228]. It also helps minimise the number and duration of visits to the citizen's residence to repair the system [198]. The maintainability of the infrastructure and the consistency of the interfaces between all different components [202] (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 [18, 187, 193]. 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 [187]. 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) [184]. 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 [228].

Heterogeneous devices, proprietary software, and different standards

Projects can have different devices on edge and endpoints generating various types of data and formats [193, 197, 250, 252, 260]. 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 [193, 197, 260]. 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 [195, 242, 260, 283, 285, 286]. It is vital to use widely open standards and possibly the same standard and format to help reduce learning times for research personnel [184, 201].

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 [248]. It must be packaged in a form that can be securely mounted [19, 194] 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, BIC 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 [19]. It might be a good idea to place the sensors in a Stevenson radiation shield[3] separate from the sealed waterproof electronic enclosure. Furthermore, it is recommended to identify a suitable enclosure

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

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 [229]. 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 [245, 287]. 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.

3.3.5 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 [183, 184].

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 [288] 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 [183].

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) [183, 239]. 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 [19]. 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. 3.7 provides the different teams 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

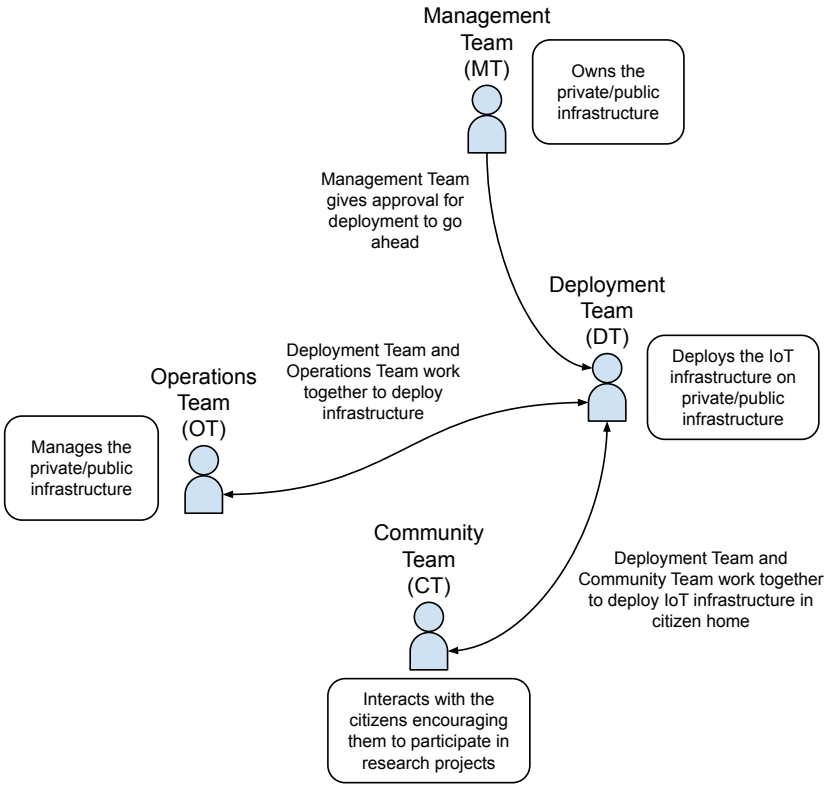

Figure 3.7: Different teams involved in smart city research projects and their relationships

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 [183] 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.

3.3.6 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 [183, 289]. It is also important to mention to whom the device belongs and where to contact for more information [248].

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 [190]. Recruitment works best using various methods, from brochures and social media to door-knocking and face-to-face visits [290]. 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 [290] 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 [177]. In such cases, CT interacts with different community groups through local community centres and social media applications, such as Facebook and Nextdoor [291]. 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, 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 [190]. 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 [198]. 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 [290]. For many people, simply getting involved is a barrier. For example, Twinergy [177] 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 [198]. 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 [290]. During the Twinergy project, one participant decided not to install the technology because it would spoil their minimalist decor.

DT would need the credentials (SSID and password) in case of Wi-Fi connectivity 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 needing to be more comfortable 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 [198].

Furthermore, the endpoint devices must remain placed throughout the deployment without damaging the participant's house (delicate surfaces such as precious antique wood and wallpaper) [197, 251]. 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 [194, 251]. 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 [197]. Therefore, it is essential to carefully identify the location of the device deployment at home. The DT must respect the citizen's house and time [189]. The longer the DT takes at a citizen's home, the more inconvenient it is for the citizen and their regular routine [198]. Home visits of citizens for deployment and maintenance purposes must be highly optimised and efficient with preparation done beforehand [197].

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 [197].

Device looks and deployment surrounding: User convenience, approval, and aesthetics of deployed devices are essential for a successful deployment (especially for wearable endpoints or visible devices) [18, 228, 229]. 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 bother users' sleep or affect user behavior [229, 251]. Furthermore, LEDs also consume a good amount of energy [248]. 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 [182]. 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 [197].

It is also essential to note the device deployment conditions or the surrounding location to understand the sensor readings [251]. For example, a temperature reading in a location with natural sunlight will vary from a temperature reading in the shadow [19]. 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 near the window. Direct sunlight leads to device heating and can affect sensor readings [185]. 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 [248].

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) [199, 266]. 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 [190].

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 [194]. 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) [194]. 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 [186, 198]. 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 [186, 193]. 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 interactions 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 [290]. 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 [290]. 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 [190]. 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 [290] 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) [190]. For example, SPHERE created a 3 min animated video [292] 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 [233].

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 [190].

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 [183]. 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 [233].

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 [199] 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 [197, 198]. 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 (§ 3.3.1) earlier in the project to understand data collection and be transparent with citizens.

3.3.7 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 [222]. Experience and knowledge in system administration, cloud infrastructure, networking, DevOps, and cybersecurity are required [277, 293, 294].

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 [293]. There may be cases where collaboration priorities are different, which can create challenges in communication and work completion. Teamwork is essential for project success [239, 293]. 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 [183].

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 [183]. 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 [295], 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 [193].

Partnerships

It is essential to have the support and partnership of the city council [296]. 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 [32], Belfast Agenda [31], Chicago Technology plan [297]) 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 [231], London Datastore [232]).

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 [233].

3.4 Conclusion

The second chapter provided the role of IoT and digitalisation in improving urban society at various levels and how urban sensing helps cities become more innovative, sustainable, and resilient. It identified and categorised what types of data can be collected from a citizen's perspective using readily available and accessible IoT devices and examined how this information can be used in city decision making and management, with all its caveats.

However, we need to deploy IoT infrastructure to collect the above data at different levels. The continued growth of wireless technologies has resulted in significant research into urban monitoring through data-gathering IoT testbeds. These research testbeds follow a typical three-tier architecture and many designs and implementation challenges. Challenges are associated

Table 3.3
Summary of the challenges

Phases	**Challenges**	**Remarks**
Requirement Analysis	1. Application/data requirements 2. Collaboration dependencies 3. Clear use-cases	1. Understand data collection requirement to meet project objectives and expected results outputs (impacts, visualisation). 2. Collaboration between different departments (IT support, estate teams) /partners(universities, city council, industries). 3. Develop use-cases that address the requirements.
System Design	1. End-to-End Security 2. Threat Modeling 3. Computation, hardware and physical security requirements 4. Resilience (network, device, thermal, power and testbed) 5. Authentication, authorization, CA and secure secret storage 6. Exposed services and security updates 7. Data storage, reduction, access and integration 8. Technology compatibility, device naming conventions and time synchronization	1. Security of all devices at each tier and the communication between them, including physical and data security. 2. Specify threats, attacks, vulnerabilities, and countermeasures that may impact the testbed infrastructure and components. 3. Determine computation capabilities (memory, storage, CPU) at each tier. Hardware and physical security. 4. Handle network loss and latency issues. Monitor health of devices. Safely provide power to edge and endpoint devices. Automation. 5. Proper authentication, authorization allowing trusted users to access services. Secure credential storage 6. Services and system must be made secure by default. No default password. Applications, devices must be securely configured. OTA. 7. Data must be encrypted at all tiers in transit and rest. Data owner to ensure data validity, quality, secure storage, access, replication. 8. Support standard libraries, packages and control interfaces. Interoperability matrix. Good naming convention.
Implementation	1. Provisioning the cloud, edge and endpoint devices 2. Endpoint-edge-cloud connectivity 3. Credential management 4. Application deployment and compatibility on different systems 5. Accounting and monitoring	1. Installation and configuration of VMs (cloud), SD/HDD/eMMC (edge), resource constrained devices. 2. Encrypted protocols for communication from endpoint to edge to cloud. 3. Secure access to cloud, edge and endpoint devices for configuration, maintenance. 4. Develop and deploy different applications on different devices with different architecture. 5. Maintain a inventory of number of devices with hardware and software details. Access logs of users-device access. Alert mechanism.
Integration Testing	1. Interlinked components dependency 2. Scalability, modularlity and extensiblity 3. Heterogeneous devices, proprietary software and different standards 4. Ruggedization 5. Testbed adaptiveness and replicability	1. Ensure data transfer via multiple interdependent components and interfaces installed. Perform regular automated integration tests. 2. Scale to tens/thousands of homes/devices in reliable manner. Adapt to ever-evolving technology. Future proof hardware. 3. Collect data from heterogenous devices, open-source or proprietary software. 4. Ingress protection and electrical testing, CE rating. 5. Testbed adaptive to project requirements or community demand.
Operational Testing	1. Lab Deployment 2. Small scale real-world deployment	1. Ensure system is working as a whole and securely sending the data from endpoint to cloud. Provisioning budget 2. Deploy small-scale real-world deployment to understand real-world challenges and build confidence with infrastructure owners.
Deployment (real world)	1. Citizen House 2. Public Spaces	1. Citizen recruitment and engagement, build citizen interest, respecting citizen time and preferences. Device looks and deployment surrounding. 2. Ensure device deployment does not hinder functioning of city infrastructure
Operational Challenges	1. Skills shortage 2. Different expectations and goals 3. Clear, concise communication 4. Risk management 5. Infrastructure availability 6. Partnerships 7. Logistics	1. Difficult to find people with appropriate skill set (system architect/DevOps) in urban monitoring research projects. 2. Each partner has different set of expertise, business models, expectations and priorities. Team work is essential for project success. 3. Good practice to invite only a few key people or technical leads for clear and concise communication. Face to face meetings are better over online discussions (especially brainstorming sessions). 4. Important to have risk management that may affect the project schedule. 5. Expect inevitable situations such as infrastructure outage, internet outage, devices affected by weather, delays in importing devices from other countries. 6. Important to have support and win the confidence of city departments, employees, city councils by involving them in project. 7. Deployment team should be aware of node designs, installation procedures, deployment locations and able to make decisions on the fly, create communication channels with public.

with developing, deploying, and maintaining the infrastructure required to collect, process, and analyse data and provide insight to citizens, policy makers, and governments.

In this chapter, we discuss these challenges by considering several projects on real-world IoT testbeds. The author analysed the projects and classified them in the context of the development phases of the V model and organised them by the requirements analysis, system design, implementation, testing, and deployment phases. The listed challenges will hopefully help other urban monitoring researchers plan future testbeds that will prove valuable and reduce the design and implementation costs of these projects. Table 3.3 provides a summary of the challenges presented in a chapter.

Resolving the challenges faced in the deployment of IoT infrastructure for an intelligent city research project is essential. Some challenges are project-dependent and cannot be solved by technology. For example, it is not easy to resolve any challenges by technology in the requirement analysis (primarily project-dependent), deployment in the real world (depends on citizen preferences and communication between different stakeholders), and operational challenges (communication and expectation-based). The next chapter (chapter 4) aims to solve the challenges that technology can solve to reduce the implementation and maintenance time needed to deploy the IoT infrastructure required for a smart city research project.

The author provided the challenges that a smart city research project faces in deploying IoT infrastructure that collects urban data (RQ3.1 - § 3.3) and classified the above challenges in different phases of research projects to help future smart city projects (RQ3.2 - § 3.3 and § 3.2.4). Contribution to the knowledge in this chapter (C2) is the systematic review of the challenges faced in designing, implementing, and deploying IoT infrastructure that a smart city research project faces that collects urban data and classification of them under different phases of research projects to help future smart city projects.

CHAPTER 4

A SMART-CITY FRAMEWORK FOR SHARING IOT INFRASTRUCTURE BETWEEN MULTIPLE RESEARCH PROJECTS AND ORGANISATIONS[1]

Government and city councils are exploring innovative and efficient approaches to tackle challenges such as urbanisation, climate change, and net zero goals. Research organizations (RO), in collaboration with the city councils, participate in multiple smart-city research project (SCRP) to solve the above challenges. In chapter 2, we introduced the different types of data that can be collected at personal, building, district and urban levels and the benefits derived from it. SCRP ranges from structural health monitoring for a bridge [152] to noise and air pollution monitoring [298] to citizen health monitoring [18] or smart electricity use [50, 161] to become carbon neutral and meet net-zero standards.

Most SCRP involves the deployment of the IoT infrastructure in three tiers, including endpoints (sensors that measure the physical environment), edge gateway (collect and process data from endpoints), and cloud (collect and process data from endpoints/edge). On the other hand, other organisations also deploy similar three-tier (Cloud-Edge-Endpoint) or two-tier (Cloud-Edge) architectures in the city. For example, in collaboration with an advertising company, BT has installed free Wi-Fi InLinkUK kiosks in the public area [299]. Transport (bus companies) display bus estimates at bus stops using a bus display system [300]. Managing the IoT infrastructure to support the SCRP or city infrastructure is challenging. Challenges (chapter 3) include security and management of multiple devices, data security and privacy, user privacy controls, visualisation, multitenancy of applications, and network resilience [301]. Although each SCRP implements the IoT infrastructure differently, depending on the project requirements, usability,

[1]This work is submitted to IEEE Communications Surveys & Tutorials entitled "SMARF: A Smart-City Research Project Framework for Sharing IoT Infrastructure among Research Projects from different Organisations". The first author wrote the abovementioned paper and proposed and implemented the ideas/approaches, designs, and experiments. The other authors provided their valuable reviews and suggestions to improve the paper.

budget, time, and technical skillset, the deployed IoT infrastructure is often quite similar. There is a similarity in the required services, such as the deployment of applications, data storage, analysis, and visualisation. The core subset of infrastructure needed to meet the research requirements often includes hardware (CPUs, RAM, HDD/SDD) and capabilities (data storage, authentication, analysis, provisioning). Based on the requirements of the projects, hardware and capabilities can differ.

In addition, local communities aim to solve the problems faced by citizens and may require similar services such as data storage, analysis and visualisation. For example, although pedestrianising streets creates more livable neighbourhoods and maximises community enjoyment, it also creates concerns among citizens and shop owners about late-night noise pollution (from young university students and restaurants) and reduced footfall (because of parking problems), respectively. They require sensors, internet connectivity, edge devices (local data storage), and cloud tier (data storage, analysis, and visualisation) to provide evidence to the council and voice their concerns to the city council.

Suppose there is a way to share the infrastructure on the cloud and edge tier between multiple SCRP, organisations, and local communities. In that case, it can create new services and speed up the implementation time for SCRP and local community projects. Sharing infrastructure reduces deployment costs and allows multiple parties (private organisations, community support groups, and individual citizens) to deploy their solutions and resolve public issues. It also adds challenges in data ownership, data management, ownership of devices, and access to the data. It also brings cyber security issues related to infrastructure and management, requiring coordination and collaboration with different people. However, before the infrastructure can be shared, it needs to be secure and reliable and resolve the challenges of maintaining a single non-shared infrastructure. The motivation of this chapter is to solve the challenges of deploying and maintaining an IoT infrastructure for a single SCRP and then explore the possibility of sharing that infrastructure between multiple SCRPs and organisations for efficient resource utilisation. Based on the above objective, we devise our research questions.

The chapter is organised as follows: § 4.1 provides the research questions and our approach. § 4.2 provides background knowledge on the smart cities' IoT infrastructure and the requirements and challenges of implementing an IoT infrastructure to collect urban data. § 4.3 introduces the smart city framework that can be used to implement a three-tier infrastructure. § 4.4 provides details on how the smart city framework can be implemented using open-source components. § 4.5 provide details on how the above infrastructure can be shared between different owners. § 4.6 and § 4.7 provide implementation and evaluation of smart city framework. § 4.8 briefly introduces different projects with similar architecture and compares the smart city framework with them. § 4.9 concludes the chapter.

4.1 Research Questions and Approach

Research Questions

The chapter provides a smart city framework containing modules and sub-modules that aim to solve the requirements of a smart city platform and the challenges faced in implementing a smart city research project. It also explores the opportunity to share the IoT infrastructure between multiple organisations and smart city research projects to save resources and reduce costs.

This chapter attempts to investigate the following research questions:

RQ4.1 Given the challenges explored in the chapter 3, what could be a solution that can solve the smart city software requirements and the challenges faced in a smart city research project?

RQ4.2 Can we share the above smart city infrastructure between multiple organisations and smart city research projects to use the resources and reduce costs efficiently?

RQ4.3 What are the other solutions that exist, and how are they compared to our work?

Research Approach

To answer the first research question (RQ4.1), the author reviewed the software requirements of smart cities and the challenges faced in implementing smart city research projects (§ 4.2.2) and presents a smart city framework (§ 4.3) and its implementation using open-source components (§ 4.4). The author explored possible stakeholders and owners (§ 4.5) and how framework components can be shared between different stakeholders to answer the second research question (RQ4.2). For the third research question (RQ4.3), the author reviewed the IoT reference architecture and the data gathering framework provided in the research literature (§ 4.7) and checked if they can be used to solve the challenges faced or implement a smart cities project. The author reviewed different research papers that mentioned the “smart city framework” [8, 302–317], “data gathering framework” [318–324] and “IoT reference architecture” [325–327] using Google scholar to understand the similarity and differences and to what degree they solve the problem compared to our work.

4.2 Background

This section briefly explains the stakeholders and public support for IoT infrastructure for a smart city, the software requirements for a smart city platform, and the challenges in maintaining such infrastructure.

4.2.1 IoT Infrastructure for a Smart City: Stakeholders and Public Support

SCRP's different stakeholders

Multiple stakeholders could be interested in SCRP. First, end users (citizens, city councils, and governments) mainly want to understand the data and its impact and benefits and prefer to view it in a visualised form, which might require only read access to the data. For example, the government could need data to release reports, comply with EU regulations (air pollution standards), and ensure lockdown measures for COVID-19 and others. Second, data scientists (Jean Golding Institute [328] and other research partners) may want to analyse the data using data analysis tools (python, jupyterhub, tableau) and provide results. They might only require data read access and computational capabilities (GPU for AI/ML). Third, application developers (students/Ph.D./RA) who want to develop and deploy applications to collect/analyse data and perform noise, air quality, dampness, and transport analysis. There may be other collaborators or interested parties (city councils, transportation and waste collection agencies, and others) who want to add their edge devices and collect data. For example, a bus transport agency may add additional hardware (cameras) and be interested in monitoring the number of people at a bus stop to predict demand and provide better services.

Public support for IoT Infrastructure

The concept of smart cities and granular urban data has been present for decades; however, it has not fully matured due to the challenges of maintaining the IoT infrastructure and financial challenges. SCRPs that involve urban data collection often fall under the management of city councils and governments. However, city councils often have more significant problems (tackling homelessness, crime, essential services) to resolve than implementing and managing infrastructure to gather urban data. They often release tenders and assign a company to deploy the infrastructure to resolve problems (smart lighting, waste collection improvement) or meet government directives (such as the 2008 Ambient Air Quality Directive (2008/50/EC)). Most SCRPs are often a collaboration between multiple entities (universities, industry, city councils, community centres) and are partly or fully funded by different government schemes or innovation funds.

For example, the EU and the Cantabria government funded Smart Santander [223], a collaboration between 15 partners from the public sector, enterprises, universities and research centres. Similarly, the AoT [19], and SAGE [329] projects are funded by the National Science Foundation, the Chicago Innovation Fund, the University of Chicago, and Argonne National Laboratory. It is a collaborative effort among scientists, universities, federal and local governments, industry partners, and communities. REPLICATE [26] is funded by the EU's Horizon 2020 Research and Innovation Programme. UMBRELLA [20] is a joint project between South Gloucestershire Council and Toshiba, with the support of the West of England Combined Authority and the Local

Enterprise Partnership. In summary, SCRP often runs for a few years and requires multiple collaborators, funders, and support to succeed. The author consider an SCRP to be a subset of a smart city. Multiple SCRPs together make a city smart. The author use the Smart City Research Project (SCRP) and smart city platforms interchangeably.

4.2.2 Smart City Platform: Requirements and Challenges

Implementing a software platform for SCRP requires understanding the requirements of a smart city platform and the challenges faced in implementation.

Software requirements for smart cities platform

Software requirements for a smart cities platform can be categorized into functional and non-functional requirements summarized below. First, we do provide from where we understood the requirements and then summarize them. Santana et al. [302] surveyed multiple smart cities projects in the area of Cyber-Physical Systems, IoT, Big Data and cloud computing and provided functional and non-functional requirements that a software platform for smart cities should meet. Similarly, Ribeiro et al. [330] analysed multiple data integration systems for smart cities and provided functional and non-functional requirements for data integration software platforms. The author also reviewed the user's requirements voiced during Questions and Answers and Panel Discussions of DAFNI in multiple road shows and events [331]. DAFNI [332] is a UK national facility to advance infrastructure system research. It allows researchers to host national infrastructure datasets, run models (user code packed in docker containers) and provide impactful visualisation. Although DAFNI is HPC platform and differs from most smart city research projects that collect urban data, few requirements are similar regarding data storage, processing and visualisation. The DAFNI user's concerns and requirements help us to understand software requirements for a smart city platform because of similar requirements. The author was also part of the UKCRIC Bristol Infrastructure Collaboratory (UBIC) [333] collaborates with local communities to solve citizen challenges in multiple areas. The areas includes structural health monitoring (Clifton Suspension Bridge), energy monitoring (Bristol Community Energy Campus), moving in shared spaces, water and built environment monitoring, and citizen sensing.

Functional Requirements

Functional requirements include data management (collection, storage, analysis, visualisation), data processing (analyse, verify, aggregate, and filter large datasets), external data access (provide API to access the data), data ingestion (importing real-time/batch data), metadata management (managing information about the data).

It also included capabilities for application run-time (deploy applications), service management (create new services or applications), software engineering tools (for the development and maintenance of services and application), WSN management (adding, removing, and monitoring

sensors and actuators), the definition of a city model (create models to understand city data) included, machine learning capabilities.

Regarding features, DAFNI [332] users requested more automation and asked questions such as *"Can the user develop machine learning models within the platform?; Can the user get messages when the job is complete, or the user needs to log on to the platform and check?; How can the user get results from the platform and feed them to the end-user-facing dashboard?; Users would appreciate git integration.; How can the user trace output to specific versions of code and data used to generate it?; Does the platform manage dynamic data, and is it possible to pass the output of multiple models to a single model?"* Regarding technical support, users requested information about specific training, pre-requisite skills, the knowledge required by users and help and support provided. UBIC [333] individual projects require infrastructure deployed in citizens' houses and public infrastructure to collect, analyse, and visualise the data. As there are multiple projects, sharing the infrastructure between multiple projects is essential without compromising data security and reliability.

Non-functional requirements

Non-funtional requirements interoperability (different devices, systems, applications, and platforms), scalability (users, data, and service can increase over time), security, privacy, context awareness, extensibility, and configurability, availability, security, and privacy. In cybersecurity, DAFNI [332] users were concerned about the platform's security and data hosted and asked questions such as *"How is security being addressed; how often are systems and networks pentested; Where is data stored? Encryption at rest/in flight?"*.

Challenges in deploying smart cities research projects

In chapter 3, the author analysed multiple SCRPs, and lessons learnt research articles, added their experience in deploying multiple SCRPs, and provided the challenges faced in implementing the IoT infrastructure for SCRPs. They categorised challenges into multiple phases of SCRP, from understanding project requirements (requirement analysis) to designing how to fulfil those requirements (system design) and setting up defined infrastructure (implementation) to ensuring that different infrastructure components work together (integration) and tested in the lab and initial small-scale deployment (operational testing), followed by deployment in the real world and operational challenges.

There are challenges that are project-dependent and cannot be solved by technology. For example, it is not easy to resolve any challenges by technology in the requirement analysis (primarily project-dependent), implementation in the real world (depending on citizen preferences and communication between different stakeholders), and operational challenges (communication and expectation-based). On the other hand, there are challenges that can be solved using technology and can be provided with a readily available solution.

The system design phase provides challenges around the platform's end-to-end security,

resilience, and data (storage, reduction, access, and integration). The platform comprises multiple components (software and hardware) that require compatibility. From a security perspective, software components expose multiple services (Webservers, MQTT, APIs) and often contain a security vulnerability. Software components require regular updates, backup, rollback, and hardening without disrupting the user experience and ensuring a secure platform. The platform should provide confidentiality, integrity, availability, non-repudiation, and a chain of trust. The physical and hardware infrastructure security required depends on the use cases of the SCRP. Less device security may be acceptable for a SCRP only collecting air pollution data; however, for a SCRP collecting citizens' health data (blood pressure, heart rate), devices should provide adequate hardware security. The device should ideally use security features (such as encrypted filesystem, secure boot, TPM and secure element (SE)). The platform must support multiple stakeholders (citizens, data scientists, application developers, and others who interact with the infrastructure), multiple SCRP use cases, and be application-agnostic. It can have resilience (network, device, thermal, power) built based on the use cases. Supporting multiple stakeholders, devices, and applications brings authentication, authorisation, certificate authority, and secure secret storage challenges. There are also challenges around data storage, reduction, access, and integration.

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 integration phase challenges include ensuring that the platform is scalable, modular, extensible, adaptive, reproducible, and supports heterogeneous devices, proprietary software, and different standards.

In summary, the SCRP platform can have multiple stakeholders. The platform should be able to collect urban data from multiple sources securely and reliably. The platform should provide services to securely develop and deploy urban data monitoring applications to multiple devices, process, analyse, visualise, and store the data on the cloud, edge, or endpoint tier. The platform should also be able to collect a large amount of data (thousands of devices sending ten thousand messages/per min) and process them reliably and securely. The SCRP platform requires a technology stack at each level to support functionalities, requirements, hardware, and multiple stakeholders. In addition, the platform should be able to provision multiple devices at the cloud, edge and endpoint tiers and provide end-to-end security. It must be scalable, modular, extensible, adaptable, and replicable.

Our work takes the challenges mentioned in chapter 3 (summarised in § 4.2.2), and requirements mentioned by [302, 330] (§ 4.2.2) and UBIC requirements as a reference and aim to resolve the challenges solvable by technology to reduce the implementation and maintenance time for deploying the IoT infrastructure required for a SCRP.

4.3 Smart City Framework

Fig. 4.1 provides a framework to deploy multiple devices in the cloud, edge, and endpoint tier, collect urban data, and provide capabilities to store, analyse, visualise, and collect insights to solve urban challenges. The smart city research project can often have a three-tier architecture: Cloud, edge and endpoint. In our smart city framework, we only consider cloud and edge tier. The author categorise the SCRP platform into three main components or modules **(i)** Infrastructure management, **(ii)** Data management **(iii)** Application management.

The infrastructure management module manages and operates the SCRP platform. The data management module ensures that data is ingested from all sorts of devices with any data rate, stored on the databases, analysed and visualised to gather insights. The application management module can host the custom system or user applications deployed on the cloud, edge or endpoint tier.

The platform should be pluggable, scalable, and adaptable and support different research projects with additional requirements. Modules can be required and optional in the smart city framework or implemented platform depending on a SCRP requirements and use cases. For example, if there exists a SCRP gathering only air pollution data with a single stakeholder, a module such as provisioning, data storage, analysis and visualisation are required, and other modules, such as authentication and security components, can be optional. On the other hand, for a project that collects noise and air pollution and performs video analytics at the edge with the deployment of multiple applications at the edge and devices at the endpoint with various stakeholders, all the components mentioned (Fig. 4.1) in the cloud level are required.

4.3.1 Infrastructure Management

A SCRP platform can include multiple devices on the cloud tier (servers), edge (SBCs/gateways) and endpoints (sensors that detect environmental parameters). Infrastructure management includes device provisioning at all tiers, authentication and authorisation (users, devices, application), security and logging of the platform, providing storage as a service, application development, deployment, and orchestration on different devices, and communication platform to the platform's stakeholders. It also includes the setup and configuration of any data management module, such as the installation and configuration of databases, Machine Learning (ML) frameworks, or other applications. The infrastructure management module contains multiple sub-modules to support the infrastructure operations such as provisioning, authentication and authorisation, CI/CD, orchestration, storage, security, logging and monitoring, and communication.

Provisioning

Multiple devices can be deployed at each tier depending on the use cases of SCRP. On the cloud, the number of servers depends on the services required to support the edge and endpoint tier and

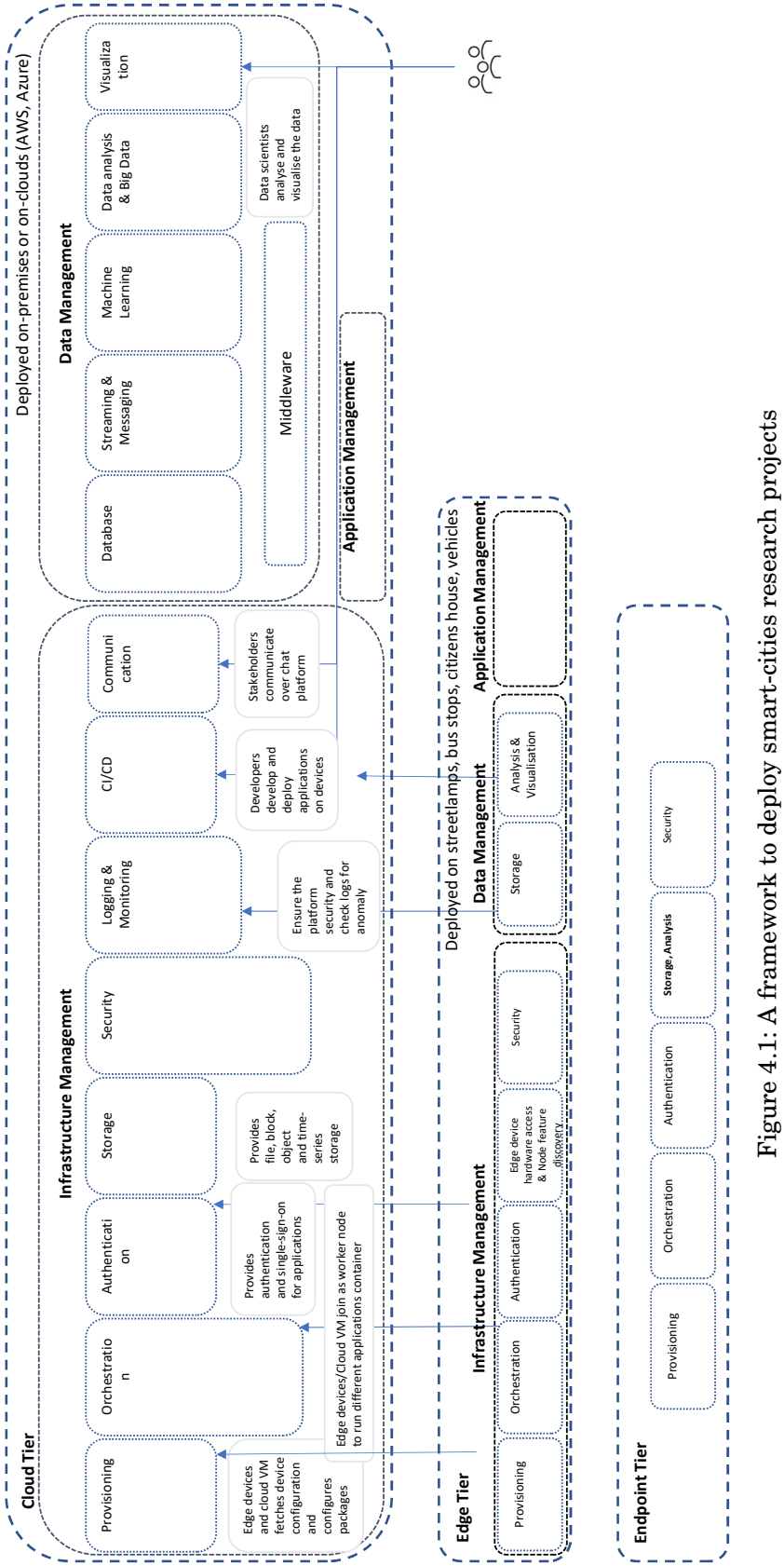

Figure 4.1: A framework to deploy smart-cities research projects

usually ranges from one to ten. Installing and configuring a server on the cloud level and devices on the edge level is tedious. It requires installing OS applications, configuring them securely, and configuring hardware allocation (e.g. RAM, CPUs, GPU passthrough). The edge device requires stable and secure connectivity to the cloud tier. The number of edge devices depends on the SCRP use case sample size (the number of houses or streetlamps) and can range from one to hundreds). Endpoint tier devices are usually resource-constrained devices, such as SCK [261], Luftdaten [262], SensorTag [263], and Smart Plugs [264]. Endpoints are usually connected to the smart home platform or the edge device. The provisioning of endpoint devices (or management of WSN) depends on the device's capabilities and the communication medium between the endpoint, the edge, and the cloud. It primarily includes configurations such as setting up connectivity (using Wi-Fi/ZigBee/ 802.15.4), the server address to publish the sensor data, and setting the time on the endpoint. 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 edge network, the encryption keys to encrypt the data, the network address/port number of the destination, and other settings. The platform requires a provisioning module to ensure the configuration of devices, servers, and different services.

Authentication and Authorisation

Testbeds consist of multiple devices and numerous applications running in 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 [246]. Conventional user management (creating user accounts for each application and granting privileges to users) is not scalable as the number of applications and user accounts grows with time. Endpoint devices must also be authenticated to join the edge network to send messages. The platform needs a software component to handle authentication and authorisation that would store the user and group credentials and allow users to log into different applications and hardware devices with preferred single-sign-on capability.

Continuous Integration (CI)/Continuous Development (CD)

Research projects and smart city platforms involve multiple researchers/organisations that develop different applications (Python/R programs) [246]. The platform should provide an application development and deployment platform for developers to develop urban applications to collect and analyse data. DevOps combines the best software development and engineering practices, quality assurance, and IT operations, including development practices for design, coding, quality testing, and risk management [334]. It requires a software component to support application development (with CI/CD) and orchestrate applications (including AI/ML) on the cloud, edge, and endpoints. The applications should also be able to access the edge device hardware (such as

cameras and radio modules). The edge device hardware (proprietary/open-source) must provide a interface to interact with it such as “dev/ttyXX“ or “COM“ ports or Real Time Streaming Protocol (RTSP) stream or a open port.

Orchestration

The applications developed by the researchers and application developers must be deployed and orchestrated on the cloud and edge devices with different architectures (arm64, amd64, armhf). The applications must also be able to access the device hardware (sensors, cameras, GPU) on the edge and cloud for data storage, analysis, and visualisation. The platform also needs to identify the nodes' different features, enabling the selection of nodes for application deployment. For example, an application is only deployed on devices with 1GB of RAM, a camera, or a node with specific CPU and GPU access. Applications (TinyML models) should also be deployed on devices in the endpoint tier. The platform must provide a module to handle the orchestration of applications on different devices.

Storage

Research projects collect different types of data from multiple devices and sources. Various applications also require data storage services to store, analyse, and visualise data. Applications can require different types of data storage, such as file (shared filesystem between multiple applications), block (block storage for applications), and object storage (large amounts of unstructured data). Storage should be distributed and provide self-managing, self-scaling, and self-healing capabilities. Furthermore, storage services must be able to encrypt the database and ensure different data protection and research data ethics. A data owner provides data validity, quality, secure storage, access and maintenance, replication, processing, backup, and deletion policy.

Security

Security of the smart city platform is essential. The security module can provide different sub-modules. First, the communication security sub-module ensures secure communication between cloud, edge, and endpoint tiers. Second, the credentials security sub-module ensures the secure storage of credentials used in multiple applications and platforms. As the testbed hosts different services (such as web servers, WebSockets, and authentication servers), the module can also have a Certificate Authority (CA) that can be used to create public-private keys and sign the certificates. Users and devices can trust CA to ensure data transmission. Third, the platform security sub-module can ensure the platform's security, device and application security updates, and application code security.

Logging and Monitoring

A SCRP can have multiple interdependent components and interfaces installed on devices to ensure data transfer from the endpoint to the cloud. When deployed in the real world, there is a probability that a system component will not work in the desired way due to a hardware or software fault [187]. Debugging and finding the misbehaving piece takes considerable time and is challenging [186, 239, 282, 283]. It requires detailed logging from different system components with timestamps, understanding what triggered the logs, and ensuring that log messages represent various failures. Therefore, performing regular automated integration and end-to-end testing is essential to prevent such failures [284]. Therefore, logging from different platform components (application, security, authentication, device logs) is required to detect any issues or anomalies, ensure security, and support application deployment. The monitoring infrastructure is essential to ensure that the hardware and software on the platform work and are operational [18, 191, 194, 197, 201, 277]. Monitoring includes detecting whether devices are reachable and sending regular data. It also includes checking the health of the infrastructure components (RAM/CPU/disk usage, system overload). Endpoints connected via 802.15.4 can include statistics about energy (battery), network (number of data/control packets, acknowledged packets), neighbourhood statistics (list of neighbour nodes and link quality), per-channel per-neighbour packet reception rates, TSCH time synchronisation performance, background noise RSSI levels, stack usage, and others [201, 248]. The monitoring infrastructure should include an effective alert mechanism (email, slack, text messages).

Communication

The smart-city platform can have multiple stakeholders. It is required to provide a communication platform to communicate with each other during the research project (email, chat platform), including audio/video and meeting capabilities.

4.3.2 Data Management

Data collected via the platform can range from Megabytes (MB) to Gigabytes (GB)s to Petabytes (PB), coming from various sources with different frequencies. The platform should provide the ability to store, analyse and visualise the data using big data analysis and visualisation tools and provide an API to enable stakeholders to slice and dice the data for external access and create applications. The data management module contains multiple submodules to support data management operations such as database, streaming and messaging, machine learning, data analysis and big data, visualisation and middleware.

Database

The data coming from different sources can range from raw, time-series data, images, blobs, videos and others. The platform should be able to support all types of data and store them safely, securely and reliably in a database.

Streaming and Messaging

Smart city use cases can have a high-traffic flow of messages that could reach millions daily. There are different use cases for Message Queue (MQ), such as multi-stage pipeline processing (messages processed in a sequence between the different services), message stream (data streamed from many data sources and processed in the pipeline, databases, storage, machine learning, and many other approaches), pub/sub real-time (a smaller number of producers need to send a message to a larger number of consumers), and application decoupling (API, databases, and storage devices would act as a router to send messages to the consumers). The platform must deploy a MQ system to manage such traffic flow. It enables a single focal point of communication. It ensures that each service communicates with the MQ broker in its language, which is delivered to the services waiting for them. The platform needs a module to provide streaming and messaging capability for different applications and use cases.

Machine Learning

ML enables a system to learn from data and undergo iterative improvement without direct human control. ML models can provide actionable insights from live data, make predictions, categorise unsorted datasets, spot trends and changes in the datasets, and predict outcomes. Deploying ML models involves a range of challenges that a platform must meet to achieve effective integration. The challenges include scaling and monitoring model performance, understanding data flow, and communicating with different teams with different specialisations and skills. MLOps combines data science, data engineering and more standard DevOps methods and manages the machine learning lifecycle, from the initial data exploration and preparation, the training, and tuning of the model, to the deployment and continuous maintenance [335]. The platform should provide the ability to train and test ML models on the collected data.

Data analysis and Big Data

Big data analysis involves collecting, processing, cleaning, and investigating extensive datasets to support organisations in operationalising their big data. Stakeholders can use big data analytics to transform terabytes of data into actionable insights that uncover trends, patterns, and correlations in large amounts of raw data to help make data-informed decisions [336]. The platform must provide capabilities to support big data applications and data processing.

Visualisation

Data insights generated using data, big data, and ML analysis must be presented to stakeholders (citizens, researchers, policymakers and government) in beautiful graphics and data visuals. The platform must provide capabilities to support different visualisation platforms. Additionally, visualisation can be done on the cloud layer or locally on the edge layer. For example, suppose an edge device is deployed at a citizen's house. In that case, the citizen can choose not to send the data to the cloud, in which case the edge device can be configured to store and visualise the data at the edge.

Middleware

Collected data can also contain a significant amount of metadata describing the data's details. Adding metadata and context information helps to understand the data and provides valuable insight. For example, outdoor data (vehicle traffic, pedestrian traffic, light levels, weather, atmospheric conditions) can be compared with indoor air pollution data and provide context [233]. The device deployment or surrounding conditions can help to understand sensor readings. For example, a temperature reading in an area with direct sunlight will differ from a temperature reading in the shade [19], or sensor readings of a device near an intersection, highway, garbage can, or recycling centre will differ from each other. A middleware component is required to provide and manage data context information, perform updates, and provide access to context information. It must also offer context data/API management, publication, monetization, processing, analysis, and visualisation of context information.

4.3.3 Application Management

The infrastructure and data management module provide a wide variety of applications that can be used to solve smart city challenges. On the other hand, the platform should provide stakeholders with the ability to create a custom system/user applications running on the cloud/edge tier to support bespoke requirements.

4.4 Implementing the Smart City Framework

At first, the author want to develop and deploy the above framework (§ 4.3) using a technology stack, enabling the basic functionalities and improving usability, security, and multi-tenancy. The platform aims to use open-source software components and features, enabling community support and being free of cost. Fig. 4.2 provides a summary of the possible technology stack at the cloud and edge tiers.

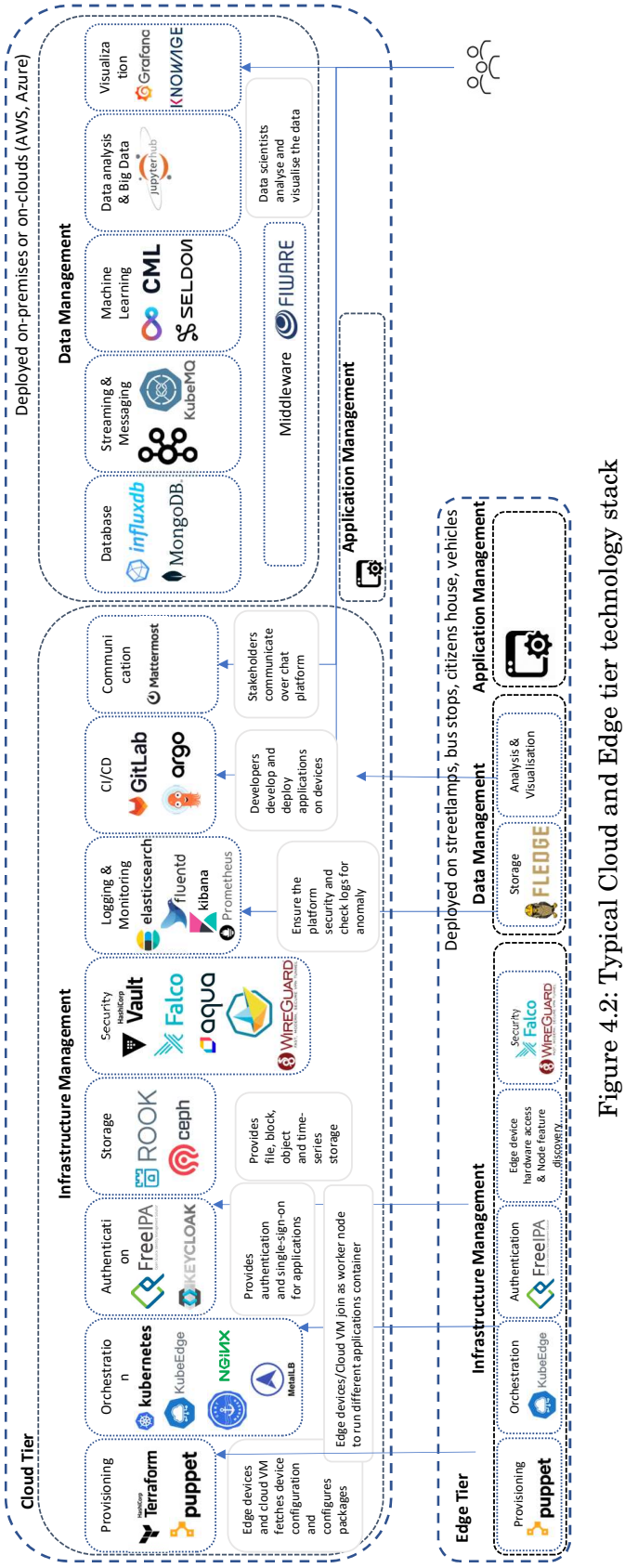

Figure 4.2: Typical Cloud and Edge tier technology stack

4.4.1 Infrastructure Management

Provisioning

Based on our requirements, such as infrastructure as code and the reduction of complexity and time in infrastructure deployment, the platform uses terraform [337], puppet forge modules [338], helm charts [339] and Kubernetes operators [340] to deploy the technology stack. Terraform [337] is an open-source infrastructure as a code software tool that enables the safe and predictable creation, change and improvement of the infrastructure. Puppet forge modules [338] simplify automation processes and define explicitly what packages to install and how they should be configured. Helm chart [339] defines, installs, and upgrades the Kubernetes application and is a highly customisable tool, and it abstracts the application deployment to a simple configuration file. Kubernetes Operators [340] include domain or application-specific knowledge to automate the entire life cycle of the software it manages. Kubernetes operators come in five different levels: Level I (basic install), which provides automated application provisioning and configuration management; Level II (seamless upgrades) patch and minor version upgrades supported; Level III (full lifecycle) ensures application and storage lifecycle tasks such as backup, failure recovery; Level IV (deep insights) provides metrics, alerts, log processing and workload analysis, and Level V (autopilot) that provides horizontal/vertical scaling, auto-config tuning, abnormal detection, scheduling tuning.

Based on the features provided, we believe that using Puppet Forge modules, Kubernetes Helm chart, and Kubernetes operators simplify the installation of the cloud-tier and edge-tier technology stack. All three are strongly community-supported and can be improved if any feature or support is required.

Authentication and Authorisation

Various open-source software provides authentication and authorisation. FreeIPA [341] is an integrated Identity and Authentication solution for Linux/UNIX network environments. It provides centralised authentication, authorisation, and account information by storing data about users, groups, hosts, and other objects necessary to manage the security aspects of a network of computers. The platform uses Keycloak [342], which provides user federation, strong authentication, user management, and fine-grained authorisation to add authentication to applications, secure services, and single sign-on.

CI/CD

Stakeholders would also develop and deploy urban applications. They would require a development environment with code storage and continuous integration and deployment capabilities. The platform uses Gitlab Community Edition (CE) [343] to provide DevOps functionality and store code and containers. GitLab CE [343] is an open-source end-to-end software development

platform with built-in version control, issue tracking, code review, CI/CD, and can be self-hosted on-premises. The platform uses GitLab Runner [344], which works with GitLab CI/CD to run jobs in a pipeline. The CI/CD pipeline can include code-linting tools that automate the checking of source code for programmatic and stylistic errors and static code analysis tools that perform automatic code review that systematically helps deliver clean code.

Furthermore, the platform uses ArgoCD [345] to enable automated deployment of applications to specified target environments and continuous declarative delivery of GitOps. ArgoCD also enables multi-tenancy and allows multiple application developer teams in the organisation to deploy applications. For example, suppose the IoT infrastructure platform contains ten edge devices, five deployed on the streetlamp and five in the citizen's house. In that case, it is possible to configure that one application developer/organisation only deploys the applications on the citizen's edge devices. On the contrary, other developers/organisations deploy on streetlamp edge devices.

Orchestration

The platform uses Kubernetes (K8s) [346], an open-source system to automate the deployment, scaling, and management of containerised applications and deploy applications to multiple edge devices. Kubernetes also allow multi-tenancy sharing clusters (a set of nodes that run containerised applications), saving costs and simplifying administration. Kubernetes is equipped with robust fault tolerance mechanisms to ensure application resilience in dynamic and distributed environments. Key features include automated detection and recovery from failures through container rescheduling and node replacement, allowing quick redistribution of workloads to healthy nodes. The platform supports multiple instances of applications, ensuring continuous service even if one instance encounters issues. Kubernetes also offers self-healing capabilities by monitoring container and node health, taking corrective actions in case of anomalies. In edge computing, Kubernetes extends its fault-tolerant design by deploying lightweight, containerized workloads on distributed devices. It adapts to resource constraints and intermittent network connectivity, facilitating dynamic scaling and incorporating features like local data caching. The platform's automated recovery mechanisms contribute to maintaining consistent performance and reliability in challenging edge scenarios.

The edge tier should be able to deploy applications and provide access to the edge hardware (endpoints, radios, cameras). Multiple edge orchestration engines deploy applications on edge devices, such as K3S [347] (Lightweight Kubernetes built for IoT & Edge computing), KubeEdge [348], OpenYurt [349], and SuperEdge [350]. K3S is a solution to manage local edge nodes with stable network connectivity. In contrast, KubeEdge provides resiliency and edge autonomy that ensures that edge nodes run autonomously even when the cloud-edge network is unstable. It also includes metadata synchronisation between cloud and edge and edge device management.

The platform uses NVIDIA device plugins for Kubernetes [351] to provide GPU access to AI/ML applications on the cloud and edge level. It also allows selecting specific nodes based on different features (RAM/CPU/radios/USB) using Kubernetes node feature discovery [351, 352]. NVIDIA GPU devices benefit from robust support within the Kubernetes NVIDIA GPU feature discovery framework. Nevertheless, there may arise situations where a newly launched device is not yet included in the supported roster. It is essential to note that both Kubernetes Node Feature and NVIDIA GPU Feature Discovery are community-backed open-source projects, ensuring that support for emerging hardware devices is actively available and continually expanded.

To allow for local resilient edge data storage and enable sending and receiving data from anywhere, the platform uses FLEdge/FogLAMP [353] software. FLEDGE [353] is open-source software that collects data from any/all sensors, aggregates, transforms and buffers, performs edge analysis and delivers data to multiple destinations.

Storage

The platform needs resilient data storage at the cloud tier. The platform uses rook [354] (open-source, cloud-native storage for Kubernetes) with CEPH [355] to have self-managing, self-scaling, self-healing distributed storage systems. It automates the tasks of a storage administrator: deployment, bootstrapping, configuration, provisioning, scaling, upgrading, migration, disaster recovery, monitoring, and resource management. Rook can provide production-ready management for file, block, and object storage.

Security

The platform security is paramount. The platform allows users to deploy application containers in the cloud and on edge devices. Container devices can be malicious. The platform uses the Falco project [356] (cloud-native runtime security) to detect threats at runtime by observing the behaviour of applications and containers. We use the Hashicorp Vault [357] to protect applications' and scripts' secrets (credentials/tokens) and for deploying the certificate authority (CA). The CA provides HTTPS certificates to applications deployed on the platform and can also be used to provide PKI certificates for encryption and mutual authentication. In addition, we use VPN (Wireguard [358]/OpenVPN [359]) to set up secure communication between edge devices and the cloud. To improve the security of Kubernetes, we use open-source Aquasec Starboard [360]. It provides automated vulnerability scanning for Kubernetes workloads, configuration audits for Kubernetes resources, infrastructures scanning and compliance checks, automated compliance report and penetration test results for a Kubernetes cluster. It uses the CIS Benchmarks and NSA, CISA Kubernetes Hardening guidelines to ensure security.

Logging and Monitoring

The platform uses the Elasticsearch [361], Fluentd [362], and Kibana [363] (EFK) to aggregate and analyse Kubernetes logs. The platform uses open-source Prometheus [364] as a monitoring and alerting toolkit.

Communication

The platform must use an open-source communication platform that can be used to message, audio/video calls and meetings. In our case, the platform used Mattermost [365]. It is an open-source, self-hostable online chat service with file sharing, search, and integrations.

4.4.2 Data Management

Database

Currently, the platform stores time series data using InfluxDB2 (an open-source time series data platform). However, most of the database can be installed on the platform based on the requirements. Databases such as PostgreSQL, MariaDB, MongoDB Redis, and MySQL can be installed using Kubernetes operators [366].

Streaming and Messaging

To provide streaming and messaging, the platform deployed Apache Kafka using the Strimzi Kubernetes operator [367]. Other streaming and messaging software, such as Apache Flink, KubeMQ, and RabbitMQ, can be deployed using Kubernetes operators [368].

Machine Learning

Currently, we have not implemented machine learning frameworks on the platform. However, open-source software such as Continuous Machine Learning (CML), CI/CD for Machine Learning Projects, is compatible with Gitlab Runners (hosted on the platform). Also, the open-source SeldonCore [369] can be deployed on the platform to deploy machine learning models on Kubernetes at a massive scale.

Data analysis and Big Data

For data analysis by different stakeholders (data scientists), the platform provides JupyterHub [370] with GPU access for AI/ML applications. JupyterHub brings the power of notebooks to groups of users. It provides users access to computational environments and resources without burdening them with installation and maintenance tasks.

Visualisation

For providing data visualisation, the platform uses Grafana CE [371]. Grafana allows users to query, visualise, alert, understand metrics, and create beautiful dashboards. Another open-source solution, Knowage [372], combines traditional data and big data sources into valuable and meaningful information. It also provides different features such as data exploration, data preparation, self-service data, ad-hoc reporting, mashup, data/text mining embedding, advanced data visualisation on big data/cloud data sources and augmented analytics.

Middleware

The platform currently does not deploy any middleware; data from edge devices are directly stored in the Apache Kafka topics or the InfluxDB database. However, middleware such as Fiware [373] can be deployed using Helm charts [374].

4.4.3 Application Management

The application management module can deploy any custom application according to the project requirements. In our case, we have deployed an application to run an MQTT server and a Python script that receives the messages from the endpoints and sends them to the InfluxDB server on the Cloud tier. We also created a Python script and deployed it on the edge device in a container to simulate the application message flows defined in the next chapter (chapter 5) to validate the resilience of the network on the edge device.

4.5 Sharing the Smart City Framework from Single to Multiple Owners

Once the infrastructure has been defined and implemented, it is first owned by a single owner/ organisation where either a single member or different team members maintain the infrastructure. However, the infrastructure (cloud, edge resources) is not shared between multiple projects or organisations. Before sharing the infrastructure, it is crucial that the infrastructure is working reliably and secure.

It is also important to highlight that establishing shared infrastructure for collaborative research presents significant challenges, particularly in identifying willing research partners. The core idea involves using software that enables multiple users to operate on the same hardware and software infrastructure while ensuring data separation. However, the practical implementation of this concept is intricate. The complexities of facilitating shared infrastructure hinder widespread adoption. To advance in this area, funding bodies support in sharing the infrastructure between research partners is crucial. If these bodies endorse and fund shared infrastructure initiatives, it could accelerate collaborative research. Kubernetes, a tool facilitating infrastructure sharing,

emerges as a potential solution. Yet, its adoption introduces challenges, such as ensuring security, optimal performance, and managing quality of service. Striking a balance between collaborative resource utilization and robust security measures is essential for the successful integration of shared infrastructure in academic research.

This section briefly describes shared infrastructure, different owners in shared infrastructure, and sharing infrastructure with multiple stakeholders/projects/organisations.

4.5.1 Definition and an Example

Shared infrastructure can be defined as a publicly deployed IoT three-tier infrastructure consisting of multiple edge and endpoint devices deployed in either citizen's houses or urban spaces that share the resources at the edge, endpoint, and cloud tier. Additionally, it allows citizens and organisations to add their own devices (BYOD) (cloud/edge/endpoint) to suit their individual needs. There can be two BYOD scenarios. First, BYOD endpoints where students/citizens/businesses want to measure dampness/noise/air pollution in the house/street using their own sensors (such as SCK) and require computation, cloud storage and visualisation services. Second, is the BYOD edge, where city councils/transport authorities/others want to install an edge device with CCTV at bus stops/streetlamps to understand pedestrianisation (footfall, bicycles, cars) and want to deploy applications on established infrastructure. For example, community/authorities deploying applications to resolve a community problem (bike theft/neighbourhood watch/biodiversity) may involve adding additional infrastructure at the edge (thermal cameras, specific sensors (ozone, NO2, radiation)). It can also include collaboration with advertising companies to share infrastructure (BT InlinkUK, JCDecaux). Shared infrastructure can have a single owner or multiple owners.

To provide an example of shared infrastructure, the city council often has an IT team and provides computing hardware (such as access to Openstack and LoadBalancer with few public IP addresses) to the different community projects. Community projects often have an IT team to create virtual machines (VM) and deploy multiple services on the provided computing hardware. The community team collaborates with the university research team to deploy edge and endpoint hardware devices in citizens' homes. In addition, it can also involve the deployment of additional edge hardware and applications in the installed infrastructure. City council involvement, infrastructure sharing, and collaboration with multiple organisations (community and university research teams) reduce infrastructure deployment costs. In the case of a single owner, the above roles would still be valid. Instead of entities as an organisation or vendors, an entity would be individual or team members in the same organisation.

There could be scenarios where the edge hardware is shared or improved. For example, addition of USRP B210 software defined radio that can prototype a GSM base station and act as a eNodeB. However, edge device hardware (RPi/JN) performance constraints can be a limiting factor.

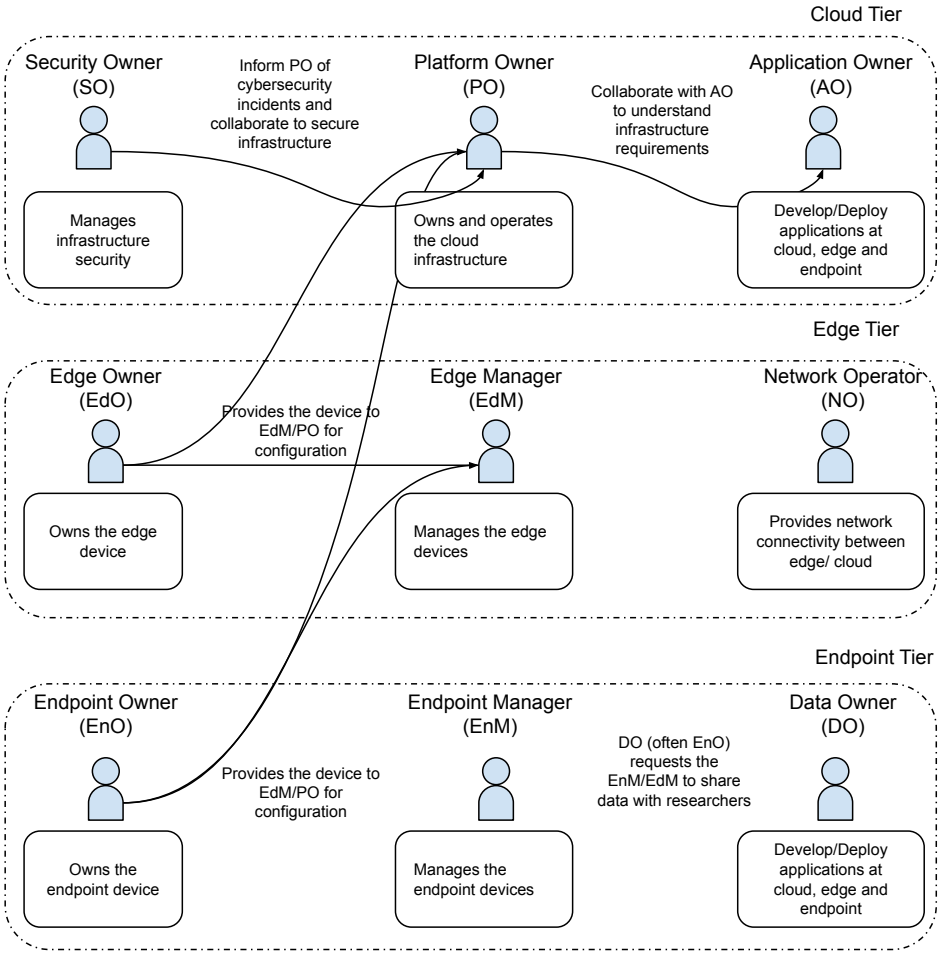

Figure 4.3: Relationship between different owners at different tiers

4.5.2 Different Owners in Shared Infrastructure

In research projects, if there is only one owner, all infrastructure would be owned and managed by them. However, in the case of shareable infrastructure, multiple entities (owners, vendors, or organisations) exists at each tier ensure end-to-end data collection, application management, and delivery. It is an extensive ecosystem containing numerous entities that work together to collect data from endpoints and store, analyse, and visualise them in the cloud. In the following, the author provides a rough set of entities that can operate an end-to-end platform at different tiers (cloud, edge and endpoint).

Fig. 4.3 summarise the owners involved at each tier and their roles. The curved-square box provides information about the responsibility of the particular entity and the directional arrow represents the interactions between different owners.

Cloud Tier

There can be multiple entities in the cloud tier.

- First, an entity can be a platform operator (PO) that owns the cloud (hardware/software). The PO owns the deployed hardware (servers, network equipment), operates the platform (configures the servers, network equipment), and ensures that the core services (credential management, data storage) are provided to the edge and endpoint devices.
- Second, an entity platform security owner (SO) manages the platform's security. Responsibilities include informing the PO of any cybersecurity incidents. The incidents can be unauthorised user logins, malicious activity on the devices deployed on the endpoint, edge, and cloud tier, unauthorised endpoint attempts to join an edge network, and ensuring the whole platform's security. It may also involve disabling an endpoint/edge for cybersecurity or operational reasons.
- Thirdly, an entity named application owner (AO) develops and deploys the applications on the endpoint, edge and cloud. The applications may require access to the edge hardware (sensors, radios, cameras) and process the endpoint's data or deploy TinyML models on the endpoints.

Edge Tier

There can be multiple entities at the edge tier.

- First, an entity edge owner (EdO) owns the edge devices. The devices can be an RPi, JN, or Intel NUC that provides the physical edge infrastructure, including a core set of generic capabilities (computing power, secure storage, GPUs). Furthermore, the PO can offer the ability to citizens/users to BYOD and attach it to the public infrastructure such that the user owns the device; however, the device is managed by a different team (PO).
- Second, an entity Endpoint Manager (EdM) manages the data storage and flow at the edge (data origin and destination) and how it is stored and transmitted securely and configures the edge device.
- Third, an entity network operator (NO) provides network connectivity between the edge and cloud. For example, BT or Vodafone provides fibre or 5G connectivity.

Endpoint Tier

The endpoint tier can also consist of different entities.

- First, Endpoint Owners (EnO) own the endpoints and purchase them from an OEM (which provides physical endpoint devices, such as the SCK team). The EnO can send the device to

the PO/EdM or configure it. They can request the EdM to set up a data flow at the edge to decrypt and process data from an Endpoint and make it available to cloud services.

- Second, the data owner (DO), often the EnO, will be the primary owner of the data generated by the endpoint and defines the other roles of who would have access to the data and in what form. For example, citizens with SCK kits are EnO and can provide data to a data scientist for analysis.

- Third, an entity named endpoint manager (EnM) manages those endpoints and ensures that endpoint devices are ready to be added to an edge network, including endpoint configuration and data encryption/decryption details. The EnO can also request the status of its endpoints and select a set of endpoint data for data scientists to process.

Fig. 4.4 provides a shareable scenario with different owners PO, SO, and AO, where the cloud tier and the devices at the edge tier are owned by different owners and are shared between different stakeholders. PO often manages the cloud tier, configuring the different components that SO and AO use. The edge tier and devices are configured by PO and EdO and used by different stakeholders. For example, in the first scenario, students, citizens, and businesses who want to measure the dampness, noise, and air pollution in the house/street can provide their own sensors (acting as EnO). They (EnO) can request PO/EdM/EnM to configure the endpoint device and collaborate with the local community (such as KWMC)(acting as AO) to deploy an application on the edge device 1-2 or cloud. They would only require data storage, analysis and visualization services from the cloud tier. In contrast, in the second scenario, a research project can either own the edge device (EdO) or collaborate with PO to add hardware (thermal camera, extra sensors) at edge devices 3 (owned by the PO) and 4 (owned by the research project - acting as EdO) and deploys urban applications on edge devices 2, 3 and 4.

4.5.3 Exploring Open-source Components Capabilities for Sharing of Infrastructure

In the smart city framework, the infrastructure is managed mainly by the platform operator, who configures the sub-modules for the security owner or application owner to use. Suppose the open-source component enabling a particular functionality provides the multi-tenancy and sharing of the open-source component. In that case, the smart city infrastructure can be shared between stakeholders and owners. Different research projects can have a set of team members that manage that research project and can include different stakeholders such as end-users, data scientists, application developers (acting as AO) and others (§ 4.2.1).

In our smart city framework, we explore the possibility of multi-tenancy in the open-source components, which can help share the infrastructure between stakeholders and owners. Multi-tenancy can be defined as multiple parties using the same resource without learning information about each other. If the application is multi-tenant, then each user/group/organisation can see

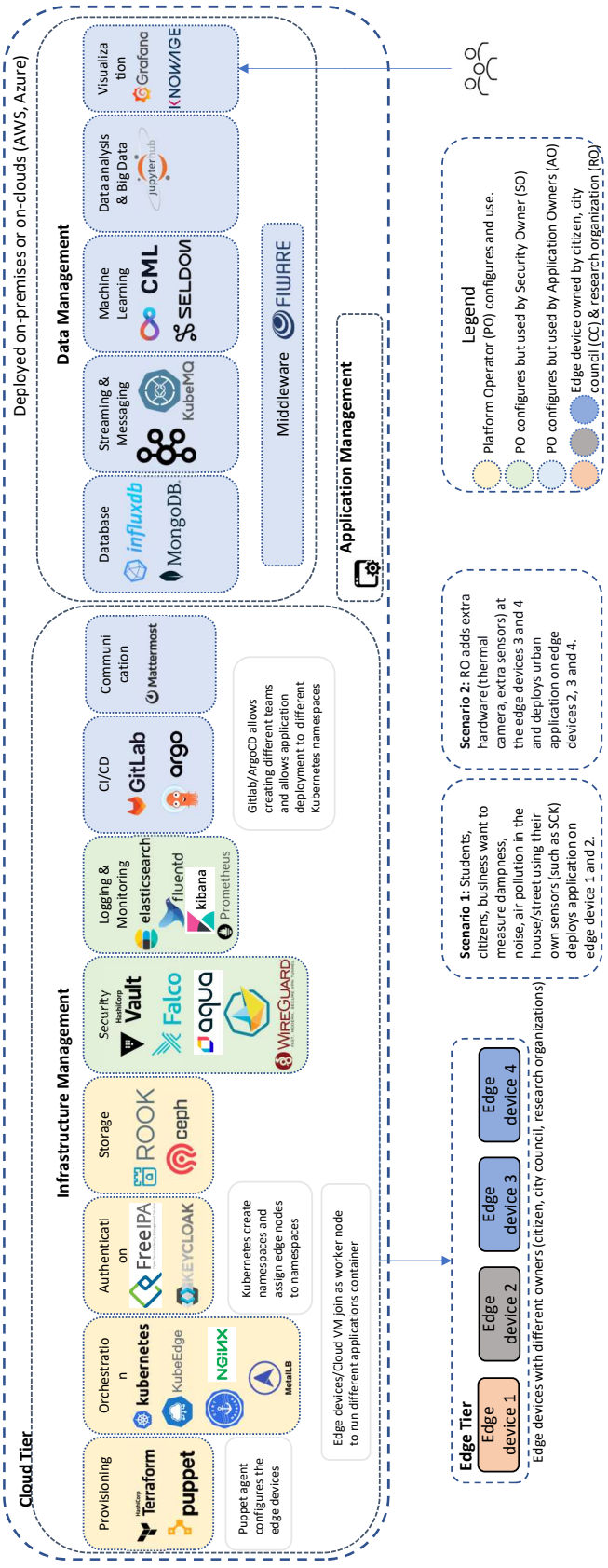

Figure 4.4: Typical Cloud and Edge tier technology stack owners and different scenarios

only their data. However, there is a single application running to host those users. Two completely different users can go to the same application/URL and log into the same system. However, the data is segmented based on their roles, groups, permissions, and access to only their data. Multi-tenancy saves costs and simplifies administration. Next, we consider the different open-source components used in the implementation of the smart city framework and how they can help in multi-tenancy.

Infrastructure Management

In the smart city framework, we use freeIPA and Keycloack to provide authentication and authorisation, Gitlab, ArgoCD and Kubernetes for enabling the development and deployment of applications on different devices.

Although FreeIPA does not provide full multi-tenancy, it does provide the capability to create the user and host groups and provide HBAC (Host-based access control)-Rules, SUDO-Rules and Role Based Access Control [375]. PO (system administrator) can apply HBAC-Rules, privileges, roles, and sudo-rules to the objects above (users and hosts). PO can use HBAC rules to limit access to a specified system (cloud/edge device) to members of a specific user group or allow only a specific service to be used to access systems. Sudo rules are similar to access control rules: they define users who are granted access, the commands within the rule's scope, and the target hosts to which the rule applies. Essential elements define who, what (services), and where (hosts). RBAC organises access to the data managed. Users who perform the same tasks within an organisation can be grouped and assigned a particular role. The role can provide members groups and users with the necessary permissions to perform their assigned tasks.

Keycloak does provide multi-tenancy by providing realms [376]. A realm manages users, credentials, roles, and groups. Realms are isolated from each other and can only manage and authenticate the users they control. It also has the concept of organisations that are "tenants" or "customers". A realm can have multiple organisations. Memberships are the relationship of users to organisations. It also supports roles that are mechanisms of role-based security specific to an organisation.

Gitlab provides multi-tenancy by providing organisations and namespaces (personal and group/subgroup) [377]. A namespace is used to provide a separate naming space. Object names within a namespace can be the same as names in other namespaces. A namespace provides a place to organise related projects for users and groups. ArgoCD provides projects with a logical grouping of applications useful for multiple teams. ArgoCD projects [378] enable restriction of what may be deployed (trusted Git source repositories), where applications may be deployed to (destination clusters and namespaces), what kinds of objects may be deployed (e.g. RBAC, CRDs, DaemonSets, NetworkPolicy and others), and defining project roles to provide application RBAC.

Kubernetes provides multi-tenancy and allows sharing clusters to meet the demands of multiple teams and customers [379]. It provides several features, such as control plane isolation

(namespace, RBAC and quota) and data place isolation (container runtime, storage and network) to help manage different tenancy requirements. RBAC and quota are also scoped to namespaces. ResourceQuota can impose restrictions on the resource usage of namespaces.

Multi-tenancy support in all the open-source components enables the sharing of infrastructure owned by different owners (PO, SO, AO, EdO, EnO) between different research organisations and stakeholders.

Data Management

The platform currently uses InfluxDB for database management, Apache Kafka for streaming, CML for machine learning, Grafana for visualisation and JupyterHub for analysis.

InfluxDB provides the ability to create organisations and buckets to store data from different sources [380]. It also provides API keys to provide access (read/write) to the buckets. Apache Kafka also provides authentication and authorisation on the Kafka topics based on an access control list, Keycloak Authorisation services and Open Policy Agent [381]. CML uses Gitlab CI/CD to perform operations that enable running the machine learning code based on each user profile [382]. Grafana also enables multi-tenancy by providing Organisations and RBAC roles. Grafana organisations [383] help isolate users and resources such as dashboards, annotations, and data sources from each other. For data analysis, JupyterHub [384] enables providing Jupyter notebook for multiple users. It manages a separate Jupyter environment for each user and can be used in a class of students, a corporate data science group, or a scientific research group.

Application Management

The applications module allows users to deploy applications on different devices. Kubernetes namespaces enable the deployment of applications on different devices based on the Nodeselector and resource limits.

4.6 Implementation: Steps involved

We implemented the above architecture and technology stack on 2 Intel NUC and one Dell Machine Tower (3 TB SSD, 192 GB RAM) serving as the cloud tier. SBC (four Raspberry Pi (64GB SD Card, 8GB RAM), 2 Jetson Nano (64GB SD Card, 4GB RAM) and 1 Jetson Xavier) serve as edge level. Installing and maintaining the components required to support SCRP is challenging. Maintaining application infrastructure requires many repetitive steps, is complex and challenging, consumes much time (if manually installed and configured), and requires skilled system architects. In order to resolve the challenges faced in maintaining the technology stack, the platform requires a declarative infrastructure, such as a code approach, software automation tools, quickly deployable and configurable software components, and a DevOps approach. Fig. 4.5 provides information on how we configure the technology stack required for smart-cities research

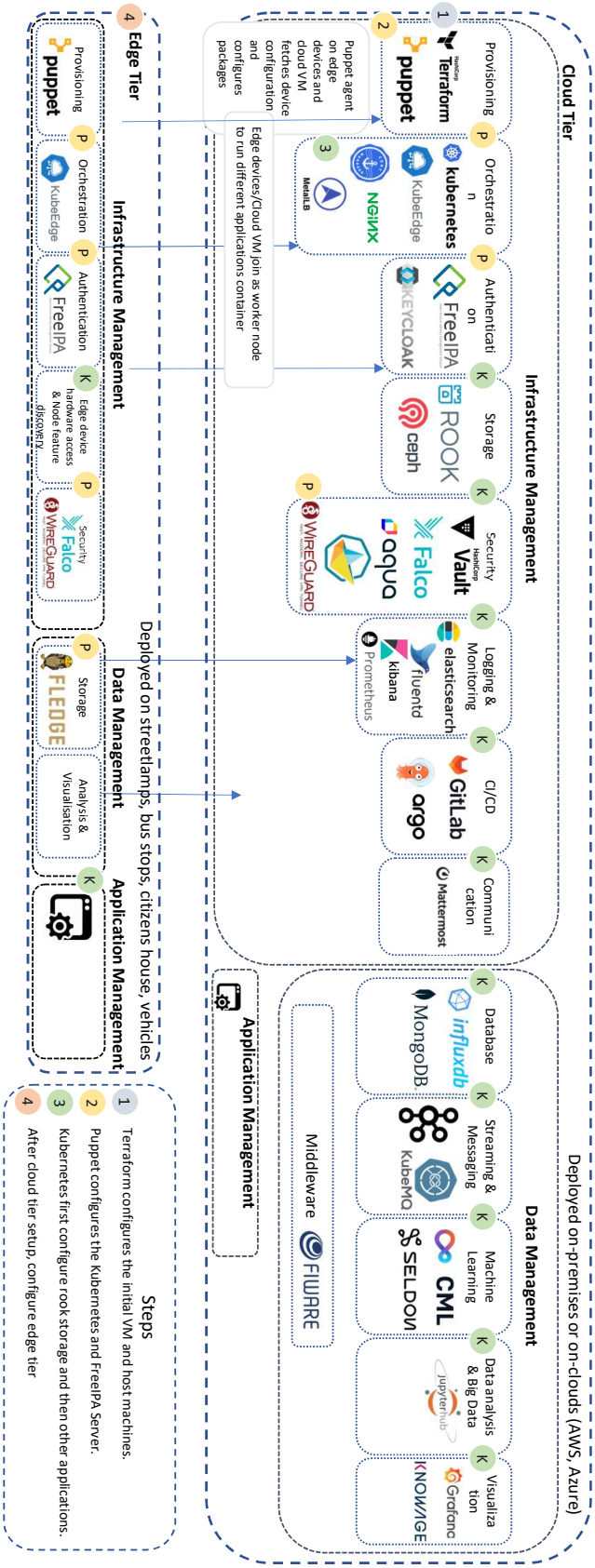

Figure 4.5: Implementation steps

projects. The implementation also resulted in X GitHub Issues, Y Features, requests, Z GitHub discussions, E Modules creations, and updates.

Initially, we start with a server with a newly installed OS (which could be Debian, Ubuntu, or the system architect is comfortable with) and install the basic packages required for virtualisation and Terraform. The platform uses Terraform to create VMs that run puppet, freeIPA, and Kubernetes servers. We first set up a Puppet server to run the rest of our infrastructure. Authentication (FreeIPA), orchestration (Kubernetes), and security (wireguard) modules are configured using Puppet. Once the Kubernetes primary node is initialised, the other virtual machines or host machines are joined as a worker node. In our setup, we use one server to host the VM and the other two machines running as worker nodes. Once the Kubernetes worker nodes are configured, storage is configured using rook, and other components are configured using helm charts and Kubernetes operators.

Once the initial infrastructure owned by a single owner is configured and ready to use, the infrastructure can be shared after understanding the requirements of the participating research organisations. The requirements gathering can involve understanding the project, the objective, the data collected, processing resources required, the number of users, user and group roles, and access required. In our case, we implemented the sharing of infrastructure between two projects in which the author participated as a research member. Project A had five different work packages implemented by different collaborators (companies). Each WP has implemented a container to be deployed on the cloud and edge devices collecting air quality data from the endpoints deployed. Project B used InfluxDB and Grafana to collect air quality data from the endpoints. The organisation, users and groups are currently configured manually based on the requirements.

4.7 Evaluation

We evaluate our work by mapping the framework modules to a smart city's challenges and software requirements. We also provide a few case studies in which we use framework modules and perform qualitative analysis to determine the saved time. Ideally, the accurate evaluation of the smart city framework would be to use the framework and architecture in at least two research projects that share the infrastructure to deploy the different blocks of the platform. However, that requires two funded projects with defined requirements, which was not feasible. Instead,

Table 4.1
Mapping the smart city framework to the challenges

Challenge	SCF Blocks	How	What
System Design			
End-to-end security	IM: Security	Continuous security tools, IaaC, DevOps	Starboard, Kubescape, Terraform, Puppet, Falco
Resilience (thermal, testbed)	IM: Provisioning/Orchestration	resource limits, declarative IaaC, Technical Wiki	Kubernetes, Terraform, Puppet
Authentication, authorisation, CA,	IM:Authentication	Credential management and central authentication	FreeIPA, Keycloak, HashiCorp Vault
Exposed services and security updates	IM: Security	Seamless upgrades of applications with rollback features	Kubernetes Operators and Helm Charts
Data storage, access and integration	IM: Storage/DM: Database, Streaming/Messaging	Self-managing data storage; different storage types, APIs	Rook, Database APIs
Implementation			
Provisioning the cloud, edge devices	IM: Provisioning	IaaC, Declarative approach	Puppet, Terraform
Application deployment and compatibility	IM: CI/CD; Applications	Development with pipelines for automatic build	GitLab, GitLab Runner, ArgoCD
Accounting and monitoring	IM: Logging and Monitoring	Log aggregation and analysis	Elasticsearch, Fluentd, Kibana, Prometheus

4.7.1 Mapping Smart City Framework to Smart City Requirements and Challenges

Challenges

Table 4.1 summarises how the smart city framework modules solve a few challenges faced in implementing smart city research projects. The declarative approach and automation tools (Puppet, Helm chart, and Kubernetes operators) resolve the challenges around “system design - exposed services and security updates” by enabling the seamless upgrade of the applications (such as rook, keycloak and other applications deployed using Kubernetes). The seamless upgrade ensures that security updates are applied appropriately with rollback features and applications remain without known vulnerabilities. Further, using open-source security tools such as Starboard, Kubescape, and others helps the system architect identify and resolve security issues to ensure the security of the Kubernetes infrastructure and the applications running. Using Kubernetes and automation tools also helps ensure the testbed is scalable, modular, and extensible.

The data storage (rook, InfluxDB), visualisation (Grafana), and analysis (JupyterHub) blocks address the “system design - data storage, reduction, access, and integration“ challenges. The data storage (rook) uses CEPH in the background and provides capabilities to encrypt the data at rest. All databases provide the ability to transmit data using secure connections. The platform also provides capabilities to collect and store data dynamically using streaming and messaging applications. The streaming block can receive the data and be delivered to multiple receivers.

The authentication and authorisation tools (FreeIPA and keycloak) solve the challenges around “system design - authentication and implementation - credential management” by providing centralised authentication mechanisms for users and devices. The CI/CD block and the open source software (GitLab, ArgoCD) provide the capability for different stakeholders to develop applications and deploy them on various devices collecting urban data.

The provisioning block (infrastructure as code automation tools) enables the provision of devices at the cloud, edge tier in a declarative way (Implementation phase). The declarative approach helps build the testbed resilience (system design phase) such that the testbed can automatically correct itself based on the required changes.

The logging block and open source tools (elastic search, fluentd, kibana) help the system architect monitor the testbed by using various device logs, performing the required accounting,

Table 4.2
Mapping the smart city framework to the software requirements for smart cities platform

	Infrastructure Management	Database Management	Application Management
Functional Requirements			
Data Management (collection, storage, analysis, visualisation)		✓	
Data processing (analyse, verify, aggregate, filter datasets)		✓	
Data ingestion (importing real-time/batch data)		✓	
Metadata management (managing information about data)		✓	
External data access (API to access data)		✓	
Definition of city models (create models to understand city data)		✓	✓
Application Runtime (deploy applications)	✓		✓
Service management (create new services/applications)	✓		✓
Software Engineering tools (develop and maintain applications)	✓		
Machine learning capabilities		✓	
WSN Management (adding, removing, monitoring sensors)			✓
Non-functional requirements			
Interoperability (different devices, systems, applications, platforms)	✓	✓	✓
Scalability (increase of users, data, service over time)	✓	✓	✓
Security, Privacy	✓	✓	✓
Context awareness		✓	
Extensibility, Configurablity	✓	✓	✓
Availability	✓	✓	✓

and resolving integration and operational testing challenges.

The communication block ensures the interaction between the users and between applications and the users. Users can configure the application to provide status updates on tasks such as start/stop/error, etc. It enables users to check their status and receive updates on the communication platform rather than log in to the application.

Smart City Software Platform Requirements

Table 4.2 summarises the functional and non-functional requirements fulfilled by the smart city framework modules. The infrastructure management module (§ 4.3.1) consists of provision, orchestration, authentication, storage, security, logging & monitoring CI/CD, and communication. It satisfies functional requirements such as application runtime, service management, software engineering tools, and the definition of a city model. The data management module (§ 4.3.2) consists of sub-modules: Database, streaming & messaging, machine learning, data analysis and big data, visualisation, and middleware. It satisfies functional requirements such as data management, data ingestion, metadata management, data processing, data analysis, machine learning capabilities, external data access, visualisation, and the definition of a city model. The applications module (§ 4.3.2) and the provisioning submodule allow the installation of a custom system and user applications, such as WSN management requirements.

Infrastructure management, data management, and applications hosted on open-source, community-based Kubernetes infrastructure provide IaaS, PaaS, and SaaS. They satisfy a considerable extent of interoperability, scalability, security, privacy, context awareness, extensibility, reconfigurability, and availability.

Table 4.3
Qualitative Analysis: Time required (hours)

Setting 20 open source components	Min	Avg	Max
Individually	40	80-160	160-320
Using SCF (Kubernetes Operators/Helm/IaaC)	20	20-40	80-160

4.7.2 Case-Studies: Validating the Framework Modules

Many parts of the smart city framework were used on small projects for the single-owner case. Based on the requirements of these research projects, different blocks could use different tools. The different blocks and tools validate the smart city framework categorisation. We deployed the InfluxDB, Grafana and JupyterHub servers using Kubernetes operators and Helm Charts to support the Cotham Hill Pedestrianzation research project that collected data from eight SCK devices deployed in the citizen's houses, analysed in Jupyter and visualised it on Grafana.

The author was also involved in the Synergia project focusing on a secure-by-design end-to-end platform for large-scale resource-constrained IoT applications. The author was responsible for setting up the infrastructure and integrating different work package outputs in the final deliverable. The author used Puppet Bolt (Provisioning Block) to configure the devices (edge and cloud tier); Kubernetes (K3S) with Ingress Controller and Cert Manager (Orchestration Block) to be able to deploy containers to the devices and provide TLS certificates. To provide secure communication between Kubernetes master and worker nodes, the author used Wireguard with K3S (Security block). The project also provided application development/CI/CD using Azure DevOps and Azure Container Registry (CI/CD block). The project used InfluxDB and Grafana (Database and Data Analysis block) to provide data storage and visualisation. The Synergia project also deployed a custom application (application management block) to manage endpoint permission to join the 802.15.4 edge network.

4.7.3 Qualitative analysis: Efficiency and Time

The author also evaluates the smart city framework from the time required to set up the infrastructure. Research projects may require in-house development and manufacturing of the technology (hardware and integration of different hardware to create a working product) that takes considerable time from the initial concept of the idea and procurement to the final working product. Suppose the research project has technology that is ready to be deployed. In that case, deployment time depends on different permissions required (such as opening a port on the firewall/deploying the hardware on a structural bridge, agreements with the owner where the infrastructure is deployed, deployment size, and required security for the infrastructure). Considering the above challenges, estimating the time required to set up the infrastructure is complex.

In contrast, if we consider a situation where the deployment team has the necessary permissions, hardware, and technology ready to deploy, we want to estimate the time required to set up software infrastructure (such as cloud tier (servers), edge tier (gateways)) that provides the necessary services to collect, process, and visualise the data in a meaningful way. Provisioning the cloud tier and edge tier depends on the different services required on edge tiers such as authentication, security, application development and deployment, visualisation and others and setting up the open-source components such as FreeIPA, Kubernetes, GitLab, ArgoCD and others. Each software component must be adequately configured for each other to allow the whole system to operate flawlessly.

The standard method of deploying a software service involves installing and configuring software components one by one, often requiring administrators to switch back and forth between components and configuring and remembering different configuration parameter values.

The author considers a hypothetical scenario that the research project has a system administrator/architect with average Linux and DevOps experience. We further assume that the operating systems to host the open-source components are readily available. Let us assume that the time required to configure a typical open-source component using the traditional method without automation takes a minimum of 2 hours to an average of 4-8 hours and a maximum of 8-16 hours. For example, if a research project requiring simple storage and visualisation requires configuring only two components, Grafana and InfluxDB, it would take a minimum of 4, an average of 8-16 and a max of 16-32 hours. On the other hand, a research project requiring an end-to-end secure system to collect, process, and visualise data with secure central authentication, logging and application development capabilities might have 20 or more open-source components. Configuring approximately 20 components might take a minimum of 40 hours, an average of 80-160 hours and a max of 160-320 hours. In terms of days, configuring two components take a minimum of 0.5 days to an average of 1-2 days and a maximum of 2-4 days, whereas configuring 20 components can take a minimum of 5 days, an average of 10-20 days and a max of 20-40 days. Suppose we add the delays that are out of control (such as hardware procurement, permissions required, administrative tasks, infrastructure deployment and others). In that case, the total time required to set up infrastructure for a research project can range from 1.5 to 4 months or more.

On the other hand, most smart city research projects have similar requirements in application development and deployment, data storage, processing and visualisation. They might use similar open-source components to fulfil the requirements. Much of the configuration information related to the interdependency between software components are often similar across the research projects, such as secure communication between different components and users.

Our work provides a smart city framework and open-source components that implement different functionalities to fulfil the requirements of the smart city framework and the challenges faced in the implementation using DevOps and infrastructure automation. The author uses

Puppet, Helm Charts and Kubernetes Operators to stand up the infrastructure. Our work enables the administrator to describe the deployment declaratively rather than manually install the components individually. Once the components configuration has been defined declaratively, IaaC tools (puppet, helm, operators) deploy the components accordingly.

Regarding time, there is an initial setup time for configuring the puppet and Kubernetes server, which can range from 8-16 hours. The components are defined declaratively, and a rough estimate for the configuration of components can take a minimum of 1 hour to an average of 1-2 hours and a max of 4-8 hours. Configuring 20 open-source components can take up to a minimum of 20 hours, avg 20-40 hours, on a max of 80-160 hours (min 2.5 days to max 10 - 20 working days with an average of 2.5 days to 5 days). Table 4.3 compares the time to set up 20 open source components individually and to use the smart city framework and automation tools. Our work provides technical documentation providing details on the configuration of the infrastructure with a declarative configuration. It can help future research projects to reduce the implementation time and ensure an end-to-end infrastructure to collect, process and visualise data with no or minor configuration changes.

The declarative infrastructure also helps in scaling up the infrastructure. For example, if there are only ten edge devices with the same configuration and one owner, the configuration is comparatively more straightforward. However, if hundreds of edge devices have different configurations and owners, managing them without IaaC becomes challenging. Logging, monitoring, security, provisioning and other components in the smart city framework help manage and configure the devices at the cloud and edge tier.

4.8 Advances in state of the art

The author reviewed the smart cities framework, the IoT reference architecture, and the data gathering framework provided in the research literature and checked if they can be used to solve the challenges faced or implement a smart cities project. We assume that a platform implements a requirement or resolves a particular challenge if the literature explicitly states so, if the platform has a component, module, or technology that fulfils that requirement, or if a Wiki with instructions is provided to implement a smart city platform.

The author looked at different research papers that mentioned the “smart city framework”, “data gathering framework” and “IoT reference architecture” using Google scholar. Multiple research papers [8, 302–317] have proposed a smart city framework; however, they are high-level, generalized, and do not provide reproducible implementation reference architecture, or the technology mentioned does not solve all the challenges. Moreover, to our knowledge, there is little evidence that these architectures have ever been used for an SCRP. Also, most SCRPs focus on a specific domain, target a specific problem or are developed from scratch with little software reuse.

Santana et al. [302] analyzed multiple platforms, two architectures (CiDAP [258] and Ope-

nIoT [246]) and provided a novel reference architecture for the software platform for smart cities. It contained the 'cloud and networking module' to identify all devices connected to the platform, 'IoT and Service Middleware' to manage the city IoT network, communication with users and IoT devices, and management of services provided to applications. The 'user management component ' provides user authentication services, whereas the 'Social network gateway' retrieves data from social networks (Twitter, Facebook). The module 'big data management' manages all data on the platform and has three repositories, app, model and data repository, to store applications, city models and city data. Stream processing, machine learning and data cleaning, application development modules, and smart city simulators are also available. They presented technology options to implement the reference architecture using tools used by the platforms described in the survey. For example, using the Security Assertion Markup Language (SAML) protocol to provide security and OpenNebula [385] and Microsoft Azure [386] for the cloud environment. However, such suggestions are relatively high-level and make it difficult to reproduce the desired infrastructure. The article did not provide information on how such tools can be implemented and integrated to form a smart city platform. Our framework (§ 4.3) provides different submodules and different open-source technology tools/software (§ 4.4) to implement an SCRP platform. It complements the unified reference architecture by extending it to edge and endpoint devices. Our work also considers the challenges faced during SCRP deployment (§ 4.2.2) and is customized to the Cloud-Edge-Endpoint infrastructure, takes a tiered approach, and is suitable for a three-tier IoT infrastructure, easy to replicate, supports, and promotes software reuse.

Similarly, Clout [255] and its successor BigClout [387] introduced City Infrastructure as a Service (CIaaS), City Platform as a Service (CPaaS) and City Software as a Service (CSaaS). CIaaS provides infrastructure management, computing, and storage services and introduces virtualization for city resources (IoT devices, legacy devices, and web applications). CPaaS layer enables city services to be created by exposing APIs to access city infrastructure and resources. CSaaS layer enables users to consume services built using CPaaS APIs. Clout was extended to BigClout to support big data analysis, self-awareness, and real-time intelligence and to expand the platform towards edge computing. Clout and BigClout provide a service approach (IaaS, PaaS, SaaS) and different services that can be subscribed to on an already-built infrastructure. In contrast, our work takes a grassroots approach, providing modules and sub-modules to create a platform infrastructure per the smart-city platform requirements. Our work does not assume that a third party provides a platform, infrastructure, or service. Our work takes an agile, low-cost, off-the-shelf, non-proprietary component approach. Our work is influenced by the challenges faced in deploying SCRP and UBIC work. We were setting up the infrastructure for a different project that required much manual effort. It supports the deployment of infrastructure connecting citizens. Clout and BigClout use custom tools created during the project to provide the functionalities; however, it does not provide any information on how the infrastructure was created or provide any Wiki or scripts to replicate the infrastructure.

The author also looked at data gathering frameworks [318–324]; however, most of the papers refer to data gathering using WSN infrastructure. Further, the author referred to different organizations' IoT reference implementation architecture (such as Toshiba IoT Reference Architecture [326], WSO2 Reference Architecture [388], IBM IoT architecture [325]); however, such reference architectures either use or promote their native products. The author further looked at the different smart cities research projects (chapter 3) involving real-world deployment implemented in different cities; however, the architecture in such projects is mostly custom and based on project requirements.

4.9 Conclusion

Multiple smart city research projects collect urban data to improve citizen's life and solve problems related to climate change, carbon neutrality, and net zero. However, a smart city platform has many challenges and different software requirements. Our work provides a smart city framework with three modules: Infrastructure management, Database management, and Application management. The infrastructure management module manages and operates the smart city research platform. The data management module ensures that data is ingested from all sorts of devices with any data rate, stored on the databases, analysed, and visualised to gather insights. The application management module can host the custom system or user applications deployed on the cloud, edge, or endpoint tier. We also provided details on implementing these components using open-source components to help future research projects that collect, process, and visualise data reduce implementation time. Furthermore, the author explored the sharing of infrastructure between different research projects and organisations to reduce costs and use resources efficiently. The author evaluate the smart city framework using three methods: mapping the framework to the challenges and software requirements, providing case studies of different projects that validate the framework's components, and qualitative analysis of how the framework reduces implementation time.

In chapter 3, we provided the challenges a smart city faces in the design, implementation, and deployment of infrastructure for collection of data. In chapter 4, the author provided a smart city framework that aims to solve the challenges solvable by technology using three components: Infrastructure management, Database management and application management. The future researchers and smart city research projects can use the challenges (chapter 3) and the smart city framework (chapter 4) to anticipate the challenges and plan their projects efficiently, reliably and securely. A smart city research project can have different applications running on the cloud and edge tier, such as air pollution monitoring and streetlight monitoring. It is important that the applications must always have network connectivity and resilience to process the data and create effective decisions. In chapter 5, we explore applications with specific QoS requirements and criticality deployed at the edge tier using the smart city framework. We explore a way to

improve network resiliency at the edge using low-energy, energy-efficient ways.

The author provided a smart city framwork that can solve the smart city software requirements and the challenges faced in a smart city research project (RQ4.1 - § 4.3 and § 4.4). The smart city infrastructure can further be shared between multiple organisations and smart city research projects to use the resources and reduce costs efficiently (RQ4.2 - § 4.5). The author also evaluated the other solutions that exist, and how are they compared to our work (RQ4.3 - § 4.8). Contribution to the knowledge in this chapter (C3) is the smart city framework that can be used by future research projects to create a secure, resilient infrastructure and collect, analyse and visualise urban data. Further, we also explored the possibility of sharing the infrastructure between research projects to improve resource utilisation and reduce costs.

CHAPTER 5

IMPROVING THE NETWORK RESILIENCE OF SHARED IOT EDGE USING ADAPTIVE AND RESILIENT MULTI-COMMUNICATION NETWORK[1]

IoT devices are everywhere sensing, collecting data and providing information to make a better-informed decision about the environment (chapter 2). Many safety-critical IoT applications such as self-health monitoring through wearable IoT devices connect to a mobile phone/local edge hub via Bluetooth, ZigBee or Wi-Fi and further send the data to a cloud service (chapter 3) or hospital central processing system through Internet [389, 390].

In the event of a network-failure, e.g., power outage or any other incidental connection failures, the network connectivity of the edge device could be disconnected temporarily, resulting in either data loss or delayed data communication [391]. Depending on the time of the day, it may take from one minute to several minutes for the edge device to regain network connectivity. On the other hand, according to a survey [240], the average amount of broadband downtime per year in the UK ranges from 25.4 hours to 168.9 hours.

However, for safety-critical applications, it is essential to maintain resilient data connectivity at all time for the delivery of a time-critical message. The LPWAN technologies have explicitly been designed to meet IoT application requirements. They are built on existing cellular systems to provide improved battery life, power efficiency and indoor and outdoor coverage area [392] at an affordable cost.

[1]We explore this subject in the research paper "*Resilient Edge*: Building an adaptive and resilient multi-communication network for IoT Edge using LPWAN and WiFi" accepted at IEEE Transactions on Network and Service Management (IEEE TNSM). The first author wrote the abovementioned paper and proposed and implemented the ideas/approaches, designs, and experiments. The other authors provided their valuable reviews and suggestions to improve the paper. Poonam Yadav and Leandro Soares Indrusiak guided the § 5.3 and § 5.4. They designed the system model and the ILP formulation.

The availability of alternate low power long-range network mediums at a meagre cost opens a new horizon of opportunities. However, LPWAN technologies also have challenges in terms of limited bandwidth; the number of messages allowed per day and payload size. On the other hand, IoT edge application requirements are defined in terms of message criticality (such as high/low priority), privacy settings, message data length, message sending frequency and user trust on a particular network. Based on the application requirements and available network medium, application traffic can be routed through a specific network medium. Further, in case of a particular network medium unavailability or failure, the application can be informed of the network state and can decide on the suitability of the network and adapt accordingly. For instance, assuming the application is sending data over Wi-Fi and because of power failure Wi-Fi is disconnected, the application can choose to send data over LTE(LTE for Machines (LTE-M)/NarrowBand-IoT (NB-IoT)), LoRa (Long Range), Sigfox and adapt parameters such as payload size and frequency accordingly.

Building network resilience can involve multiple strategy to strengthen the reliability and continuity of network systems in the face of potential disruptions. It includes using backup systems (redundancy) for both hardware and network paths, ensuring that failures can be quickly addressed. Load balancing helps by spreading network traffic across different servers or paths, preventing overload on any one part. Failover mechanisms automatically switch to backup systems when needed, keeping operations smooth. Virtualization and cloud-based solutions provide flexibility, adapting to changing needs. Security measures, regular backups, and constant monitoring with timely alerts add layers of protection. Geographic dispersion and scalability further enhance resilience. Documentation, training, and incident response plans are crucial for effective issue resolution. By combining these strategies, we can create networks that withstand various challenges, maintaining smooth operation even in dynamic and potentially challenging situations. It is important to define criticality in the context of IoT applications, it refers to assessing the significance of messages within a network. It involves determining the importance of specific information or data transmitted by IoT devices. Understanding the criticality of these messages is crucial for prioritizing resources, ensuring reliable communication, and mitigating potential risks. In IoT systems, critical messages may include data related to safety, security, or vital operations, and their proper handling is essential to maintain the overall functionality and resilience of the network.

This chapter's motivation is to understand if we can achieve network resiliency at the Edge using LPWAN and Wi-Fi for time-critical IoT applications. Therefore in this work, we propose a hypothesis that using LPWAN technologies and Wi-Fi, we can achieve network resiliency at the edge IoT device by providing a capability to choose a suitable network medium based on the application requirements. In our work, we focus on improving the network resilience using backup systems (hardware and network path) and also failover mechanisms that switch to other mediums. For the implementation, we utilise affordable, readily-available MicroPython enabled,

multi-network micro-controller Pycom FiPy board [393] providing connectivity to Bluetooth, Wi-Fi, LoRa, LTE (CAT-M1/NB-IoT) and Sigfox.

The remainder of this chapter is organised as follows: § 5.1 provides the research questions and our approach. § 5.2 provides a technical background about the different LPWAN technologies. In § 5.3, we define the adaptive *Resilient Edge* to meet the application resiliency requirements by providing two example applications. In § 5.4, we formulate a criticality-aware QoS allocation problem using Integer Linear Programming (ILP) and bin packing algorithms. § 5.5 provides the implementation details of *Resilient Edge* prototype and evaluate the baseline metrics. In § 5.6, we perform the evaluation of our prototype and discuss hardware and network limitations. In § 5.7 and § 5.8, we present related work and conclusion, respectively.

5.1 Research Questions and Approach

Research Questions

This chapter attempts to investigate the following research questions:

RQ5.1 What are the different resiliency requirements for different applications using shared IoT edge networks and understand and evaluate the state-of-the-art LPWAN technologies in terms of their bandwidth, latency, throughput and maximum packet size?

RQ5.2 Can we identify and compare resource management approaches that consider QoS requirements at multiple levels of criticality?

RQ5.3 Can we define an adaptive system to meet application resiliency requirements using low power, energy-efficient networks such as LPWAN technologies; also provide an open-source implementation of *Resilient Edge* and detailed insights considering hardware and network limitations?

Research Approach

To answer the first research question (RQ5.1), the author provided two sample applications that support assisted living facilities, such as the health of residents (HealthApp) and the monitoring of residential units (HomeApp). Applications are defined in terms of message size, the minimum time interval between subsequent messages, and the application's criticality. We also performed experiments to determine various performance metrics (max payload length, the possibility of sending continuous data, latency, throughput, time to connect and reconnect) stated and observed in the wild. The applications are deployed on the edge devices using the applications management module of smart city framework (chapter 4). We formulated a critically-aware QoS allocation problem with an ILP formulation to answer the second research question (RQ5.2). We used

simple bin-packing algorithms to provide multiple levels of criticality. For the third research question (RQ5.3), we developed *Resilient Edge* on real off-the-shelf hardware and using real network deployments.

5.2 Background

In this section, we provide a background on multi-mode communication network technologies such as LPWAN technologies (LoRa, Sigfox, LTE (CAT-M1/NB-IoT)) and Wi-Fi that we use to provide resilience through redundancy in the *Resilient Edge* end-to-end system as shown in Figure 5.1. We provide a brief introduction to the technology, its range, use-case, security and energy-efficiency. We also provide various performance metrics (max payload length, the possibility of sending continuous data, latency, throughput, time to connect and reconnect) stated and observed in the wild in § 5.5.1.

5.2.1 LPWAN Technologies

LoRa

LoRa is an RF modulation technology for low-power, wide area networks (LPWANs) protocol developed by Semtech. It has a range of up to 5 KM in urban areas and up to 15 KM or more in rural areas (line of sight) [394]. LoRa is suitable for specific use cases having requirements of long-range, low power, low cost, low bandwidth, secure with coverage everywhere. For example, measuring water flow using a water flow meter [395] sending data over LoRa.

A LoRa based network consists of end devices, gateways, a network server, and application servers. End devices send data to gateways (Up link (UL)) using single-hop LoRa or Frequency-shift keying (FSK) communication. The gateways send the data to the network server via a secured Internet Protocol (IP) connection, which, in turn, passes it on to the application server. Additionally, the network server can send messages (either for network management or on behalf of the application server) through the gateways to the end devices (Down link (DL)). LoRa allows intermediate gateways to relay messages between the end-devices to the network server, which routes it to the associated application server. Communication between the end-devices and gateway is performed on different frequencies and data rates, which is a trade-off between message length, communication range [396]. The data transfer from the end device to the application server is encrypted using Advanced Encryption Standard (AES) [397].

From the energy-efficiency perspective, LoRa devices have three classes [398]. Class A device can send data anytime and opens two receive windows after one and two seconds after an UL transmission. They are the most energy-efficient; however, the DL is only available after transmission. Class B is energy efficient with latency controlled DL. They utilize time-synchronized beacons transmitted by the gateway to sync up receive windows. Class C is not efficient in

terms of power as they keep the receive window open after transmission [399]. LoRa also implements Adaptive Data Rate (ADR) by managing the data rate and RF output for each end-device individually to maximize battery life and maintain network capacity.

Sigfox

Sigfox uses publicly available and unlicensed bands to exchange radio messages over the air (868-869 MHz and 902-928 MHz). It uses Ultra-Narrow Band (UNB) technology combined with differential binary phase-shift keying (DBPSK) and Gausian FSK (GFSK) modulation. It has a range of approximately 10 km (urban), 40 km (rural). Sigfox mainly caters to IoT applications allowing small messages. For example, a letterbox sensor [400] sending a message to the user on receiving a post.

The end-device sends the message to the base stations (gateways), which forwards it to the Sigfox backend via a backhaul (3G/4G/digital subscriber line (DSL)/Satellite). The backend stores the messages to be retrieved by the end-user via browser/Representational state transfer (REST) API or set up a callback. For achieving high QoS, the end-device sends the message at a random frequency and then sends two replicas on different frequency and time (time and frequency diversity). The message can be received by any number of base stations (spatial diversity). However, Sigfox does not provide any authentication or encryption for the message and device [397].

From an energy-saving perspective, the end-device does not require pairing or sending synchronization messages to send the message, thus increasing battery life [401].

LTE (CAT-M1/NB-IoT)

NB-IoT is a 3rd Generation Partnership Project (3GPP) radio technology standard designed for extended range operation, higher deployment density, and in-building penetration. It utilizes 180 kHz bandwidth and is deployed in-band, guard-band, or standalone mode. On the other hand, LTE-M provides high latency communication, support for extended coverage, LTE-M half-duplex mode/full-duplex, short message service (SMS), coverage enhancement, connected mode mobility [402]. Both NB-IoT and LTE-M have a range of approx. 1 km (urban) and 10 km (rural) [397]. Both follow 3GPP standards and have LTE encryption by default.

NB-IoT is suited for static, low throughput, and low power applications. For example, Nortrace tracked sheep's location and well-being over mountainous regions using NB-IoT [403]. In contrast, LTE-M is best for applications requiring mobility, voice, and SMS [404]. For example, Telstra tracks the location of high-value, non-powered assets, such as shipping containers, semi-trailers, rail freight wagons, and large machinery using LTE-M [405].

From the energy-efficiency perspective, both include Power Saving Mode (PSM) and Extended Discontinuous reception (eDRX). PSM reduces the energy used by User Equipment (UE) which defines how often and how long the UE will be active to send and receive data. eDRX improve

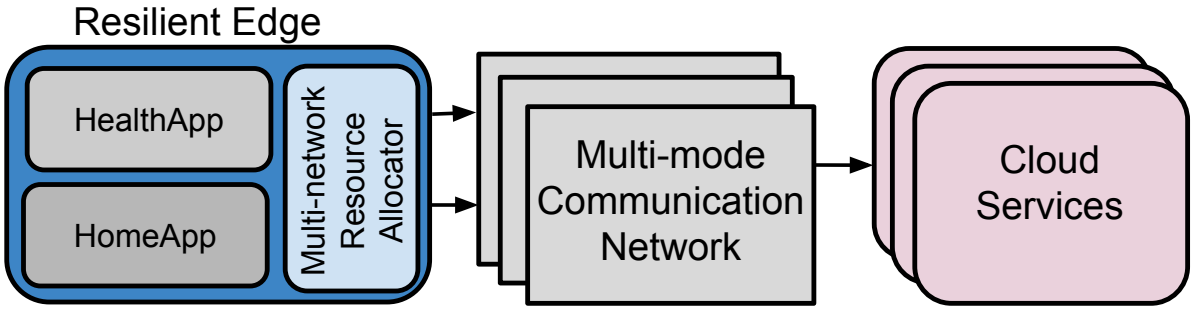

Figure 5.1: Resilient Edge End-to-end System.

end-device life for mobile-terminated traffic by switching off the receiver circuit for a defined period [406].

Integration - LoRa/Sigfox

NB-IoT/LTE-M is an IP-based network allowing data to be transferred to its associated cloud server. However, in the case of the LoRa and Sigfox, data is sent to The Things Network (TTN) console and Sigfox backend, respectively. Currently, TTN console and Sigfox backend provide multiple integration methods to retrieve data such as AWS IoT, AllThingsTalk, Microsoft Azure IoT Hub, HTTP, Emails, and other callbacks.

5.2.2 Wi-Fi

Mostly IoT devices have a low-cost, low-power system on a chip micro-controller with integrated Wi-Fi and dual-mode Bluetooth. Wi-Fi on IoT devices support different wireless modes such as 802.11 b/g/n/e/i, provide automatic beacon monitoring and scanning. The Pycom FiPy board used in our prototype has a Wi-Fi radio system on chip with $1KM$ Wi-Fi range.

5.3 System Model and Motivating Example

To better understand the network requirements of *Resilient Edge* applications, we start by considering a use-case with a concrete example. Figure 5.1 and 5.2 show an edge device running two sample applications to support assisted living facilities: one of the applications monitors the health of the resident (HealthApp), the other monitors their residential unit (HomeApp). In the real-world setting, the similar new applications can be configured for the data rates defined by the application QoS requirements and maximum network bandwidth availability. To achieve continuous network connectivity needed by these applications, we make use of a multi-mode communication network (details are provided in the next section).

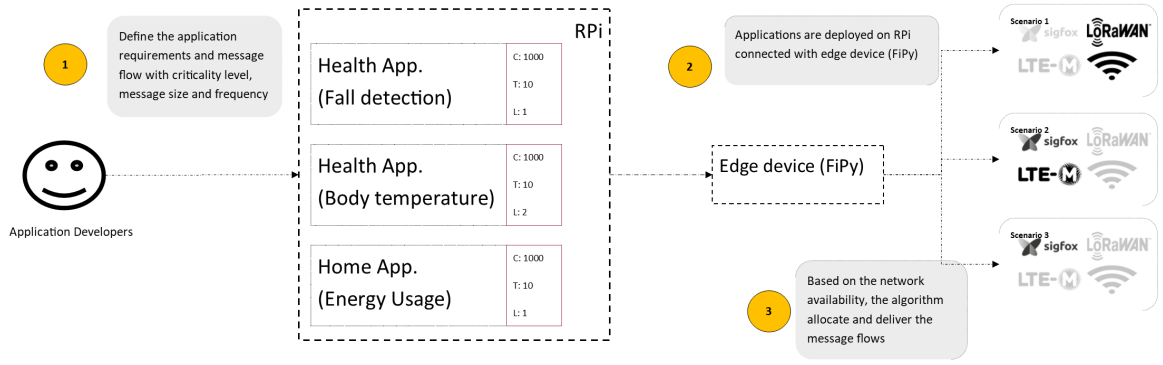

Figure 5.2: System model summary with different applications with criticality, message size and frequency defined by application developers and different network availability scenarios (Faded symbol represents network unavailability).

We now present an abstract system model defining the attributes of application data flows so that we can reason about the QoS needs of each application, and about ways to (partially) fulfill those needs under different scenarios and different levels of multi-network connectivity. We propose that the communication needs of specific applications must be explicitly declared as message flows. An application can declare an arbitrary number of message flows, and each message flow represents a potentially infinite series of messages to be sent through one of the local network interfaces. To allow application developers to quantitatively declare the QoS needs for each message flow, we revisit the notion of mixed-criticality communication proposed in [407] and support the definition of QoS requirements at distinct levels of criticality. As in [407], our goal is to allow the system to guarantee a predefined level of service for all message flows during normal operation, but also provide graceful degradation of service in adverse circumstances by allowing the most critical communication to be maintained. Unlike [407], however, we are not interested in meeting hard real-time deadlines and will instead use the notion of criticality-specific QoS requirements to manage multi-network resources.

Our model allows system designers and administrators to decide how many levels of criticality $L = L_{max}$ to support, and then to allow the specification of the QoS requirements of each message flow at each of those levels. The *Resilient Edge*, as shown in Figure 5.1 is designed to support three levels of criticality, and the Table 5.1 shows the QoS required by each message flow at each level. The message flows in Table 5.1 have been defined by taking a bottom to top approach. We assume that high criticality level messages are necessary to be delivered messages and are rare and have smaller size. In this example, for message flow 1, when criticality level 3 is requested and served, underlying network interface guarantees a message delivery service with message size of 10 bytes with 60 seconds subsequent message interval. The applications (designed by application developers) can request any message size and message interval; however, the values in our example reflect the prototype application data size and are intuitively set by authors

Table 5.1
Application message flows on an edge device for assisted living facilities.
C: maximum message size (bytes)
T: minimum interval between subsequent message (seconds).

	Criticality Level	**1**		**2**		**3**	
Applications	**Message Flow τ**	**C**	**T**	**C**	**T**	**C**	**T**
HealthApp	1 fall detection	1000	10	40	20	10	60
"	2 heart monitoring	1000	5	80	10	10	20
"	3 body temperature	30	30	10	120		
HomeApp	4 sensor bedroom	40000	10	10	30		
"	5 sensor bathroom	80	10	10	30		
"	6 sensor lounge/kitchen	40000	10	10	30		
"	7 sensor front door	40000	10	10	30		
"	8 energy usage	40	3600				

considering several state-of-the-art IoT applications. Additionally, the message flows and the requirements are driven by the network capacity available on the edge device (e.g., FiPy[393] in our prototype).

We define $L = 1$ as the criticality level denoting normal operation mode, so the QoS requirements at that level should declare the largest communication volumes and injection rates of each message flow to account for all critical and non-critical traffic. QoS requirements at higher levels of criticality ($L = 2$ and $L = 3$) should only be declared for message flows that carry critical data and should account only for the necessary communication volumes and injection rates at each of those levels. By declaring or not a QoS requirement at a given level, application developers can explicitly distinguish the criticality of each message flow, and to explicitly define a number of service degradation levels each of them can support.

We can now define a message flow τ_i as a tuple (A_i, C_i, T_i) where A_i denotes the application to which the message flow belongs to, C_i denotes the maximum message size (in bytes) and T_i denotes the minimum interval between subsequent messages of the flow (in seconds). The bandwidth utilisation U_i of a flow τ_i can be calculated by the quotient C_i/T_i.

To support multiple criticality levels, C_i and T_i are defined as arrays of length L_{max}, so C_i^L and T_i^L denote, respectively, the maximum message size and the minimum interval between subsequent messages of τ_i at criticality level L.

In normal operation (i.e. $L = 1$), message flows declare their most generous QoS requirements, with larger data volumes for home monitoring (e.g. including camera snapshots in most of them) and resident monitoring (e.g. detail accelerometer data for fall detection, full (electrocardiogram) ECG data for heartbeat monitoring). The next criticality level (i.e. $L = 2$) allows the declaration of degraded QoS levels, which in this example is provided for all message flows except for the one monitoring energy usage (which will not be forwarded by the edge device in case of degraded

Table 5.2
System Model Nomenclature

Symbol	Meaning
L	criticality
τ_i	message flow as a tuple (A_i, C_i, T_i)
A_i	application to which the message flow belongs to
C_i	maximum message size (in bytes)
T_i	minimum interval between subsequent messages of the flow (in seconds)
U_i	bandwidth utilisation
C	maximum message size (bytes)
T	minimum interval between subsequent message (seconds).

service). Notice that the QoS requirements declared for $L = 2$ and show that monitoring will be performed less often and less data will be provided (e.g. simple movement detectors for home monitoring, average temperature and heartbeat for health monitoring). Finally, only two message flows declare QoS requirements at the highest level of criticality (i.e. $L = 3$), representing the alarms for fall or severe arrhythmia/cardiac arrest. In the case of degraded service, all available resources should be used to provide those two flows with their declared QoS requirements.

5.4 Multi-Network Resource Management

Given the system model proposed in Section 5.3, we can now formulate a criticality-aware QoS allocation problem.

A straightforward way to ensure QoS to the application message flows is to prevent the over-utilisation of the network interfaces they are assigned to. For example, by providing criticality level $L = 2$ guarantees to all message flows of the HealthApp application from Table 5.1 it would be possible to allocate all of them to a LoRa network (as their compound bandwidth utilisation would not exceed 6 bps), but the same network would be over-utilised if flows operate at criticality level $L = 1$ (where their compound bandwidth utilisation would exceed 1700 bps).

We can therefore formulate the criticality-aware QoS allocation problem as the choice, for each message flow of each application, of its allowed criticality level of service and its allocated network interface. Such problem is similar to a Variable Size Bin Packing Problem (VSBPP) [408], but with a fixed number of bins (i.e. the different networks, each of them with their bandwidth and payload size limitations) and with a choice of sizes for each element (i.e. the message flows, with their choice of criticality level).

5.4.1 ILP Formulation

Similarly to the standard VSBPP, we can formulate our problem with an ILP model. For the sake of simplicity, we describe the size of bins and elements by their bandwidth capacity and

utilisation, respectively. We claim that an extension to a multi-dimensional formulation (i.e. that can also capture maximum payload sizes, maximum number of daily messages, etc.) is straightforward but left as future work. The assumption is that the QoS requirements of all applications can be satisfied provided there is enough bandwidth of one network or combined bandwidth of multiple networks. However, in practice, the network capacity is limited, and there would be applications whose QoS requirements cannot be satisfied.

Let us then consider a set $\mathscr{T}$ of elements representing our message flows $\tau_i, i = 1...n$, each of them with a potential choice of values U_i^L representing the different bandwidth utilisations C_i^L/T_i^L at each level of criticality they are designed to support (or ∞ if that flow does not specify service at a given criticality level, e.g. *energy usage* flow at levels $L = 2$ and $L = 3$ in Table 5.1).

Likewise, let us consider a set Γ of bins representing our network interfaces $\gamma_j, j = 1...m$, each of them with a bandwidth capacity B_j. Finally, we define a set of binary variables $x_{i,j,L} \in \{0,1\}$, and assume that $x_{i,j,L} = 1$ if message flow τ_i is assigned to network γ_j and configured to operate at criticality level L, or $x_{i,j,L} = 0$ otherwise. Given the ranges $1 \leq i \leq n,\ 1 \leq j \leq m,\ 1 \leq L \leq L_{max}$ we will have at most $n \times m \times L_{max}$ binary variables for a given problem.

To ensure the assignment of values to the binary variables represent a valid solution to our problem, we must now state a number of constraints. First, we make sure that a message flow τ_i is allocated to a single network interface and configured to operate at a single criticality level by stating that $\sum_{j=1}^{m} \sum_{L=1}^{L_{max}} x_{i,j,L} = 1$ for all $1 \leq i \leq n$. Secondly, we ensure that no network interface γ_j is overloaded by stating that $\sum_{i=1}^{n} \sum_{L=1}^{L_{max}} x_{i,j,L} \times U_i^L \leq B_j$ for all $1 \leq j \leq m$.

Finally, we can state our maximisation objective function as:

$$objective = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{L=1}^{L_{max}} x_{i,j,L} \times (1 + L_{max} - L) \tag{5.1}$$

The rationale behind the maximisation of the objective is to configure message flows at the lowest possible levels of criticality (i.e. lowest values for L), thus providing each message flow with the most generous possible QoS, while avoiding network overload. The unit added to the last term of the equation is crucial to allow the objective to distinguish a flow that is allocated at the highest criticality and one that is not allocated at all.

An additional constraint could be formulated, in case all message flows must be allocated to a network interface and receive some level of service: $\sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{L=1}^{L_{max}} x_{i,j,L} = n$. This is not always necessary or desirable, as it may the intention of application designers that, under limited network availability, only a subset of the application message flows should be provided service (e.g. in the example from Table 5.1, under the most stringent conditions at $L = 3$, only the *fall detection* and *heart monitoring* message flows require service). In such cases, such a constraint may be rewritten to ensure that specific message flows are always allocated service, or even be reformulated as part of the objective function, aiming to maximise the number of message flows that are guaranteed some level of service.

5.4.2 Bin-Packing Algorithms

While the formulation given in subsection 5.4.1 can be optimally solved by an ILP solver, it may not be reasonable to expect that such a software package could be installed and executed by resource-constrained edge devices such as the ones considered in this work. We, therefore, propose the use of simple bin-packing algorithms that are able to achieve acceptable results with a much lower computational overhead. In particular, we define a criticality-aware best fit (CABF) algorithm and show its performance compared to classic first fit, best fit, and worst fit algorithms (FF, BF, and WF) as well as their decreasing variants (FFD, BFD, and WFD).

Since the classic algorithms are unaware of the different criticality levels, we implemented two alternatives for each of them, one that tries to fit message flows to networks at their highest level of criticality (i.e. H-FF, H-BF, H-WF and their decreasing counterparts) and another that does the same with the lowest defined criticality of each message flow (i.e. L-FF, L-BF, L-WF and their decreasing counterparts).

Algorithm 1 describes the proposed CABF algorithm, which takes as inputs the sets $\mathcal{T}$ of message flows and Γ of networks, and outputs a set Q of 3-tuples $q = (\tau_i, \gamma_j, L)$, each of them representing the allocation of a message flow τ_i to a network γ_j at criticality level L. Algorithm 1 uses the following notation: $q(\tau)$, $q(\gamma)$ and $q(L)$ denote the first, second and third element of a 3-tuple q, and likewise $Q(\tau)$, $Q(\gamma)$ and $Q(L)$ denote the sets of all first, second and third element of the 3-tuples in Q; $\mathcal{T}_L$ is the subset of $\mathcal{T}$ including all message flows that are declared at a given criticality level L (as not all flows must be declared for all levels); and $BestFit(\tau, L, \Gamma)$ denotes a function which returns the network $\gamma \in \Gamma$ which is the best fit allocation for the message flow τ at its criticality level L, or $\emptyset$ if τ does not fit in any of the networks in Γ, taking into account the allocations already in Q.

The intuition behind the CABF algorithm is as follows. It tries to allocate first the message flows defined at higher criticalities, as shown by the outer *for* loop decreasing from L_{max} to 1. As it iterates over that loop towards lower criticality levels, and before it allocates flows defined at the criticality level of the current iteration, it first attempts to lower the criticalities of flows allocated in the previous iterations. This is shown by the first inner *forall* loop, which iterates over tuples in Q with flows that have definitions at the criticality level of the current iteration (i.e. $q(\tau) \in \mathcal{T}_l$). Within the first inner *forall* loop, the algorithm removes the original allocation from Q, then tries to find a network γ_{reloc} which is the best fit for the flow using its lower criticality figures. If the best-fit algorithm succeeds to find an allocation with the lower criticality values, a 3-tuple representing that new allocation is added to Q. If it fails to find a network that is able to accommodate the requirements at a lower criticality, the original allocation is returned back to Q. Once the first inner *forall* loop finishes, the second inner *forall* loop uses the best-fit algorithm to allocate, when possible, all unallocated message flows that have definitions at the criticality level of the current iteration.

The proposed order of the two inner *forall* loops reflects an assumption that flows that have

Algorithm 1: Criticality-Aware Best Fit (CABF)

Result: Set of 3-tuples indicating the allocated network and configured criticality level for all message flows that can be provided service

CABF $(\mathcal{T}, \Gamma)$

inputs : set $\mathcal{T}$ of message flows, set Γ of networks

output : set Q of 3-tuples $q = (\tau_i, \gamma_j, L)$

$Q \leftarrow \emptyset$;

for $(l = L_{max};\ l > 0;\ l = l - 1)$ **do**

  foreach $(q \in Q \mid q(\tau) \in \mathcal{T}_l \wedge q(L) > l)$ **do**

    $Q \leftarrow Q - q$;

    $\gamma_{reloc} \leftarrow BestFit(q(\tau), l, \Gamma)$;

    if $\gamma_{reloc} \neq \emptyset$ **then**

      $q \leftarrow (\tau, \gamma_{reloc}, l)$;

    $Q \leftarrow Q + q$;

  foreach $(\tau_{new} \in \mathcal{T}_l \mid \tau_{new} \notin Q(\tau))$ **do**

    $\gamma_{new} \leftarrow BestFit(\tau_{new}, l, \Gamma)$;

    if $\gamma_{new} \neq \emptyset$ **then**

      $Q \leftarrow Q + (\tau_{new}, \gamma_{new}, l)$;

return Q;

definitions at higher levels of criticality should always be given more resources if they become available. This will not always be the case in every application domain, and in many cases it may be better to first use resources to provide some service to less-critical message flows rather than improve the service to highly-critical ones. Reversing the proposed order of the two inner *forall* loops would achieve exactly that, therefore we name that variant $CABF_{inv}$.

5.4.3 Evaluation - Motivating Example

Table 5.3 shows the network allocations and choice of criticality level for each of the message flows of the motivating example described in Section 5.3. The table shows allocations produced by each of the baseline bin-packing algorithms, by both variants of the proposed algorithm, and one solution (out of many possible ones) produced by an optimal solver. The allocations assume the availability of three networks with bandwidths of 64000, 1760 and 48 bits per second, representing Wi-Fi, LoRa SF9 and Sigfox networks (but disregarding maximum payload size or the number of daily messages), and represented by the symbols $*$, # and +, respectively.

Both variations of the proposed algorithm are able to produce optimal solutions in this example, providing service to all flows, with all-but-two at their lowest criticality level (which leads to an objective result of 22 according to Equation 5.1).

Table 5.3
Obtained criticality level (1 | 2 | 3) and network allocation (* Wi-Fi | # Lora | + Sigfox) for motivating example

	Message Flows								**% flows served**	**avg crit level**	**objective**
	1	2	3	4	5	6	7	8			
Requested crit level	1,2,3	1,2,3	1,2	1,2	1,2	1,2	1,2	1			
Allocation algorithms	**Allocated Criticality Level**										
L-FF	1*	1*	1*	1*	1*			1*	75	1	18
L-FFD		1#	1#	1*	1#	1*		1+	75	1	18
H-FF	3*	3*	2*	2*	2*	2*	2*	1*	100	2.12	15
H-FFD	3*	3*	2*	2*	2*	2*	2*	1*	100	2.12	15
L-WF	1*	1*	1*	1*	1*			1*	75	1	18
L-WFD		1#	1#	1*	1#	1*		1+	75	1	18
H-WF	3*	3*	2*	2*	2*	2*	2*	1*	100	2.12	15
H-WFD	3*	3*	2*	2*	2*	2*	2*	1*	100	2.12	15
L-BF	1#	1*	1+	1*	1#			1+	75	1	18
L-BFD		1#	1+	1*	1#	1*		1+	75	1	18
H-BF	3+	3+	2+	2+	2+	2+	2+	1+	100	2.12	15
H-BFD	3+	3+	2+	2+	2+	2+	2+	1+	100	2.12	15
CABF	2+	1#	1+	2+	1#	1*	1*	1+	100	1.25	22
CABFinv	1#	2#	1+	2+	1#	1*	1*	1+	100	1.25	22
Optimal	1*	1*	1*	1*	1*	2*	2*	1*	100	1.25	22

5.5 Implementation: Resilient Multi-network Edge Platform

This section describes a resilient multi-network edge platform we have designed and implemented, aiming to validate the concepts proposed in the previous sections over real off-the-shelf hardware and using realistic network deployments. Firstly, we describe the chosen hardware platform based on a Raspberry Pi and a Pycom FyPy communication board. The FyPy board supports multi-network connectivity over five different networks Wi-Fi, Bluetooth, LoRa, Sigfox and LTE (CAT-M1/NB-IoT). The Raspberry Pi, in turn, handles the proposed multi-network resource management approaches. A detailed description of the functionality of each board, and their integration will be provided, as well as their inter-operation with the cloud to perform realistic data transfer scenarios. Additionally, we will provide details about key performance metrics that show the strengths and weaknesses of each type of network supported by the platform, namely maximum payload length, inter-message gap, latency, throughput, connection and reconnection time. The detailed description of our multi-network edge platform is then followed by a practical evaluation, where we implement the proposed multi-network resource management algorithms from Section 5.4.

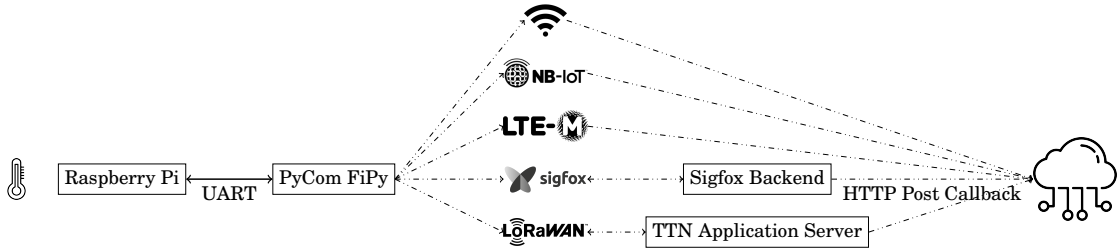

Figure 5.3: Block diagram of the current experimental setup.

Platform Overview

The *Resilient Edge* prototype setup is shown in Figures 5.3 and 5.4. A Raspberry Pi model 4 [409] (RPi) is interfaced with Pycom FiPy [393]. RPi has Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz with 4 GB RAM, FiPy has an Xtensa® dual-core 32–bit LX6 microprocessor and on-chip SRAM of 520KB and external SRAM 4MB with an external flash of 8 MB. FiPy provides connectivity to five different networks Wi-Fi, Bluetooth, LoRa, Sigfox and LTE (CAT-M1/NB-IoT). More details about the interworking of FiPy can be found in the FiPy datasheet [410]. The RPi UART (GPIO14-TXD/ GPIO15-RXD) is connected to the expansion board pins (P3-TXD/P4-RXD) of FiPy to transfer the data from RPi to FiPy. The RPi runs the KubeEdge orchestration engine (§ 4.4.1) and deploys applications using containers. We implemented and simulated the message flow of HealthApp and HomeApp on RPi using the application management module (§ 4.3.3) of the smart city framework (chapter 4) and multi-network resource allocator on FiPy, respectively. To enable data transfer between RPi and FiPy, a message payload from the applications is written to the RPi UART and read by FiPy continuously. We provide UART access to the applications running on Kubernetes by using the smart device manager [411] that enables access to device drivers on containers for K8s. On FiPy, a Python script checks the messages received from RPi, the network interface assigned to the message flow, its criticality level, and attempts to send them via that network interface.

Implementation details of RPi components

We utilise TCP/IP serial bridge[2] to create a socket listening on port 8080 connecting to the UART (/dev/ttyAMA0) to send and receive data to and from the UART. Furthermore, we use *python select lib* [3] to monitor the sockets for incoming data to be read and send outgoing data when there is room in the buffer and use message queues to store outgoing messages. To send and receive a message over UART efficiently and without breaking, we add a header with `(:ML:<MessageLength>)` at a start and a newline `'\n'` at the end. On both sides, RPi reading a socket and FiPy reading the UART, we ensure that we have received the full message. To

[2]TCP/IP - serial bridge `https://pyserial.readthedocs.io/en/latest/examples.html#tcp-ip-serial-bridge`

[3]Select - Waiting for I/O completion `https://docs.python.org/3/library/select.html`

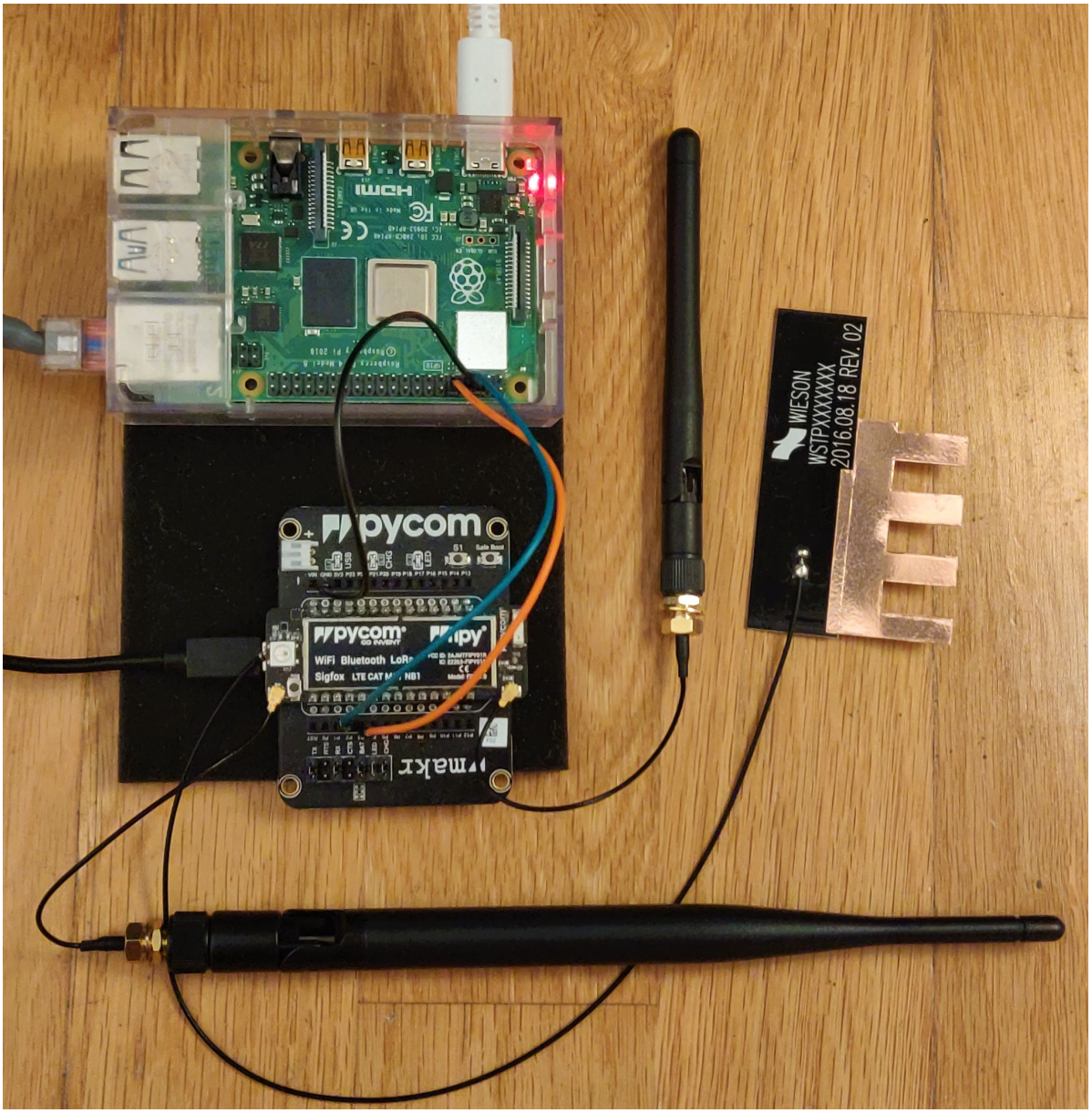

Figure 5.4: Current experimental setup.

simulate the message flows running on RPi, we use threads to write a message payload on the socket with `<MessageFlow Name, Criticality level, the payload>`. The thread sleeps for the period specified by the message flow before sending the next message. We store the statistics about the number of messages sent by a particular message flow, acknowledgement, or error received.

The FiPy runs a multi-network resource allocator and sends an allocation message back to the RPi stating which message flows have been assigned with which criticality level. A sample Message Flow Element Allocation (MFEA) message is:

`MFEA:['PS': 41, 'N': 'Wi-Fi', 'PE': 10, 'MF': 'Kitchen Sensor', 'CL': 1]`
where PS is payload size, N is network assigned, PE is the period (time in seconds) between two subsequent messages, MF is message flow name, and CL is criticality level assigned. If the network conditions at the FiPy are changed, and a new MFEA message is received, the previous thread sending the messages are stopped, and new threads for sending a message with a particular criticality level and period are launched.

Implementation details of FiPy components

On FiPy, before assigning any network to a message flow, we need to create a network `"bin"` of the available networks (Wi-Fi, LTE (CAT-M1/NB-IoT), LoRa and Sigfox) and add the corresponding network interfaces to the network `"bins"`. While doing so, we take into some limitations that are posed by underlying hardware such as if Wi-Fi is available, we do not add a network `"bin"` for LTE (CAT-M1/NB-IoT) because in the current version of FiPy if both Wi-Fi and LTE (CAT-M1/NB-IoT) are connected at the same time, FiPy does not provide routing capabilities to direct the traffic [412]. If Wi-Fi is unavailable, then we connect via LTE (CAT-M1/NB-IoT). Similarly, if the LoRa network is available, we add LoRa to the network `"bin"`. If LoRa is unavailable, we add Sigfox, mainly because Sigfox and LoRa share the same radio module. As part of the Multi-network resource allocator - we implement variant Criticality-Aware Best Fit ($CABF_{inv}$) and set the initial parameters, and perform the allocations of the message flows to the network interface. After the allocations, we continuously read the UART for the messages from the RPi. The messages from the RPi are in the format of `<MessageFlow Name, Criticality Level, Payload>`. On the FiPy, we check if the Message Flow with criticality level has been assigned; if assigned, an attempt to send the payload is made. If the message flow is not allocated, an error message is sent back to RPi, mentioning message flow is not allocated. Similarly, if the payload is delivered, an ACK message is sent to the RPi; if not delivered, an error message with not delivered is sent through UART.

FiPy provides multi-network connectivity, and powering on all the network interfaces could result in significant power consumption. With that in mind, currently, we initialize all the network interfaces at the boot and connect to a specific network based on network availability and conditions. For instance, the NB-IoT connection is skipped if the Wi-Fi network is available; if the LoRa network joins successfully, the Sigfox socket is not created. Further, we have mentioned the time (in seconds) for different technologies to connect to the network (§ 5.5.1) and time complexity and context switching of the $CABF_{inv}$ algorithm (§ 5.6.2) that provides a rough estimate on switching overhead if the IoT devices need to switch between network interfaces and to turn on/off the interface.

The multi-network resource meets the QoS requirements of IoT applications by determining the different network interfaces available and the communication parameters of the selected

Table 5.4
Summary of baseline metrics

Metrics	LoRaWAN	SigFox	LTE (CAT-M1/NB-IoT)	Wi-Fi
Max-payload length	1 - 222 bytes	1-12 bytes	UDP/TCP/IP	UDP/TCP/IP
Sending continuous data	0.165 ms	10.5 s	1-100 ms	1-100 ms
Latency	24 - 2800 ms + 1-100 secs	1 - 4.5 s	500 ms (avg)	8 ms (avg)
Throughput	250 - 11000 bps	UL: 100 bpsDL: 600 bps	NB-IoT: UL: 66 kbps; DL: 26 kbpsLTE-M: DL: 300 kbps; UL: 380 kbps	Local: 3550 KbpsRemote: 770 Kbps
Time to connect to network	OTAA: 5.6 sABP (join not required)	1-100 ms	With LTE Reset: 20 sWithout LTE Reset: 15.5 s	7.7 s

technology (bandwidth). For instance, when a network interface is defined (whether it is available or not), we determine the bandwidth provided by that network. For instance, LoRa starts the connection with adaptive Spreading Factor (SF), i.e., it would start with SF7; if it did not connect with SF7, it would try with SF8 and so on. Based on the connection, we take the bandwidth of the connected SF.

Receiving messages on Cloud

To store the messages sent by the FiPy (as shown in Fig. 5.3), we utilise Tornado - a python web framework and asynchronous networking library [4] to run an `HTTP` server on a machine hosted on a cloud. The `HTTP` server accepts `HTTP POST` messages and receives them directly from the FiPy via Wi-Fi, TTN application server via LoRa, Sigfox backend via Sigfox and Pybytes[5] via NB-IoT. When the message flow allocated interface is Wi-Fi, an `HTTP POST` request is sent from FiPy to the cloud machine using `urequests micro-python library`. When the assigned interface for message flow is LoRa, Sigfox and NB-IoT, the message is sent via the respective interface. On TTN application server, Sigfox backend and Pybytes for NB-IoT, we have configured the `HTTP` Integration as defined in § 5.2.1. HTTP integration sends the UL data received from FiPy to our cloud machine. The `HTTP` server checks for the `URI` and fetch the data from the post data and stores it in a `influxdb` database.

5.5.1 Platform Metrics

For performance evaluation, we considered the following metrics: maximum payload length, inter-message gap, latency, throughput, time to connect and reconnect, and performed the initial experiments to get the baseline results for each network (LoRa, Sigfox, NB-IoT, Wi-Fi) before deploying the multi-network resource allocator on FiPy. In Table 5.4, we provide a summary of the metrics found in these baseline experiments. Application developers can decide on the suitable network medium for the application based on the application requirements and the use-case. First, we provide how each network compares with each other, followed by more information about the experiment.

[4]Tornado Web Server `https://www.tornadoweb.org/en/stable/`
[5]Pybytes `https://docs.pycom.io/pybytes/`

Table 5.5
LoraWAN airtime for max payload in Europe [1]

Configuration (kHz)	**Bitrate (bits/sec)**	**Max payload size (bytes)**	**Time on Air (ms)**	**Max number of messages/day**
SF12/125	250	51	2793.5	12
SF11/125	440	51	1560.6	23
SF10/125	980	51	698.4	51
SF9/125	1760	115	676.9	53
SF8/125	3125	222	655.9	54
SF7/125	5470	222	368.9	97
SF7/250	11000	222	184.4	195

Maximum Payload Length

Maximum payload size determines how much information (in bytes) can be sent in one message and helps to determine the suitability for an IoT application. For *LoRa*, the max payload size varies from 51 bytes to 222[6]/242 bytes based on the configuration settings. On the other hand, *Sigfox* allows an UL payload of up to 12 bytes and a limit of up to 140 messages per day bytes payload with a limit of 4 messages per day DL. Most suitable from the payload perspective, is *NB-IoT*/*Wi-Fi*. LTE Transport block sizes (TBS) can support a maximum of 85 bytes DL and 125 bytes UL. However, as TCP/UDP protocol is used in Wi-Fi/NB-IoT, the payload is sent as multiple packets (the size and number of which depend on the path Maximum Transmission Unit (MTU)). So, the payload length for NB-IoT/Wi-Fi is bounded by the memory assignment capability of the device.

Table 5.5 represents the max payload sizes with max number of messages per day at different SF/bandwidth and respective airtime for LoRa [414]. We use TTN, a public community network having a fair access policy [415] that limits the UL airtime to 30 seconds per day per node and the DL messages to 10 messages per day per node. The max number of messages in Table 5.5 is calculated based on the 1 percent duty and the fair usage policy with maximum payload message. Further, to utilize application payloads efficiently, LoRa best practices [416] to limit application payloads can be referred. Sigfox provides Link Quality Indicator (LQI) [417] based on the RSSI and number of base stations that received a message. However, as only four DL messages per day are allowed, it is advisable to set up an HTTP/Email callback to get service-related information.

Inter-message gap

We conducted this primitive experiment to understand the limitation of the time between sending two consecutive messages. For *LoRa*, on average it took 0.165 ms to send a message. For *Sigfox*, in terms of sending a continuous message on Pycom FiPy end-device, it takes around an average

[6]The payload size is 222 bytes when the device is a repeater and requires optional FOpt control field [413].

Table 5.6
Sigfox payload time approximate time provided by Sigfox [2] and observed for average, good/excellent quality at RC1 region

	Stated	**Observed**	**Observed**
Payload Length	**Approximate (sec)**	**Average (sec)**	**Good/Excellent (sec)**
<1 bit	1.1	2	1
2 bit - 1 byte	1.2	1.6	2.0
2-4 byte	1.45	2.3	2.1
5-8 byte	1.75	4.5	2.5
9-12 byte	2	4.5	3

of 10.5 s, with the minimum 9 s and maximum 12 s to send a message on Sigfox in RC1 region with 100 bps. Suppose the application priority is to send the messages fast. In that case, sending a message via Wi-Fi and NB-IoT takes a few milliseconds.

To experiment, for LoRa, we sent 40 messages with different payloads, ten messages with four SF options offered by LoRa, i.e., (SF7 - 1 byte, SF12 - 1 byte, SF7 - 242 bytes, SF12 - 51 byte). For Sigfox, a message with a 12-bytes payload takes 2.08 s over the air with a rate of 100 bps. Further, the device emits a message on a random frequency and then sends two replicas on different frequencies and time [401]. We experimented with sending continuous data on Sigfox of variable length starting from 1 byte to 12 bytes. We measured the time before sending the message using '`socket.send(msg)`' and after that. We sent 60 (5×12) messages, three times on average LQI, and one time each on good and excellent LQI.

Latency

We define latency as the delay between transmitting a packet and its arrival at its destination. It combines transmission, propagation, and processing time at both ends. For *LoRa*, TTN latency ranges between 24 ms (smallest payload - fastest bit-rate) to 2800 ms (max-payload on slowest bit-rate) from the end-device to the gateway.

For Sigfox, Table 5.6 provides the approximate time taken by the message to reach Sigfox backend from the edge device provided by Sigfox [2] and observed time taken by payload of different sizes at different LQI (average/good/excellent) in RC1 region for Sigfox. For NB-IoT, on average, it has a latency of 576 ms. It is important to mention that when ping is used the first time, the latency is high in the range of 10 s and then stabilises slowly (after $5-10$ pings) to the range of $500-800$ ms. From literature, NB-IoT latency ranges around $1-10$ s [418] depending on normal coverage or extended coverage. Latency in LTE-M is around $100-150$ ms in normal coverage.

For *Wi-Fi*, latency has an average of 8.32 ms and 16.70 ms with a standard deviation of 9.93 ms and 12.19 ms for the machine in local and remote networks, respectively. Fig. 5.5 provides

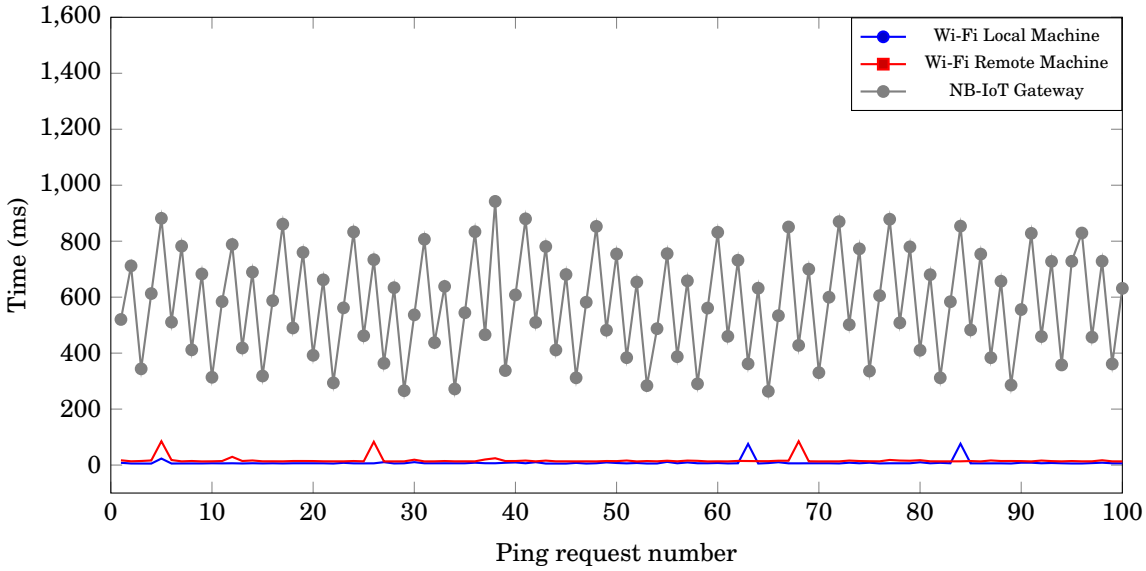

Figure 5.5: Latency results for pinging a local machine and cloud machine via Wi-Fi and gateway via NB-IoT network.

latency of the Wi-Fi network when FiPy pings a machine in the same local network and remotely in the cloud network. The network latency of NB-IoT varies significantly compared to the Wi-Fi.

For *LoRa*, the transit time from the gateway to the application completely depends on the solution implemented. On TTN and with a gateway connected through wired Ethernet, it will take tens of milliseconds (at the current load levels). If the gateway uses a slow cellular connection, the delay will increase. Further, up to a few seconds can add up based on the selected callback mechanism (HTTP, AWS IoT, others). At a high level, latency would be a sum of time-on-air, gateway to network server network latency, duplication window, routing services processing time, a selected callback to application network latency. LoRa TTN fair usage policy only allows at most 10 DL messages. If we also consider the DL latency, one or two seconds could be added to the latency as there are two receive windows after a UL message. For *Sigfox*, to understand the latency, we calculated the time when we started sending the message using the device and when it was received at the Sigfox backend. We synced the end-device time using NTP with `pool.ntp.org` server. For *NB-IoT*, we connected to the NB-IoT Vodafone network with Pycom provided subscriber identity module (SIM) [419] having pycom.io Access Point Name (APN). We figured out our IP Address using AT command 'AT+CGCONTRDP' and sent around 100 ping requests to the gateway, which was three hops away (calculated from TTL). For Wi-Fi, we connected the end-device FiPy to the home Wi-Fi network and calculated the latency by sending 100 $uping$ [420, 421] requests to a local machine in the same network and to a remote machine on a cloud.

Table 5.7
Sigfox radio configuration [3]

Frequency (MHz)	**RC1/RC3/RC5**	**RC2/RC4**
Uplink center	868.130/923.200/923.300	902.200/920.800
Downlink center	869.525/922.200/922.300	905.200/922.300
Uplink data rate (bit/s)	100	600
Downlink data rate (bit/s)	600	600

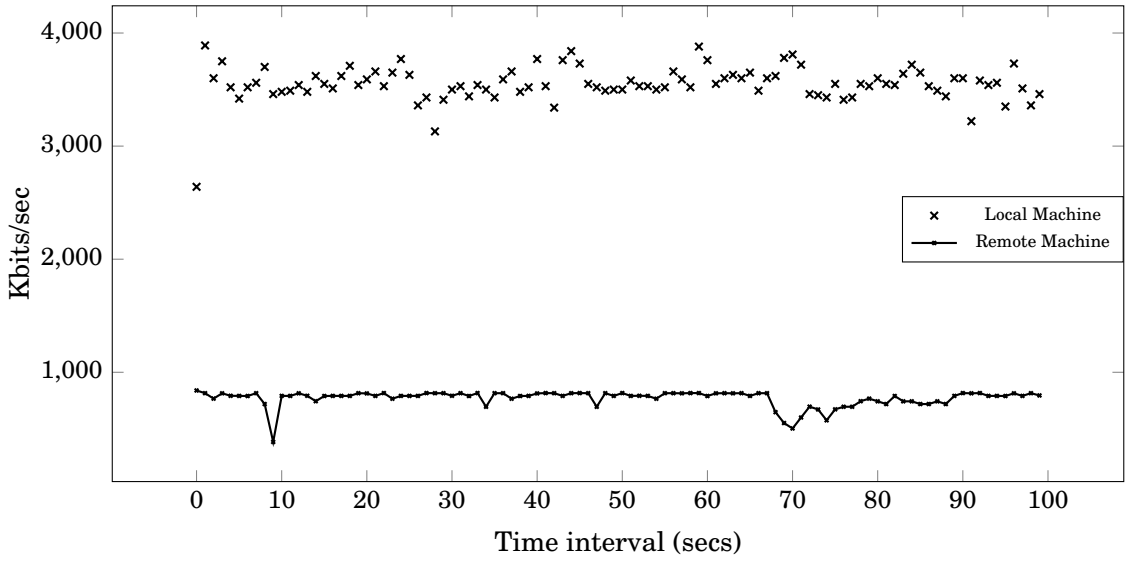

Figure 5.6: Bandwidth results using iperf when running on local network and cloud.

Throughput

This experiment measured the average throughput (bits per second) achieved on each network individually. For *LoRa*, bit-rate depends on the bandwidth and SF. In Europe, the regional parameters [422] allow a bandwidth of 125 KHz to 250 KHz and SF of $7-12$ [413]. LoRa data rates range from 0.3 $Kbps$ to 50 $Kbps$ [396]. For *Sigfox*, Table 5.7 provides UL and DL frequency and data rate for different regions. Based on the Sigfox frequency, the data rate could be determined. For *NB-IoT*, data rate [418] is 26 $Kbps$ in the DL, and 66 $Kbps$ in the UL. LTE-M has approximately 300 $Kbps$ in DL and 380 $Kbps$ in the UL in half-duplex. On an average on the field, 100 to 150 $Kbps$ are reached in both directions. For Wi-Fi, bandwidth has an average of 3550.8 $Kbps$ and 770.181 $Kbps$ with a standard deviation of 157.19 $Kbps$ and 71.86 $Kbps$ when $iperf3$ is hosted locally in the local network and cloud network, respectively. Fig. 5.6 provides the bandwidth of the Wi-Fi network when FiPy pings and connects to the `iperf` server in the same local network and remotely in the cloud network.

For our experiments, for NB-IoT, we are using Pycom provided Vodafone SIM; the User Equipment (UE) can only communicate to a white-listed IP address because of which we were

Table 5.8
NB-IoT connect times in seconds

NB-IoT		**Reset**	**Init**	**Attach**	**Connect**	**Disconnect**	**Deattach**	**Deinit**
With Reset	Avg	6.37	0	12.64	1.29	7.23	1.13	0.09
	Min	6	0	11	1	7	0	0
	Max	7	0	17	2	8	2	1
Without Reset	Avg	-	2.07	12.15	1.31	7.23	0.98	0.15
	Min	-	1	8	1	7	0	0
	Max	-	3	20	2	8	2	1

unable to host an instance of `iperf` on a server and calculate throughput. For *Wi-Fi*, Pycom FiPy utilises `ESP32` which provides 20 $Mbps$ TCP RX/TX in the test [423] performed in the lab. The bandwidth and throughput was calculated using `uiperf3` [424].

Time to connect to the network

We conducted baseline experiments to understand the connection time an end device takes to join the different networks. It helps to estimate switching overhead if the IoT devices need to switch from one network technology to another. *LoRa* allows activation by two methods Over-the-Air Activation (OTAA) and Activation by Personalisation (ABP). OTAA took on an average 5.6 s with a minimum 4 s to a maximum 7 s to join the network. On the other hand, ABP provides hard-coded session keys and allows the sending of data without joining. In case of an emergency where the device tries to send only one message and is unsure about the coverage of LoRa, the message can be sent using maximum SF12 to have a maximum range. For *Sigfox*, creating a socket for Sigfox taken an average of few ms. For NB-IoT, we present the timings for the different methods in Table 5.8. When the LTE modem is connected to the network first time, it takes a significant amount of time to connect to the network as it searches, registers itself to the network, it could take approximately 15 $mins$ to 60 $mins$ to attach to the network, whereas Wi-Fi takes approx 5.6 seconds to connect to the specified network.

We conducted a baseline experiment for Lora to understand the connection time an end device takes to join the LoRa network through OTAA and repeated it 24 times. For NB-IoT, we conducted experiments to measure the time taken for initialisation, attach, connect, detach, disconnect, deinit and modem reset. We performed two experiments - one when LTE modem is reset before initialisation and one without the reset. LTE allows PSM by configuring the period how often the device will connect and how long it will stay actively connected. During the sleep, the LTE modem will go into a low power state, but it will stay attached to the network; thus, no time is spent for attaching after waking up. For *Wi-Fi*, to understand the time taken to connect to Wi-Fi, we calculated the time taken for Wi-Fi init, scan, connect, disconnect, deinit. We experimented 80 times and found that it takes approximately 2.1 seconds to scan the Wi-Fi networks and approximately 5.6 seconds to connect to the specified network. Wi-Fi init, disconnect, and deinit

were almost instantaneously in the range of milliseconds.

Time to reconnect to Wi-Fi, Internet

To understand how much time an IoT device takes to reconnect with the Wi-Fi and the internet. We connected a SCK [425], Pycom FiPy to Home Wi-Fi, a machine via Ethernet to the home router and turn-off-on the Wi-Fi. We created a python script that pings the three hosts: the router, the IoT device (SCK, FiPy), and the cloud machine (google.com) and provided time between the device going offline and coming online. It took approx 1 min 16 sec, 1 min 40 sec, 3 min 5 seconds to get the connectivity back to the router, IoT device, and the internet.

5.6 Evaluation and Discussions

In § 5.4, we have shown that both variants of the proposed resource management algorithm ($CABF$ and $CABF_{inv}$) perform better than the baseline bin-packing algorithms we considered. In this section, we implemented one of the variants, namely $CABF_{inv}$, as part of a multi-network resource allocator running over our *Resilient Edge* platform. We then performed a number of experiments to evaluate the algorithm performance over an edge-node prototype following the experiment setup as shown in the Figures 5.3 and 5.2. The choice of the $CABF_{inv}$ was made in order to try to provide service to all message flows (rather than focus on increasing service for the most critical flows, which would perhaps be the goal in a real deployment) for the sake of demonstrability, i.e. so we can show the sharing of the network interfaces by several flows operating at different levels of criticality.

5.6.1 Criticality-aware allocation of network resources using $CABF_{inv}$

In this section, we show the performance of the proposed $CABF_{inv}$ algorithm when allocating network resources to application flows in a criticality-aware manner. We consider the same application flows and QoS requirements presented in Section § 5.3 and follow the approach described in Section § 5.4 where application flows can request service at different criticality levels from the multi-network resource allocator running at FiPy. The multi-network resource allocator running $CABF_{inv}$ algorithm provides service according to those requirements while considering the network availability conditions, so in this experiment we consider a number of realistic scenarios and evaluate the percentage of served requests and the corresponding criticality levels they were assigned.

In this experiment we consider the four available networks have following maximum bandwidths Wi-Fi (750 $Kbps$), NB-IoT UL (55 $Kbps$), LoRa SF7-125KHz (5.47 $Kbps$) and Sigfox UL (100 bps).

Table 5.9 shows the allocated criticality level, percentage of flows served and average criticality level for the messages flows defined in Table 5.1. Each row of the table shows the metrics obtained

Table 5.9
Obtained criticality level (1 | 2 | 3) and network allocation (* Wi-Fi | # LoRa | + Sigfox | - NB-IoT) for motivating example in FiPy

	Message Flows								**% flows served**	**avg crit level**
	1	2	3	4	5	6	7	8		
Requested Criticality level	1,2,3	1,2,3	1,2	1,2	1,2	1,2	1,2	1		
Network Interfaces	**Allocated Criticality Level**									
Wi-Fi	1	1	1	1	1	1	1	1	100	1
LoRa	1	1	1	2	1	2	1	1	100	1.25
NB-IoT	1	1	1	1	1	1	1	1	100	1
Sigfox	2	2	1	2	1	2	2	1	100	1.625
Wi-Fi + LoRa	1#	1#	1#	1*	1#	1*	1#	1#	100	1
Wi-Fi + Sigfox	1#	1#	1+	1#	1+	1#	1#	1+	100	1
NB-IoT + LoRa	1*	1*	1*	1-	1*	1-	1*	1*	100	1
NB-IoT + Sigfox	1-	1-	1+	1-	1+	1-	1-	1+	100	1

by running the $CABF_{inv}$ algorithm over a different network scenario. Scenarios include situations such as when only a single network is available (only Wi-Fi, LoRa, NB-IoT or Sigfox) or when two different networks are available (such as Wi-Fi or NB-IoT with LoRa and Sigfox). When a high-bandwidth network such as Wi-Fi and NB-IoT is available, we can see that $CABF_{inv}$ is able to assign the lowest criticality level to all flows and to provide all of them with service. We also observe that when only low-bandwidth network interfaces are available (e.g. Sigfox), all flows are still serviced but the average allocated criticality is higher (i.e. flows are only allowed to use the network under more constrained levels of service). Average criticality level is calculated as the sum of all the assigned criticality level divided by the number of message flow allocated.

Such results, which are based on a realistic scenario and network bandwidths, consistently show the same outcomes that were obtained in Section 5.4 for our motivating example and for the synthetic applications: the proposed algorithms are superior to all baselines when one considers together the ability to allocate bandwidth to message flows according to their criticality and to the availability of multiple networks. There are, of course, limitations with regard to the performance of the proposed algorithms, our ability to fully exploit its advantages over the current platform, and the algorithms' ability to handle highly dynamic scenarios. We provide more details and discussion in the following subsections.

5.6.2 Time complexity and Context Switching of $CABF_{inv}$ algorithm

We measured the running time of $CABF_{inv}$ algorithm on the FiPy board and the RPi, repeated it ten times, and average run on FiPy takes $1300ms$ whereas on RPi it takes $7.1ms$.

When a networking event (such as Wi-Fi is disconnected), the allocation algorithm $CABF_{inv}$

has to be executed again. This results in time delay due to de-allocation of old message flows and allocation of new message flows. During this time delay, there's a possibility that the RPi would have written a message on the UART.

As there is a possibility that by the time, Message Flow Element Allocation (MFEA) message was received by RPi, the previous running threads (simulating message flows) would have written few messages to the UART. To resolve this, before doing the re-allocation, we send a message <INFO:RE-ALLOC:INIT> to RPi that, we are going to do the re-allocation, stop sending any message to the UART to minimise the loss of messages. On receiving that message, RPi pauses all the current threads of message flow. Further, FiPy store the old allocations and until it receives an acknowledgement message <INFO:RE-ALLOC:ACCEPTED> from the RPi that it has received the MFEA, it keeps allocating using previous allocation (except the network interface which was lost).

In this case, we log the time, when RE-ALLOC:INIT message was written to the UART by FiPy initiating re-allocation, the time RPi received MFEA message flow allocation message from FiPy, and the time taken by RPi to stop all previous threads (which are simulating the message flows) and generate new threads (as per new allocation). We calculated the time for context switching as the time difference between re-allocation `init` message written by FiPy and the re-allocation accept message received by FiPy. This whole context switching takes 1.3 *s* to 1.5 *s* which includes stopping thread, creating new threads, re-alloc `init` message, re-alloc MFEA time from RPi to FiPy and re-alloc accept from FiPy to RPi.

5.6.3 Discussions

Our work and device limitations: Our work also has certain limitations. Firstly, the CABF algorithm currently does not handle network dynamics such as a change in network bandwidth due to dynamic change of wireless channel and link conditions. The preliminary decision about the network capacity is based on the network availability (whether the network is available or not), and the algorithm calculates the network bandwidth at the start of the network connection. Currently, it is difficult to generate or simulate network problems during application communication to evaluate the consequences on the flows (latency, loss, throughput). For instance, currently, FiPy does not provide the Wi-Fi callback function [426] and does not provide any way to know that Wi-Fi is disconnected. In LTE, we can remove the SIM card or the LTE antenna during a stable connection to simulate network connectivity loss. However, removing SIM or antenna is not officially recommended as they can cause damage to the device. Regarding generating network loss in Lora and Sigfox, both are stateless. FiPy provides a way to check if the device has joined LoRaWAN; however, no way to find whether it is still connected or not. Because of the above reasons, to simulate the loss of Wi-Fi, we have manually set the Wi-Fi bandwidth to zero and then called the re-allocation function. The multi-network resource allocator successfully allocates the message flows to the available network interfaces. From the network bandwidth

perspective (change in bandwidth due to network conditions), a for loop that checks for the LoRa SF, Wi-Fi, and NB-IoT bandwidth at regular intervals can be implemented. However, it requires better support for threading. We will eventually implement the features based on the device support for Wi-Fi callback in the future.

Secondly, there are few device limitations. FiPy does not provide Wi-Fi callback to indicate if the device got disconnected from the Wi-Fi network. Currently, when FiPy is connected to both Wi-Fi and NB-IoT simultaneously, it does not provide a way to define the network interface to be used for sending the packet. Further FiPy team does not advise using both networks simultaneously to simulate a WiFi-LTE bridge, as it will be very slow and expensive [412].

Thirdly, currently, we take a set of message flows and allocate them all together. Because of this, old message flows are de-allocated and re-assigned with either the same or different criticality levels. In future work, we will provide the capability to allow an application to define a new message flow and allocate it from the existing networks without de-allocating and re-allocating the old ones.

Other industrial products: Further, there are different industrial products [427, 428] in the market that provide communication via multiple radio interfaces (such as Wi-Fi, 4G, LoRa, LTE (CAT-M1/NB-IoT)). However, either they provide only LoRa or LTE (CAT-M1/NB-IoT) with Wi-Fi. Currently, we are only aware of FiPy that provides multi-network connectivity for LoRa, LTE (CAT-M1/NB-IoT), Sigfox, Wi-Fi, and Bluetooth. Further, our work enables criticality-aware applications to send messages by allocating resources (network) per the criticality level and network availability. The transmission range of Wi-Fi and other WPAN is different, and it is possible to assign the communication resources to different types of traffic. There can be different factors for consideration in the case of multiple radio devices, e.g., bandwidth, delay, rate adaptation, IP support, and others. Currently, our work considers bandwidth and availability to ensure that applications can send messages as per the defined criticality level.

With the development and popularization of 4G/5G networks, the IoT edge has also shown more possibilities in IoT, VR, and AI intelligence. In this context, NB-IoT, LoRa, and Sigfox provide low-bandwidth network communication methods that are very limited. There might be a case where LPWAN might seem insignificant. On the other hand, our work targets critical edge applications that need to work even when high-bandwidth networks are unavailable.

Resilience via application layer coding and redundancy: Our work acts a switch that moves between different types of protocols (Lora, NB-IoT, Wi-Fi, Sigfox). In practical IoT and network scenarios, the simultaneous use of diverse communication protocols such as Sigfox, Wi-Fi, Bluetooth, NB-IoT, and LoRa is the norm, necessitating sophisticated approaches at the application layer to facilitate seamless interoperability. Smart techniques employed in application layer coding include the abstraction of protocols through middleware solutions, the establishment of unified APIs for standardized interaction, and the implementation of message-oriented middleware (MOM) with message queues and brokers. Additionally, IoT platforms often offer a

higher-level abstraction, simplifying interactions with various devices and protocols. Dynamic protocol switching at runtime, edge computing for distributed intelligence, and the integration of machine learning and AI contribute to adaptive and intelligent decision-making regarding protocol selection. Quality of Service (QoS) management mechanisms ensure prioritization and adaptation based on changing network conditions. Collectively, these techniques enable developers to craft applications capable of interacting seamlessly with a multitude of protocols, fostering flexibility and interoperability in the complex landscape of heterogeneous networks.

Redundancy is crucial for bolstering communication resilience by duplicating paths, devices, or data [429]. However, smarter resilience strategies go beyond a simple switch between protocols. Dynamic protocol selection adapts in real-time to changing network conditions, and hybrid strategies leverage multiple protocols concurrently, offering more flexibility and adaptability. These approaches recognize the complexities of modern networks, providing adaptive and intelligent solutions beyond traditional redundancy.

Taxonomy for resilience: A prospective taxonomy for resilience in heterogeneous networks provides advantages and challenges. On a positive note, the flexibility of these systems shines as a pivotal strength, facilitating the seamless integration of diverse communication protocols like LoRa, Sigfox, Wi-Fi, Bluetooth, and NB-IoT. This adaptability ensures the network's evolution alongside technological advancements and its ability to accommodate an extensive range of devices. Additionally, interoperability emerges as a critical benefit, fostering a collaborative environment where devices with distinct communication styles interact seamlessly, contributing to a unified and cohesive network architecture. Furthermore, fault tolerance mechanisms, adaptive communication strategies, and scalability collectively fortify the network's resilience, ensuring robust performance in dynamic and evolving scenarios.

Nevertheless, these advantages coexist with inherent challenges. The integration of diverse technologies and protocols introduces complexity, posing a notable limitation that necessitates meticulous management to mitigate operational inefficiencies. Security challenges arise due to the diverse components within the network, demanding robust measures to safeguard against potential vulnerabilities and ensure the integrity of communication channels. Issues such as resource utilization, potential integration hurdles, and the imperative for systematic maintenance further accentuate the nuanced landscape of resilience in heterogeneous networks. Achieving a delicate balance between these benefits and challenges becomes imperative for architects and administrators striving to construct adaptive, resilient, and efficient network infrastructures.

Energy Management: Energy considerations are vital in ensuring network resilience, particularly given the prevalence of devices with limited energy and the increasing emphasis on sustainability. Effectively managing energy use is crucial for resilience, especially in scenarios like IoT deployments where devices operate with constrained energy resources. Implementing strategies such as dynamic protocol adjustments and low-power modes [430] extends the lifespan of devices, strengthening the overall resilience of the network. The significance of energy con-

siderations becomes more apparent as the sustainability of devices and the ecological impact of network operations gain prominence in contemporary technology landscapes.

Furthermore, energy itself can be strategically utilized to enhance resilience. Measures such as turning off devices during non-essential periods help conserve energy for critical tasks, improving overall network sustainability. Thoughtful placement of computational algorithms contributes to energy efficiency by distributing tasks to nodes with optimal energy profiles or scheduling energy-intensive operations during surplus availability. Integrating energy considerations into resilience strategies proactively ensures the endurance of energy-constrained devices and aligns with broader sustainability and environmental responsibility goals in network operations.

5.7 Advances in the state of the art

This section presents related work that crosses the intersection of LPWAN, edge resilience, and ILP (Integer Linear Programming) formulations for IoT and edge computing.

Chaudhari et al. [431] provided a comprehensive survey on various LPWAN technologies and presented these technologies concerning application requirements, such as coverage, capacity, cost, low power, and deployment complexity, and provided a comprehensive survey on both standard and non-standard LPWAN technologies. Hossain et al. [432] presented the comparison of different LPWAN technologies in terms of cost structure and scalability and stated that a large rollout with a single LPWAN technology is not cost-efficient.

Similarly, from the use case perspective, Santos et al. [433] evaluated LPWAN technologies for air quality application during “City of Things” project. Further, it also performed anomaly detection for smart city applications using different unsupervised and outlier detection algorithms. Roque et al. [434] created a prototype to detect fire detection in outdoor environments (forests) based on LPWAN networks (Sigfox) and temperature and gas sensor measurements. Rubio-Aparicio et al. [392] implemented an LPWAN residential water management solution supported by hybrid IoT LoRa-Sigfox architecture. All the above solutions provide resiliency by sending data on Lora and Sigfox without guaranteeing applications’ QoS requirements. The work aims to achieve network resiliency by connecting the end devices with inadequate coverage to a Lora-Sigfox Gateway device via LoRa and then forwarding the data to a Sigfox network. These use-uses demonstrate the use of LPWAN for meeting resiliency and low-power communication requirements.

ILP formulations for resource provisioning are widely used for many scheduling problems and are well studied in the literature. For IoT applications, ILP has been used at the gateway level. For example, Santos et al. [435] presents a MILP (Mixed ILP) formulation for resource provisioning in Fog computing, taking into account the Service Function Chaining (SFC) concepts, different LPWAN technologies (LoRa, IEEE 802.11 ah), and multiple optimization objectives. The solution considers end-to-end systems into three segments - sensors/things level, gateways/routers (Fog),

and the cloud and presents smart-city use-cases for garbage collection, air quality monitoring, and closed-circuit television (CCTV) monitoring. Tajiki et al. [436] used ILP to select a set of monitoring flow injected into the network to infer a link delay vector and meet the QoS for delay-sensitive applications in the network. Kim et al. [437] use ILP formulation to create secure migration policies for the communication between things (sensors) and a trusted edge system providing authentication services in the event of Denial-of-Service (DoS) attacks or failures, resulting in resilient authentication and authorization for IoT. In comparison, our work shows that IPL can also be used at the IoT device level to optimize the latency and resiliency of different applications using a Multi-communication network.

From the QoS perspective, multiple research papers have highlighted that the end-to-end perceived QoS on cloud-edge continuum deployment environments depends on many complex system factors [438, 439].

Each abstraction (either vertical or horizontal) adds another level of complexity and delays, affecting QoS. The delays can depend on each edge node's virtualization and containerization techniques [440]. Additionally, many QoS (latency and processing delays) metrics depend on the current load of the local physical/virtual CPU/memory, network acceleration and service invocation techniques [440, 441].

For example, Cicconetti et al. [439] identified four reference execution models (external, in-edge, in-function, in-client) for providing state to enable stateful applications on serverless platforms deployed on the edge nodes. Similarly, Pfandzelter et al. [442] designed a lightweight serverless platform, tinyFaaS, designed explicitly for edge environments and IoT applications.

From the literature, it is evident that edge-enabled FaaS scenarios with serverless support are emerging, and our work is complementary to state-of-the-art work. It can be integrated with middleware or service orchestration architecture by interfacing the Resilient Edge at the communication layer interface or as the network functions virtualization (NFV) in Software-Defined Networking (SDN) architecture.

From the perspective of improving resilience, Qin et al. implemented Multinetwork INformation Architecture (MINA) [443–445] a reflective Observe Analyse Adapt (OAA) middleware approach to manage dynamic and heterogeneous multi-network (such as ZigBee, Bluetooth, PANs, MANETs, 3G/4G, WLAN) in pervasive environments to ensure reliable communication for end applications. The paper presented a formal analysis that can guide network administrators in their decisions to proactively adapt network configurations to achieve mission or application objectives. Compared to this work in our paper, we analyzed seamless switching of the networks on a hardware testbed to meet the resiliency requirements. In our prototype we provided seamless switching while maintaining the critical application requirements without overhead of virtualization and service orchestration middleware. However, our solution could be easily integrated to other intermediate middleware to support application critical requirements while providing seamless network connectivity.

The SCALE2 [446] leveraged MINA and implemented a multi-tier and multi-network approach to drive data flow from IoT devices to cloud platforms. The authors implemented a local Software-defined networking (SDN)-enabled the network, which is adaptive to the network changes to which IoT client devices are connected. This solution's architecture and deployment examples used separate adapters (device) for each communication radio, thus needing another computing device to run the SCALE client software. However, in our work, we use all radios integrated on a single board to allow fast switching between networks on the device level.

Wider aspects of resilience have been discussed in mission-critical applications like autonomous driving, tactile healthcare, and public safety. For example, Modarresi et al. [447] presented a graph-theoretical approach to model IoT systems in smart homes with integrated heterogeneous networks and explored resilience properties. Similarly, Chaterji et al. [448] presents the resilience of Cyber Physical System (CPS) and discusses two techniques resilience-by-design and resilience-by-reaction. Harchol et al. [449] proposed a framework to improve edge-computing resilience for session-oriented applications. They utilized message replay and checkpoint-based mechanisms to make client-edge-server systems more tolerant to edge failures and client mobility. Carvalho et al. [450] implement a replication mechanism LoRa-REP for replicating critical messages on LoraWAN by sending them at different SF and improving redundancy in LoRaWAN for mixed-criticality scenarios.

The literature shows that different forms of replication and redundancy mechanisms are used to achieve resilience in the networks. However, none of those mentioned above work used different LPWAN and Wi-Fi as seamless multi-network infrastructure at the device and the Edge network to meet the guaranteed message delivery.

Our work focuses on achieving network resiliency using the LPWAN network on resource-constrained end devices by providing the capability to the end device to evaluate the application requirements and select the suitable network medium while allowing graceful degradation of services in the event of failures. Further, our work implements an ILP solver in micro-python that can run on a resource-constrained device. Also, multi-network connectivity has benefits in terms of deployment in mission-critical applications (tactile healthcare, public safety in smart cities). For mobility-based IoT like autonomous driving, for example, if one type of network exists in one area. In contrast, there is another network in another geographical location, and the application can perform smooth and seamless network switching.

5.8 Conclusion

The resiliency and reliability requirements of IoT applications vary from non-critical (best delivery efforts) to safety-critical with time-bounded guarantees. In this work, we systematically investigated how to meet these applications mixed-criticality QoS requirements in multi-communication networks.

We presented the network resiliency requirements of IoT applications by defining a theoretical multi-network resource system model and proposed and evaluated a list of resource allocation algorithms and found Criticality-Aware Best Fit ($CABF_{inv}$) algorithm works better to meet high criticality requirements of the example applications. The algorithm provides the best-effort QoS match by taking into consideration the underlying dynamic multi-network environments. We analysed and evaluated the bandwidth, latency, throughput, maximum packet size of LPWAN technologies, such as Sigfox, LoRa, and NB-IoT and implemented and evaluated an adaptive *Resilient Edge* system with Criticality-Aware Best Fit (CABF) resource allocation to meet the application resiliency requirements using underlying LPWAN technologies on RPi and FiPy.

In the current implementation of *Resilient Edge*, we took bandwidth and subsequent inter-message period into consideration for defining criticality. In future, we would like to extend multi-network resource allocator to include message payload size, message transmission frequency, security, privacy and energy consumption parameters in the allocation algorithm. The new allocator would provide applications more flexibility to choose and optimise their resources and QoS for a multi-communication network. In summary, we investigated the limits and metrics required for the best-effort high criticality resilience in multi-communication networks. We presented our findings on how to achieve 100% of the best-effort high criticality level message delivery using multi-communication networks. Our work will help build reliable applications on IoT Edge and provide solutions from the perspective of communication networks to improve service quality and fault tolerance on resource-constrained edge devices. It also opens up new research directions to build reliable and trustworthy IoT applications over robust and resilient IoT Edge.

The author presented the different resiliency requirements for different applications using shared IoT edge networks and understand and evaluate the state-of-the-art LPWAN technologies in terms of their bandwidth, latency, throughput and maximum packet size (RQ5.1 - § 5.3). The author identified and compared resource management approaches that consider QoS requirements at multiple levels of criticality (RQ5.2 - § 5.4) The author defined an adaptive system to meet application resiliency requirements using low power, energy-efficient networks such as LPWAN technologies; also provided an open-source implementation of *Resilient Edge* and detailed insights considering hardware and network limitations (RQ5.3 - § 5.5). Contribution to the knowledge in this chapter (C4) is the improved network resilience for the critically aware applications with different QoS requirements deployed on the edge device that is shared between stakeholders using LPWAN and Wi-Fi.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, in chapter 1, we introduce the research motivation (§ 1.1), contributions (§ 1.2), and thesis outline (§ 1.3).

In chapter 2, we document the different data that can be collected using IoT and digitalisation and the benefits derived from it at different levels such as personal (Table 2.1, Fig. 2.1), building (Table 2.2, Fig. 2.2), district (Table 2.3, Fig. 2.3) and urban (Table 2.4, Table 2.5, Fig. 2.4) and their interconnection (Fig. 2.5). We explore the different initiatives that enable data collection and help cities become more innovative, sustainable and resilient (§ 2.4). The contributions of the chapter 2 (C1) are

- List of IoT data collected at different levels (personal (§ 2.3.1), building (§ 2.3.2), district (§ 2.3.3) and urban (§ 2.3.4)), the benefits derived from them,
- How the data collected at different levels is interconnected (§ 2.3.5), and
- The different initiatives for solving urban data challenges and challenges because of Urbanisation (§ 2.4).

The contribution from chapter 2 enables a reader to quickly identify the IoT data collection landscape at different levels, its benefits and interlinkage and help the curation of data collected at a city level by organisations such as city councils or urban observatories. However, the benefits can only be reaped when data are collected securely and resiliently. Data collection comes with challenges. Citizen concerns about the security and privacy of their data, the transparency of decision-making, and the IoT infrastructure are at the heart of these concerns.

Research organisations often deploy a three-tier architecture (§ 3.2.3) to collect urban data compromising cloud tier (servers), edge (SBCs/gateways) and endpoints (sensors that detect environmental parameters). In chapter 3, we extracted the challenges faced in the smart city research project that collects and processes urban data (§ 3.3). We present these challenges (Fig. 3.3) from our own experiences of participating in smart city research projects (Table 3.1) and reviewing different real IoT testbed from different projects (Table 3.2). We classify these challenges in the context of the life-cycle phases of the V model (§ 3.2.4). The contributions of the chapter 3 (C2) are

- Systematic review of the challenges faced in designing, implementing and deploying IoT infrastructure (§ 3.3) to collect urban data in smart cities research projects.

- Challenges faced in different stages of a smart city research project such as requirement analysis (§ 3.3.1), system design (§ 3.3.2), implementation (§ 3.3.3), integration testing (§ 3.3.4), operational testing (§ 3.3.5), implementation/deployment in the real world (§ 3.3.6) and operational challenges (§ 3.3.7).

The contribution of chapter 3 will help future urban monitoring researchers plan future smart city research projects more efficiently and learn early about the challenges in designing, implementing and deploying infrastructure and hopefully reduce the design and implementation costs of these projects.

Solving all the challenges faced in smart city research projects is difficult. Hence, we focus on the challenges that can be solved using technology and that are commonly faced in all research projects. In chapter 4, we presented a smart city framework (§ 4.3) with an implementable open-source solution (§ 4.4) that solves the software requirements of a smart city platform (§ 4.2.2) and the challenges faced (§ 4.2) in developing, implementing and deploying IoT infrastructure to collect urban data. In addition, we provide details on how the above IoT infrastructure can be shared between stakeholders (§ 4.5) to reduce implementation costs and efficiently use resources. The accurate evaluation of the smart city framework would be to use the framework and architecture in at least two research projects that share the infrastructure to deploy the different blocks of the platform. However, that requires two funded projects with defined requirements, which were unavailable. Instead, we evaluated our work (§ 4.7) by mapping the framework modules to the challenges (Table 4.1) and software requirements (Table 4.2) of a smart city research project. We also provide a few case studies that use and validate the modules in the framework (§ 4.7.2). We also performed a qualitative analysis to understand the approximate time saved using the framework (Table 4.3). Furthermore, we also provide a brief how our work advances the state of the art (§ 4.8). The contributions of chapter 4 (C3) are

- Smart city framework to solve the software requirements of smart cities and the challenges faced during the design, implementation and deployment (§ 4.3).

- Smart city framework implementation using open-source components (§ 4.4).
- Details on how the implemented IoT infrastructure can be shared among research projects and organisations to reduce cost and improve resource efficiency (§ 4.5).

The contribution of chapter 4 will enable future researchers and smart city research projects to deploy IoT infrastructure software stack at the cloud and the edge tier quickly and securely, reducing implementation costs and increasing productivity. It will also enable organisations such as city councils and urban observatories to share the cloud and edge-tier infrastructure with different research projects leading to better resource utilisation.

The above smart city infrastructure contains an edge tier that runs different urban applications. Applications can range from air quality and street light monitoring to criticality-aware applications and can be deployed using containers. IoT applications' network resiliency and reliability requirements vary from non-critical (best delivery efforts) to safety-critical with time-bounded guarantees. In chapter 5, we explored how to achieve resiliency and reliability requirements of applications running on the edge tier that vary from non-critical (best delivery efforts) to safety-critical with time-bounded guarantees using multi-communication networks. We provide two sample applications (§ 5.3): HealthApp and HomeApp, with different messages, flows having different message sizes, minimum intervals between subsequent messages, and criticality level (Table 5.1). We presented the network resiliency requirements of IoT applications by defining a theoretical multi-network resource system model (§ 5.3) and proposed and evaluated a list of resource allocation algorithms (§ 5.4) and found that the Criticality-Aware Best Fit (CABFinv) algorithm works better to meet the high criticality requirements of the example applications. We performed different experiments (§ 5.5.1) to evaluate the suitability of LPWAN for network resiliency, such as payload length, inter-message gap, latency, throughput, and time to connect to the network (Table 5.4). The contributions of chapter 5 (C4) are

- Resilient edge system that improves the network resiliency at the edge tier using LPWAN connection for the critical applications running on the edge tier with defined criticality and QoS requirements (§ 5.3).
- Implemented and evaluated a list of resource allocation algorithms and found Criticality-Aware Best Fit (CABFinv) algorithm works better to meet high criticality requirements of the example applications (§ 5.5).

The contribution of chapter 5 will help build reliable applications on IoT Edge and provide solutions from the perspective of communication networks to improve service quality and fault tolerance on resource-constrained edge devices. It also opens new research directions to build reliable and trustworthy IoT applications over robust and resilient IoT Edge.

To summarise the thesis, we present the benefits derived from the data collection, the challenges faced in the research projects collecting the data, and a smart city framework that

solves a few of the challenges (solvable by technology) and then improves the network resilience of the edge device serving criticality-aware applications with different QoS requirements.

6.2 Research Process Evaluation

Traversing a PhD brings forth a unique challenge to a student. It involves an intricate combination of practical, intellectual, and emotional endeavours and takes its learner through a series of transformations. In the course of this journey, they learn and acquire different attributes such as novelty, competence in research, substantial depth of understanding within a specific discipline, critical use of knowledge, ability to situate research and generate newly acquired knowledge in a broader field, make informed judgements, enhance communication skills, and foremostly, an ability to work autonomously and independently in complex and often unpredictable situations. Additionally, it includes understanding and applying a systematic and critical approach to knowledge and demonstrating skills in designing, executing and reporting research at an advanced level [451].

The author's PhD journey has been the aforementioned and more. Their journey began in September 2018 with an industrially sponsored pre-defined PhD topic on "Service level assurance of Smart Infrastructures", focusing on techniques and approaches for End-to-end SLA management in an IoT/Data hub ecosystem. Due to unavoidable circumstances, the author could not contact their sponsor and receive clarity on their PhD topic. For this reason, in their first year, the author explored the diverse data collected by the IoT sensors and gathered information on improving the service level agreement by predicting the failure of IoT devices. After this year, the author encountered a concern in a meeting with their sponsor, where it was revealed that the prediction failure of devices is a small part of intelligent analytics, and the sponsor did not wish to focus on the IoT sensors since they are easily replaceable due to the reduced cost of electronics and sensors. Furthermore, in Dec-2019 (COVID), most of the author's contact with the sponsor for the next three years was tethered.

Fortunately, earlier in September 2019, the author gained an opportunity to visit the Argonne National Laboratory to understand the Array of Things project and implement the same in Bristol. In the process, the author realised that there are various challenges in implementing IoT infrastructure for research projects that gather data from urban spaces and citizen homes. The author applied a systematic and critical approach to perform an initial literature review to understand the challenges faced in smart cities research projects. Most of the research has been on the experiences of deployment of wireless sensor networks rather than the deployment of full-fledged cloud-edge-endpoints, including community engagement. To further confirm the challenges faced in the smart city research project, the author utilised the Bristol Infrastructure Collaboratory (BIC) network to perform semi-structured interviews with the practitioners of European research projects such as Array of Things, REPLICATE, SPHERE, EuroVal, and others.

The author utilised a systems development process model (V-model) to classify and categorise the challenges learned during the interviews and literature review in different phases of a research project.

BIC works with the city council and other stakeholders and often creates infrastructure for research projects related to air quality, noise monitoring, structural health monitoring and other smart city initiatives. With a substantial understanding of the challenges faced in research and BIC projects, the author designed and developed an approach (smart city framework) to solve the challenges and share the three-tier infrastructure between multiple research projects and organisations, in turn, situating research and generating new knowledge. The author further proposed a hypothesis that using LPWAN technologies and Wi-Fi, network resiliency at the edge IoT device provides the capability to choose a suitable network medium based on the application requirements. The author demonstrated this hypothesis by implementing using an affordable, readily-available, MicroPython-enabled, multi-network microcontroller Pycom FiPy board. The author also demonstrated research dissemination and interpersonal skills by submitting and publishing research papers at conferences and journals.

Overall, the author learned to identify problems and gaps in a specific area, develop a substantial depth of understanding within a specific discipline, apply a systematic and critical approach to solve a problem and work autonomously under challenging situations. These skills allowed them to design and implement practical solutions and report research at an advanced level. In the process, the author has learned different skills that will benefit them in the future when performing research, directing other people's research and advancing state of the art.

6.3 Future Work

One of the main takeaways of this thesis is that we still need much work in the area of smart cities to collect data and develop insights and powerful visualisations to change human behaviour and resolve urban challenges.

One city/county - One infrastructure

The first area directly influenced by this thesis is the sharing of infrastructure between the city council and private and research organisations. Currently, the general practice is that each research project and organisation has its own infrastructure, which may or may not be fully utilised to its potential. This thesis presented a small, readily available example that infrastructure can be shared securely between multiple organisations and must be explored at a larger scale. Ideally, we would like to implement the shared infrastructure approach between real (actual) research projects rather than simulated case studies. In the future, depending on the projects and funding of the Bristol Infrastructure Collaboratory, we may set up a shared infrastructure.

The second area that needs work is the area of smart city hardware security. Our thesis mainly explored the implementation of the IoT infrastructure, assuming that the inline hardware is secure (physical and security features). However, implementing hardware security features, such as secure boot, trusted platform modules, and secure elements, is challenging. A readily available and easily implementable secure solution should be provided for people to implement.

Now, we have a clear idea of what data can be collected and the benefits derived from it. To fast-track and provide more impact, it might be a good idea to have readily available high-impact visualisation that clearly defines the inputs (in terms of data) and the outputs (in terms of visualisation numbers and impacts). Such visualisation and impacts would help all countries and citizens to provide citizen awareness and contribute to reducing climate change impacts and net zero.

Easy manageable endpoints

Our work of smart city framework focuses on managing the cloud and the edge tier using open-source components. Endpoints (sensors) such as SCK, Luftdaten and other products are still widely heterogeneous and need to be more manageable. Although there are standards such as LWM2M that enable proper endpoint device management, it has yet to be widely adopted. During the PhD, we did explore managing endpoint devices using open-source Eclipse Leshan that uses OMA Lightweight M2M protocol from the Open Mobile Alliance for M2M or IoT device management. We deployed the Leshan server on the edge and cloud, providing more privacy and administrative control to the user, enabling the selection of only those parameters that the user wants to share with the organisations. For example, if an SCK kit collects temperature, humidity, and air quality, the user can pause sending temperature data using privacy controls.

The thesis highlighted and attempted to solve the challenges faced by the smart city research project and bring the smart city milestones closer to reality. Therefore, we hope the work and directions will inspire future research on shared infrastructure, resilience, smart cities, and efficient deployment and management of smart city infrastructure. We hope that work will help people and make the world a better place by helping research organisations and city councils to reduce costs and implementation time.

BIBLIOGRAPHY

[1] A. van Bentem, "Airtime calculator for lorawan." https://avbentem.github.io/airtime-calculator/ttn/eu868, 2020.
2020-06-30.

[2] Sigfox Support, "A technicality: padding bits," 2020.
Accessed:2020-06-30.

[3] Sigfox Build, "Radio configurations : Rc technical details." https://build.sigfox.com/sigfox-radio-configurations-rc, 2020.
Accessed:2020-06-30.

[4] K. Forsberg and H. Mooz, "The relationship of system engineering to the project cycle," *INCOSE Int. Symp.*, vol. 1, pp. 57–65, Oct. 1991.

[5] Nigel Cassidy, "Uk to benefit from infrastructure research and innovation," 2021.
[Online; accessed 4-December-2021].

[6] E. Al Nuaimi, H. Al Neyadi, N. Mohamed, and J. Al-Jaroodi, "Applications of big data to smart cities," *Journal of Internet Services and Applications*, vol. 6, no. 1, pp. 1–15, 2015.

[7] P. Barnaghi, M. Bermudez-Edo, and R. Tönjes, "Challenges for quality of data in smart cities," *Journal of Data and Information Quality (JDIQ)*, vol. 6, no. 2-3, pp. 1–4, 2015.

[8] R. Wenge, X. Zhang, C. Dave, L. Chao, and S. Hao, "Smart city architecture: A technology guide for implementation and design challenges," *China Communications*, vol. 11, no. 3, pp. 56–69, 2014.

[9] I. Yaqoob, I. A. T. Hashem, Y. Mehmood, A. Gani, S. Mokhtar, and S. Guizani, "Enabling communication technologies for smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 112–120, 2017.

[10] J. Tapadinhas, "Business analytics from basics to value | business intelligence and analytics | gartner," 2014.
Accessed:2020-06-30.

[11] J. Wood, R. Beecham, and J. Dykes, "Moving beyond sequential design: Reflections on a rich multi-channel approach to data visualization," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2171–2180, 2014.

[12] J. Herring, M. S. VanDyke, R. G. Cummins, and F. Melton, "Communicating local climate risks online through an interactive data visualization," *Environmental Communication*, vol. 11, no. 1, pp. 90–105, 2017.

[13] F. T. Neves, M. de Castro Neto, and M. Aparicio, "The impacts of open data initiatives on smart cities: A framework for evaluation and monitoring," *Cities*, vol. 106, p. 102860, 2020.

[14] B. C. Council, "Open data bristol," 2018.
[Online; accessed 17-December-2018].

[15] L. C. Council, "London datastore," 2018.
[Online; accessed 17-December-2018].

[16] P. Filion, M. Moos, and G. Sands, "Urban neoliberalism, smart city, and big tech: The aborted sidewalk labs toronto experiment," *Journal of Urban Affairs*, pp. 1–19, 2023.

[17] S. Gunner, *A Systematic Methodology for Technology Interventions in the Urban Environment: Three Case Studies*.
PhD thesis, University of Bristol, 2022.

[18] A. Elsts, X. Fafoutis, P. R. Woznowski, E. L. Tonkin, G. Oikonomou, R. Piechocki, and I. Craddock, "Enabling Healthcare in Smart Homes: The SPHERE IoT Network Infrastructure," *Communications Magazine*, vol. 56, pp. 164–170, Dec. 2018.

[19] 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.

[20] T. Farnham, S. Jones, A. Aijaz, Y. Jin, I. Mavromatis, U. Raza, A. Portelli, A. Stanoev, and M. Sooriyabandara, "UMBRELLA collaborative robotics testbed and IoT platform," *2021 IEEE 18th Annual Consumer Communications and Networking Conference, CCNC 2021*, no. RoboCom, 2021.

[21] J. Wilmoth, "Press briefing upon release of data from world urbanization prospects: The 2018 revision," *UN Headquarters, New York: UN*, 2019.

[22] British Standards Institute (BSI), "PD 8100:2015 - Smart cities overview – Guide," 2015.

[23] V. Gutiérrez, E. Theodoridis, G. Mylonas, F. Shi, U. Adeel, L. Diez, D. Amaxilatis, J. Choque, G. Camprodom, J. McCann, *et al.*, "Co-creating the cities of the future," *Sensors*, vol. 16, no. 11, p. 1971, 2016.

[24] J. Davies and S. Stincic-Clarke, "Creating & supporting iot innovation ecosystems: Lessons from cityverve," in *Living in the Internet of Things (IoT 2019)*, pp. 1–3, IET, 2019.

[25] D. Gooch, A. Wolff, G. Kortuem, and R. Brown, "Reimagining the role of citizens in smart city projects," in *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, UbiComp/ISWC'15 Adjunct, (New York, NY, USA), p. 1587–1594, Association for Computing Machinery, 2015.

[26] I. Calzada, "Replicating smart cities: The city-to-city learning programme in the replicate ec-h2020-scc project," *Smart Cities*, vol. 3, no. 3, pp. 978–1003, 2020.

[27] D. Pavithra and R. Balakrishnan, "Iot based monitoring and control system for home automation," in *2015 global conference on communication technologies (GCCT)*, pp. 169–173, IEEE, 2015.

[28] C. Wang, Z. Bi, and L. Da Xu, "Iot and cloud computing in automation of assembly modeling systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1426–1434, 2014.

[29] F. Shrouf, J. Ordieres, and G. Miragliotta, "Smart factories in industry 4.0: A review of the concept and of energy management approached in production based on the internet of things paradigm," in *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 697–701, 2014.

[30] V. Miori and D. Russo, "Improving life quality for the elderly through the social internet of things (siot)," in *2017 Global Internet of Things Summit (GIoTS)*, pp. 1–6, 2017.

[31] Belfast City Council, "The Belfast Agenda," 2014.

[32] Bristol City Council, "One City Plan 2020," 2020.

[33] Chicago Data Portal, "Chicago data portal," 2019.

[34] S. Thornton, "A guide to chicago's array of things initiative," Jan. 2018.

[35] Portal, Chicago Data, "Urban collaboratory at the university of michigan," 2022.

[36] Apple, "Empowering people to live a healthier day," 2020.

[37] G. D. Fulk, S. A. Combs, K. A. Danks, C. D. Nirider, B. Raja, and D. S. Reisman, "Accuracy of 2 Activity Monitors in Detecting Steps in People With Stroke and Traumatic Brain Injury," *Physical Therapy*, vol. 94, pp. 222–229, Feb. 2014.

[38] X. Fafoutis, A. Vafeas, B. Janko, S. Sherratt, J. Pope, A. Elsts, E. Mellios, G. Hilton, G. Oikonomou, R. Piechocki, and I. Craddock, "Designing Wearable Sensing Platforms for Healthcare in a Residential Environment," *EAI Endorsed Trans. on Pervasive Health and Technology*, vol. 17, Sept. 2017.

[39] A. Vafeas, X. Fafoutis, A. Elsts, I. Craddock, M. I. Biswas, R. Piechocki, and G. Oikonomou, "Wearable Devices for Digital Health: The SPHERE Wearable 3," in *Proc. ACM EWSN*, 2020.

[40] X. J. Wang and M. Camilleri, "A smart toilet for personalized health monitoring," *Nature Reviews Gastroenterology & Hepatology*, vol. 17, no. 8, pp. 453–454, 2020.

[41] J. M. Kortelainen, M. van Gils, and J. Pärkkä, "Multichannel bed pressure sensor for sleep monitoring," in *2012 Computing in Cardiology*, pp. 313–316, Sept. 2012.

[42] 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.

[43] P. Woznowski, X. Fafoutis, T. Song, S. Hannuna, M. Camplani, L. Tao, A. Paiement, E. Mellios, M. Haghighi, N. Zhu, G. Hilton, D. Damen, T. Burghardt, M. Mirmehdi, R. Piechocki, D. Kaleshi, and I. Craddock, "A multi-modal sensor infrastructure for healthcare in a residential environment," *2015 IEEE International Conference on Communication Workshop, ICCW 2015*, pp. 271–277, 2015.

[44] A. Vafeas, A. Elsts, J. Pope, X. Fafoutis, G. Oikonomou, R. Piechocki, and I. Craddock, "Energy-Efficient, Noninvasive Water Flow Sensor," in *Proc. SMARTCOMP*, pp. 139–146, 2018.

[45] A. Ghandeharioun, A. Azaria, S. Taylor, and R. W. Picard, ""kind and grateful": A context-sensitive smartphone app utilizing inspirational content to promote gratitude," *Psychology of well-being*, vol. 6, no. 1, pp. 1–21, 2016.

[46] Action for Happiness, "Action for Happiness," 2022.

[47] S. Brasche and W. Bischof, "Daily time spent indoors in german homes – baseline data for the assessment of indoor exposure of german occupants," *International Journal of Hygiene and Environmental Health*, vol. 208, no. 4, pp. 247–253, 2005.

[48] B. Von Neida, D. Manicria, and A. Tweed, "An analysis of the energy and cost savings potential of occupancy sensors for commercial lighting systems," *Journal of the Illuminating Engineering Society*, vol. 30, no. 2, pp. 111–125, 2001.

[49] C. F. Ratti, "Localized thermal management system," Mar. 2016.
US Patent App. 14/856,679.

[50] M. R. Porto, "3e-Houses FINAL REPORT ," 2013.

[51] "Twinergy," 2022.

[52] Knowle West Media Centre, "Community led housing Bristol - We can make homes," 2022.

[53] N. Katuk, T. Jayasangar, and Y. Yusof, "Design and development of smart list: a mobile app for creating and managing grocery lists," *Baghdad Science Journal*, vol. 16, no. 2, pp. 462–476, 2019.

[54] R. Roubein, "Apps help reduce, reuse, recycle," *USA Today*, pp. 02B–02B, 2011.

[55] A. Tadigadapa, "Smart containers for food storage in refrigerators," *IEEE Potentials*, vol. 41, no. 3, pp. 29–34, 2022.

[56] BEEP, "Expiration Date Tracking Solution," 2022.

[57] I. P. Mohottige, T. Sutjarittham, N. Raju, H. H. Gharakheili, and V. Sivaraman, "Role of campus wifi infrastructure for occupancy monitoring in a large university," in *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, pp. 1–5, IEEE, 2018.

[58] K. Akkaya, I. Guvenc, R. Aygun, N. Pala, and A. Kadri, "Iot-based occupancy monitoring techniques for energy-efficient smart buildings," in *2015 IEEE Wireless communications and networking conference workshops (WCNCW)*, pp. 58–63, IEEE, 2015.

[59] T. Banerjee, M. Skubic, J. M. Keller, and C. Abbott, "Sit-to-stand measurement for in-home monitoring using voxel analysis," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, pp. 1502–1509, July 2014.

[60] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the microsoft kinect," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, pp. 290–301, Jan. 2015.

[61] D. Meadows-Klue, "Inside the Smart Home," *J Direct Data Digit Mark Pract*, vol. 5, no. 3, pp. 307–308, 2004.

[62] L. Wouters, E. Marin, T. Ashur, B. Gierlichs, and B. Preneel, "Fast, furious and insecure: Passive keyless entry and start systems in modern supercars," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 3, pp. 66–85, 2019.

[63] G. Camprodon, Ó. González, V. Barberán, M. Pérez, V. Smári, M. Á. de Heras, and A. Bizzotto, "Smart citizen kit and station: An open environmental monitoring system for citizen participation and scientific experimentation," *HardwareX*, vol. 6, p. e00070, 2019.

[64] A. Hamm, "Particles matter: A case study on how civic iot can contribute to sustainable communities," in *Proceedings of the 7th International Conference on ICT for Sustainability*, pp. 305–313, 2020.

[65] ATMO, "ATMO - Air Quality Monitors for Consumers and Businesses," 2022.

[66] World Health Organization, *WHO guidelines for indoor air quality: selected pollutants*. World Health Organization. Regional Office for Europe, 2010.

[67] U. S. E. P. Agency, "Introduction to indoor air quality - primary causes of indoor air problems," 2018.
[Online; accessed 17-December-2018].

[68] N. Zhu, T. Diethe, M. Camplani, L. Tao, A. Burrows, N. Twomey, D. Kaleshi, M. Mirmehdi, P. Flach, and I. Craddock, "Bridging e-health and the internet of things: The sphere project," *IEEE Intelligent Systems*, vol. 30, no. 4, pp. 39–46, 2015.

[69] R. Valancius, A. Mutiari, A. Singh, C. Alexander, D. A. De La Cruz, and F. E. del Pozo Jr, "Solar photovoltaic systems in the built environment: Today trends and future challenges," *Journal of Sustainable Architecture and Civil Engineering*, vol. 23, no. 2, pp. 25–38, 2018.

[70] J. Bourgeois, S. Foell, G. Kortuem, B. A. Price, J. van der Linden, E. Y. Elbanhawy, and C. Rimmer, "Harvesting green miles from my roof: An Investigation into Self-Sufficient Mobility with Electric Vehicles," *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15*, pp. 1065–1076, 2015.

[71] N. Chowdhury, B. Price, A. Smith, G. Kortuem, J. van der Linden, and J. Moore, "EV Charging: Separation of Green and Brown Energy Using IoT," *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pp. 674–677, 2016.

[72] J. Bourgeois, J. Van Der Linden, G. Kortuem, B. A. Price, and C. Rimmer, "Using Participatory Data Analysis to Understand Social Constraints and Opportunities of Electricity Demand-Shifting," *Proceedings of the 2014 conference ICT for Sustainability*, 2014.

[73] R. Cherrington, V. Goodship, A. Longfield, and K. Kirwan, "The feed-in tariff in the uk: A case study focus on domestic photovoltaic systems," *Renewable Energy*, vol. 50, pp. 421–426, 2013.

[74] Ofwat, "Out in the cold - Water companies' response to the 'Beast from the East'," 2018.

[75] V. Ranjan, M. V. Reddy, M. Irshad, and N. Joshi, "The internet of things (iot) based smart rain water harvesting system," in *2020 6th International Conference on Signal Processing and Communication (ICSC)*, pp. 302–305, 2020.

[76] H. Rezaei, P. Melville-Shreeve, and D. Butler, "Smart rainwater management systems powered by the internet of things: a uk case study,"

[77] P. Melville-Shreeve, "Building resilient networks for severe weather," 2018. [Online; accessed 17-December-2018].

[78] T. Litman, "Measuring transportation, traffic, mobility and accessibility," *ITE Journal (Institute of Transportation Engineers)*, 2011.

[79] New York City Parks, "New York City Street Tree Map," 2022.

[80] Council, Bristol City, "Bristol city council | open data bristol," 2015.

[81] Mayor of London, "London data store," 2018.

[82] R. Pijnenburg, "Enhancing supplier retention on the peerby go platform," 2017.

[83] "KONNECKTid," 2020.

[84] "ShareTheMeal," 2020.

[85] "Olio - The #1 Free Sharing App," 2022.

[86] "Too Good To Go - Food Waste Movement," 2022.

[87] "Helpfulpeeps - Social Marketplace for local help," 2022.

[88] "MK Food Revolution launching local Food Passports funded by MK:Smart," 2022.

[89] "Hood Champions by Singapore Kindness Movement," 2022.

[90] L. R. West, "Strava: challenge yourself to greater heights in physical activity/cycling and running," *British journal of sports medicine*, vol. 49, no. 15, pp. 1024–1024, 2015.

[91] D. in2 Social, "Walkers group santander: Reducing loneliness through outdoor activities that strengthen social and mental resilience of citizens," 2017.
[Online; accessed 17-December-2018].

[92] "Walks in the City: Improving the wellbeing of senior citizens and their neighbourhoods," 2022.

[93] "Breastfeeding Hub," 2022.

[94] "Air Quality Sensing - Umbrella," 2022.

[95] I. Weeks, B. Holden, and A. Stanoev, "A low power system for synchronising buffered air quality data," in *2022 IEEE International Symposium on Measurements & Networking (M&N)*, pp. 1–6, IEEE, 2022.

[96] M. I. G. Daepp, A. Cabral, V. Ranganathan, V. Iyer, S. Counts, P. Johns, A. Roseway, C. Catlett, G. Jancke, D. Gehring, C. Needham, C. von Veh, T. Tran, L. Story, G. D'Amone, and B. H. Nguyen, "Eclipse: An end-to-end platform for low-cost, hyperlocal environmental sensing in cities," in *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 28–40, 2022.

[97] J. Lanza, L. Sánchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, and V. Gutiérrez, "Large-scale mobile sensing enabled internet-of-things testbed for smart city services," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, p. 785061, 2015.

[98] Traffic Pollution Flow, "Traffic controlled by air quality," 2018.
[Online; accessed 17-December-2018].

[99] O. Sensifai, "Brussels qualitative and quantitative noise map," 2018.
[Online; accessed 17-December-2018].

[100] The City of Amsterdam, "Noise map - 2018 amsterdam," 2018.
[Online; accessed 17-December-2018].

[101] University of Edinburgh, "Edinburg citysounds - exploring the sounds of the city," 2018.
[Online; accessed 17-December-2018].

[102] S. Corwin and T. L. Johnson, "The role of local governments in the development of china's solar photovoltaic industry," *Energy Policy*, vol. 130, pp. 283–293, 2019.

[103] N. Bernal, "University towns consider rolling out thermal cameras on waterfronts to stop revellers drowning," 2018.
[Online; accessed 17-May-2019].

[104] "Street Light Monitoring - Umbrella," 2022.

[105] K. Mclaughlin, "Lighting the way: Norway's auto-dimming street lamps get brighter only as traffic approaches and then return to 20 percent power - and will help reduce country's carbon footprint," 2018.
[Online; accessed 17-December-2018].

[106] A. Medvedev, P. Fedchenkov, A. Zaslavsky, T. Anagnostopoulos, and S. Khoruzhnikov, "Waste management as an iot-enabled service in smart cities," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pp. 104–115, Springer, 2015.

[107] Team USC, "Conversational interfaces for urban data," 2018.
[Online; accessed 17-December-2018].

[108] Bristol.Gov.UK, "State of the art operations centre opens in bristol," 2018.
[Online; accessed 17-May-2019].

[109] Transport for London, "Review of the TfL WiFi pilot," 2017.

[110] F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, and C. Ratti, "Real-time urban monitoring using cell phones: A case study in rome," *IEEE transactions on intelligent transportation systems*, vol. 12, no. 1, pp. 141–151, 2010.

[111] L. Adler, "How smart city barcelona brought the internet of things to life," 2018.
[Online; accessed 17-December-2018].

[112] N. K. Giang, V. C. M. Leung, M. Kawano, T. Yonezawa, J. Nakazawa, R. Lea, and M. Broadbent, "Cityflow: Exploiting edge computing for large scale smart city applications," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 1–4, 2019.

[113] Bristol.Gov.UK, "How a bike sharing scheme benefits everyone in manchester," 2018.
[Online; accessed 17-May-2019].

[114] I. D. E. E. Paris, "Madame Mayor, I have an idea - Your ideas for Paris | The digital platform for the Paris of tomorrow," 2020.

[115] I. Bojic, G. Marra, and V. Naydenova, "Online tools for public engagement: case studies from reykjavik," *arXiv preprint arXiv:1611.08981*, 2016.

[116] B. Hecht, "Carticipe/ debatomap," 2018.
[Online; accessed 17-December-2018].

[117] Ten Four, "Ushahidi," 2018.
[Online; accessed 17-December-2018].

[118] OpenStreetMap, "Openstreetmap," 2018.
[Online; accessed 17-December-2018].

[119] OpenPlans, "Shareabouts," 2018.
[Online; accessed 17-December-2018].

[120] "Cyclestreets uk-wide cycle journey planner and photomap," 2022.
[Online; accessed 4-December-2021].

[121] S. Gunner, E. Wilson, and T. Tryfonas, "Using Telematics to Gather User Behaviour Data from a Fleet of Electric Bicycles," 2021.

[122] M. B. Barcena and C. Wueest, "Insecurity in the internet of things," *Security response, symantec*, p. 20, 2015.

[123] H. Lin and N. Bergmann, "Iot privacy and security challenges for smart home environments," *Information*, vol. 7, p. 44, July 2016.

[124] X. Fafoutis, L. Marchegiani, G. Z. Papadopoulos, R. Piechocki, T. Tryfonas, and G. Oikonomou, "Privacy leakage of physical activity levels in wireless embedded wearable systems," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 136–140, 2016.

[125] P. Fussey and D. Murray, "Independent report on the london metropolitan police service's trial of live facial recognition technology," 2019.

[126] F. Liang, V. Das, N. Kostyuk, and M. M. Hussain, "Constructing a data-driven society: China's social credit system as a state surveillance infrastructure," *Policy & Internet*, vol. 10, no. 4, pp. 415–453, 2018.

[127] R. Creemers, "China's Social Credit System: An Evolving Practice of Control," *SSRN Electronic Journal*, vol. 222, no. 2015, pp. 59–71, 2018.

[128] M. Chorzempa, P. Triolo, and S. Sacks, "China's Social Credit System: A Mark of Progress or a Threat to Privacy?," Policy Briefs PB18-14, Peterson Institute for International Economics, June 2018.

[129] K. N. Mishra and C. Chakraborty, "A novel approach toward enhancing the quality of life in smart cities using clouds and iot-based technologies," in *Digital Twin Technologies and Smart Cities*, pp. 19–35, Springer, 2020.

[130] S. Chatterjee, "Influence of iot policy on quality of life: From government and citizens' perspectives," *International Journal of Electronic Government Research (IJEGR)*, vol. 15, no. 2, pp. 19–38, 2019.

[131] A. Sharida, A. Hamdan, and M. AL-Hashimi, "Smart cities: The next urban evolution in delivering a better quality of life," in *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications*, pp. 287–298, Springer, 2020.

[132] A. Rostami, M. Vigren, S. Raza, and B. Brown, "Being hacked: Understanding victims' experiences of {IoT} hacking," in *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, pp. 613–631, 2022.

[133] R. S. Yamaguchi, K. Hirota, K. Hamada, K. Takahashi, K. Matsuzaki, J. Sakuma, and Y. Shirai, "Applicability of existing anonymization methods to large location history data in urban travel," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 997–1004, IEEE, 2012.

[134] D. Münch, A.-K. Grosselfinger, E. Krempel, M. Hebel, and M. Arens, "Data anonymization for data protection on publicly recorded data," in *International Conference on Computer Vision Systems*, pp. 245–258, Springer, 2019.

[135] C. A. Flanagan, T. Kim, A. Pykett, A. Finlay, E. E. Gallay, and M. Pancer, "Adolescents' theories about economic inequality: Why are some people poor while others are rich?," *Developmental Psychology*, vol. 50, no. 11, p. 2512, 2014.

[136] O. P. Hauser, G. T. Kraft-Todd, D. G. Rand, M. A. Nowak, and M. I. Norton, "Invisible inequality leads to punishing the poor and rewarding the rich," *Behavioural Public Policy*, vol. 5, no. 3, pp. 333–353, 2021.

[137] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[138] D. T. L. Shek, "Protests in hong kong (2019–2020): a perspective based on quality of life and well-being," *Applied Research in Quality of Life*, pp. 1–17, 2020.

[139] E. Livni, "Hong kong protesters are attacking smart lampposts," 2019.
[Online; accessed 4-December-2019.

[140] G. C. A. Peng, M. B. Nunes, and L. Zheng, "Impacts of low citizen awareness and usage in smart city services: the case of london's smart parking system," *Information Systems and e-Business Management*, vol. 15, no. 4, pp. 845–876, 2017.

[141] P. Kamnuansilpa, S. Laochankham, C. D. Crumpton, and J. Draper, "Citizen awareness of the smart city: A study of khon kaen, thailand," *The Journal of Asian Finance, Economics and Business*, vol. 7, no. 7, pp. 497–508, 2020.

[142] P. Cooper, T. Crick, T. Tryfonas, and G. Oikonomou, "Whole-life environmental impacts of ict use," in *2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–7, IEEE, 2015.

[143] G. van Capelleveen, J. Pohl, A. Fritsch, and D. Schien, "The footprint of things: A hybrid approach towards the collection, storage and distribution of life cycle inventory data.," in *ICT4S*, pp. 350–364, 2018.

[144] B. Baccarne, P. Mechant, and D. Schuurman, "Empowered cities? an analysis of the structure and generated value of the smart city ghent," in *Smart city*, pp. 157–182, Springer, 2014.

[145] Y. Zhao, H. Haddadi, S. Skillman, S. Enshaeifar, and P. Barnaghi, "Privacy-preserving activity and health monitoring on databox," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, pp. 49–54, 2020.

[146] R. M. Keenan and L.-N. Tran, "Fall detection using wi-fi signals and threshold-based activity segmentation," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–6, IEEE, 2020.

[147] Bristol City Council, "Bristol is open," 2022.
[Online; accessed 4-Nov-2022].

[148] Bristol City Council, "One City Plan Climate Strategy," 2020.

[149] J. Armitt, *The Armitt review: An independent review of long term infrastructure planning commissioned for Labour's Policy Review*.
Labour Party, 2013.

[150] UKCRIC, "Ukcric missions," 2022.
[Online; accessed 4-Nov-2022].

[151] UO, "Urban observatory," 2022.
[Online; accessed 4-Nov-2022].

[152] S. Gunner, P. Vardanega, T. Tryfonas, J. Macdonald, and E. Wilson, "Rapid deployment of a wsn on the clifton suspension bridge, uk," *Proceedings of the ICE - Smart Infrastructure and Construction*, vol. 170, pp. 59–71, Dec. 2017.

[153] E. Coraggio, D. Han, W. Liu, and T. Tryfonas, "Hydroinformatics of smart cities: Real-time water quality monitoring and prediction," in *International Association for Hydro-Environment Engineering and Research World Congress 2019*, Sept. 2019.
International Association for Hydro-Environment Engineering and Research World Congress 2019 : Water - Connecting the World, IAHR 2019 ; Conference date: 01-09-2019 Through 06-09-2019.

[154] T. Strain, S. Gunner, and E. Wilson, "Estimation of vehicle counts from the structural response of a bridge," in *2019 International Conference on Smart Infrastructure and*

Construction (ICSIC 2019), (United Kingdom), Thomas Telford (ICE Publishing), July 2019.

[155] P. James, J. Jonczyk, L. Smith, N. Harris, T. Komar, D. Bell, and R. Ranjan, "Realizing smart city infrastructure at scale, in the wild: A case study," *Frontiers in Sustainable Cities*, 2022.

[156] P. James, R. Das, A. Jalosinska, and L. Smith, "Smart cities and a data-driven response to covid-19," *Dialogues in Human Geography*, vol. 10, no. 2, pp. 255–259, 2020.

[157] U. Observatory, "Effects of virus measures," 2022.
[Online; accessed 4-Nov-2022].

[158] D. Burtan, K. Joyce, J. F. Burn, T. C. Handy, S. Ho, and U. Leonards, "The nature effect in motion: visual exposure to environmental scenes impacts cognitive load and human gait kinematics," *Royal Society Open Science*, vol. 8, no. 1, p. 201100.

[159] Bristol Infrastructure Collaboratory, "Travel exhibition," 2022.
[Online; accessed 4-Nov-2022].

[160] GridKey, "Metrology and communications units (mcu) 20," 2022.
[Online; accessed 4-Nov-2022].

[161] T. Tryfonas, S. Gunner, U. Baloglu, P. Tully, S. Karatzas, and C. Tryfona, "Causal loop mapping of emerging energy systems in project twinergy: towards consumer engagement with group model building," in *Proceedings of the 15th International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 254–259, 2022.

[162] "Clifton Suspension Bridge Trust," 2022.

[163] M. Balestrini, Y. Rogers, C. Hassan, J. Creus, M. King, and P. Marshall, "A city in common: A framework to orchestrate large-scale citizen engagement around urban issues," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, (New York, NY, USA), p. 2282–2294, Association for Computing Machinery, 2017.

[164] 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)*, (United Kingdom), pp. 605–611, Thomas Telford (ICE Publishing), July 2019.

[165] "Clifton Suspension Bridge Harp," 2022.

[166] S. Ho, A. Mohtadi, K. Daud, U. Leonards, and T. Handy, "Using smartphone accelerometry to assess the relationship between cognitive load and gait dynamics during outdoor walking," *Scientific Reports*, vol. 9, Feb. 2019.

[167] D. Pancoast, "ofThings: Experiments in Object Mobility," 2021.

[168] Y. Chen and D. Han, "Water quality monitoring in smart city: A pilot project," *Automation in Construction*, vol. 89, pp. 307–316, 2018.

[169] Bristol Infrastructure Collaboratory, "Bristol air quality," 2021.
[Online; accessed 4-December-2021].

[170] S. S. Ii, L. Fitzgerald, M. M. Morys-Carter, N. L. Davie, and R. Barker, "Knowledge translation in tri-sectoral collaborations: an exploration of perceptions of academia, industry and healthcare collaborations in innovation adoption," *Health Policy*, vol. 122, no. 2, pp. 175–183, 2018.

[171] B. Ahlgren, M. Hidell, and E. C.-H. Ngai, "Internet of things for smart cities: Interoperability and open data," *IEEE Internet Computing*, vol. 20, no. 6, pp. 52–56, 2016.

[172] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[173] S. B. Merriam, J. Johnson-Bailey, M.-Y. Lee, Y. Kee, G. Ntseane, and M. Muhamad, "Power and positionality: Negotiating insider/outsider status within and across cultures," *International journal of lifelong education*, vol. 20, no. 5, pp. 405–416, 2001.

[174] 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*, pp. 315–333, Springer, 2017.

[175] 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.

[176] "Replicate: Renaissance of places with innovative citizenship and technology." `https://replicate-project.eu/`.
Accessed: 2021-06-30.

[177] "Twinergy - digital twin." `https://www.twinergy.eu/`.
Accessed: 2021-06-30.

[178] 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.

[179] S. Gunner, P. J. Vardanega, T. Tryfonas, J. H. G. 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.

[180] D. Nepomuceno, T. Tryfonas, and P. J. 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*, pp. 605–611, ICE Publishing, 2019.

[181] "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.

[182] 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.

[183] 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.

[184] 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.

[185] 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.

[186] 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.

[187] 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.

[188] 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)*, pp. 145–150, 2015.

[189] 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.

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

[191] 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.

[192] 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*, pp. 1–6, 2014.

[193] 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.

[194] 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. 14309–14322, 2017.

[195] 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.

[196] 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), pp. 225–237, Springer Berlin Heidelberg, 2012.

[197] 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.

[198] 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.

[199] 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.

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

[201] 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.

[202] L. Nussbaum, "Testbeds support for reproducible research," in *Proceedings of the Reproducibility Workshop*, Reproducibility '17, (New York, NY, USA), pp. 24–26, Association for Computing Machinery, 2017.

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

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

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

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

[207] "Fed4fire+." https://www.fed4fire.eu/.
Accessed: 2021-06-30.

[208] "Fit cloudlab." https://www.cloudlab.us/.
Accessed: 2021-06-30.

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

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

[211] "Pragma - pacific rim applications and grid middleware assembly." http://www.pragma-grid.net/resources/testbed/.
Accessed: 2021-06-30.

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

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

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

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

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

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

[218] "3e-houses." https://ses.jrc.ec.europa.eu/3e-houses.
Accessed: 2021-06-30.

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

[220] 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, pp. 609–613, IEEE, 2010.

[221] I. Bevers, "Intelligence at the edge part 1: The edge node | analog devices," 2019.

[222] 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.

[223] L. Sanchez, L. Munoz, 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.

[224] I. Beavers, "Intelligence at the Edge Part 2 : Reduced Time to Insight," 2019.

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

[226] P. Rook, "Controlling software projects," *Software Engineering Journal*, vol. 1, pp. 7–16, Jan. 1986.

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

[228] 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.

[229] 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.

[230] E. 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.

[231] "Open data bristol." `https://opendata.bristol.gov.uk/pages/homepage/`.
Accessed: 2021-06-30.

[232] "London datastore - greater london." `https://data.london.gov.uk/`.
Accessed: 2021-06-30.

[233] 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.

[234] 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.

[235] 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.

[236] 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.

[237] "A guide to threat modelling for developers." `https://martinfowler.com/articles/agile-threat-modelling.html`.
Accessed: 2020-06-30.

[238] "Threat modeling." `https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling`.
Accessed: 2020-06-30.

[239] 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.

[240] M. Sweney, "Bristol is worst uk city for broadband outages with 169 hours a year," 2020. Accessed:2020-06-30.

[241] 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.

[242] 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.

[243] U. I. S. Communications, "Reboot your computer for best performance," 2019.

[244] "Thermal testing raspberry pi 4." `https://www.raspberrypi.org/blog/thermal-testing-raspberry-pi-4/`.
Accessed: 2021-06-30.

[245] 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.

[246] 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*, pp. 13–25, Springer, 2015.

[247] "Tutorial: Rpl border router." `https://github.com/contiki-ng/contiki-ng/wiki/Tutorial:-RPL-border-router`.
Accessed: 2021-06-30.

[248] 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.

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

[250] S. Y. Kumar R and N. Champa H, "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.

[251] 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*, pp. 1359–1362, 2004.

[252] 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.

[253] 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*, pp. 445–450, IEEE, 2013.

[254] 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)*, pp. 117–122, 2014.

[255] 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*, pp. 324–327, IEEE, 2014.

[256] 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*, pp. 428–432, 2013.

[257] 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.

[258] 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.

[259] Home Assistant, "Docker monitor broken on debian 11," 2022. [Online; accessed 30-May-2022].

[260] 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.

[261] "Smart citizen docs." `https://docs.smartcitizen.me/Smart%20Citizen%20Kit/`. Accessed: 2021-06-30.

[262] "Luftdaten sensor.community." `https://sensor.community/en/`. Accessed: 2021-06-30.

[263] "Tidc-cc2650stk-sensortag." `https://www.ti.com/tool/TIDC-CC2650STK-SENSORTAG`. Accessed: 2021-06-30.

[264] "Tp-link smart plugs." `https://www.tp-link.com/uk/home-networking/smart-plug/`. Accessed: 2021-06-30.

[265] Open Mobile Alliance, "Lightweight machine to machine requirements," *Candidate Version*, vol. 1, 2013.

[266] 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.

[267] 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.

[268] A. Vafeas, *End to End System Optimisation of Wireless Interconnected Sensors for Healthcare Monitoring at Homes: Advancements in multimodal sensing technology for healthcare in residential environments*.

PhD thesis, University of Bristol, 2021.

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

[270] 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.

[271] 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.

[272] 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.

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

[274] "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.

[275] A. Del Sole, *Building Applications with Python*, pp. 211–233. Berkeley, CA: Apress, 2021.

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

[277] K. Keahey, J. Anderson, P. Ruth, J. Colleran, C. Hammock, J. Stubbs, and Z. Zhen, "Operational lessons from chameleon," *ACM International Conference Proceeding Series*, 2019.

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

[279] "Tp-link smart plug." `https://www.tp-link.com/uk/home-networking/smart-plug/`. Accessed: 2020-06-30.

[280] "Tesla powerwall." `https://www.tesla.com/en_gb/powerwall`. Accessed: 2020-06-30.

[281] "Open energy monitoring." `https://openenergymonitor.org/`. Accessed: 2020-06-30.

[282] 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.

[283] 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.

[284] 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.

[285] 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 Telecommunications Engineering, LNICST*, vol. 170, pp. 415–426, 2016.

[286] 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)*, pp. 1–3, IEEE, 2012.

[287] 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.

[288] 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.

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

[290] Knowle West Media Centre, "3-E HOUSES Best practice guide," tech. rep., European Commission, Joint Research Centre, Smart Electricity Systems and Interoperability, 2013.

[291] "Nextdoor - tap into your neighbourhood." `https://nextdoor.co.uk/`.
Accessed: 2021-06-30.

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

[293] 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.

[294] 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*, pp. 1–7, 2011.

[295] "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.

[296] 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*, pp. 3–18, Springer International Publishing, Dec. 2020.

[297] The City of Chicago, "The City Of Chicago Technology Plan," 2013.

[298] M. Zappatore, A. Longo, and M. A. Bochicchio, "Crowd-sensing our smart cities: A platform for noise monitoring and acoustic urban planning," *Journal of Communications Software and Systems*, vol. 13, no. 2, pp. 53–67, 2017.

[299] BT, "Bt street hubs | bt business," 2022.
Accessed:2022-11-30.

[300] B. Y. O. Low, S. H. Dahlan, and M. H. Abd Wahab, "Real-time bus location and arrival information system," in *2016 IEEE Conference on Wireless Sensors (ICWiSE)*, pp. 50–53, 2016.

[301] L. Belli, A. Cilfone, L. Davoli, G. Ferrari, P. Adorni, F. Di Nocera, A. Dall'Olio, C. Pellegrini, M. Mordacci, and E. Bertolotti, "Iot-enabled smart sustainable cities: challenges and approaches," *Smart Cities*, vol. 3, no. 3, pp. 1039–1071, 2020.

[302] 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 Computing Surveys*, vol. 50, Nov. 2017.

[303] A. M. S. Osman, "A novel big data analytics framework for smart cities," *Future Generation Computer Systems*, vol. 91, pp. 620–633, 2019.

[304] I. Ganchev, Z. Ji, and M. O'Droma, "A generic iot architecture for smart cities," 2014.

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

[306] A. Krylovskiy, M. Jahn, and E. Patti, "Designing a smart city internet of things platform with microservice architecture," in *2015 3rd international conference on future internet of things and cloud*, pp. 25–30, IEEE, 2015.

[307] L. Calderoni, A. Magnani, and D. Maio, "Iot manager: An open-source iot framework for smart cities," *Journal of Systems Architecture*, vol. 98, pp. 413–423, 2019.

[308] N. Z. Bawany and J. A. Shamsi, "Smart city architecture: Vision and challenges," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 11, 2015.

[309] A. B. Haque, B. Bhushan, and G. Dhiman, "Conceptualizing smart city applications: Requirements, architecture, security issues, and emerging trends," *Expert Systems*, vol. 39, no. 5, p. e12753, 2022.

[310] K. Laubhan, K. Talaat, S. Riehl, M. S. Aman, A. Abdelgawad, and K. Yelamarthi, "A low-power iot framework: From sensors to the cloud," in *2016 IEEE International Conference on Electro Information Technology (EIT)*, pp. 0648–0652, IEEE, 2016.

[311] P. Bellini, P. Nesi, M. Paolucci, and I. Zaza, "Smart city architecture for data ingestion and analytics: Processes and solutions," in *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 137–144, IEEE, 2018.

[312] R. Jalali, K. El-Khatib, and C. McGregor, "Smart city architecture for community level services through the internet of things," in *2015 18th International Conference on Intelligence in Next Generation Networks*, pp. 108–113, IEEE, 2015.

[313] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129, 2017.

[314] A. Gaur, B. Scotney, G. Parr, and S. McClean, "Smart city architecture and its applications based on iot," *Procedia computer science*, vol. 52, pp. 1089–1094, 2015.

[315] M. Al-Hader and A. Rodzi, "The smart city infrastructure development & monitoring," *Theoretical and Empirical Researches in Urban Management*, vol. 4, no. 2 (11, pp. 87–94, 2009.

[316] E. Theodoridis, G. Mylonas, and I. Chatzigiannakis, "Developing an iot smart city framework," in *IISA 2013*, pp. 1–6, IEEE, 2013.

[317] L. Sánchez, I. Elicegui, J. Cuesta, L. Munoz, and J. Lanza, "Integration of utilities infrastructures in a future internet enabled smart city framework," *Sensors*, vol. 13, no. 11, pp. 14438–14465, 2013.

[318] A. Dagliati, L. Sacchi, M. Bucalo, D. Segagni, K. Zarkogianni, A. M. Millana, J. Cancela, F. Sambo, G. Fico, M. T. M. Barreira, *et al.*, "A data gathering framework to collect type 2 diabetes patients data," in *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 244–247, IEEE, 2014.

[319] C. Wang, J. Li, F. Ye, and Y. Yang, "A mobile data gathering framework for wireless rechargeable sensor networks with vehicle movement costs and capacity constraints," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2411–2427, 2015.

[320] K. Yuen, B. Liang, and B. Li, "A distributed framework for correlated data gathering in sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 578–593, 2008.

[321] W. Choi and S. K. Das, "A novel framework for energy-conserving data gathering in wireless sensor networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3, pp. 1985–1996, IEEE, 2005.

[322] S. Say, H. Inata, J. Liu, and S. Shimamoto, "Priority-based data gathering framework in uav-assisted wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 14, pp. 5785–5794, 2016.

[323] X. Xiang, J. Gui, and N. N. Xiong, "An integral data gathering framework for supervisory control and data acquisition systems in green iot," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 714–726, 2021.

[324] M. Zhao, J. Li, and Y. Yang, "A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2689–2705, 2014.

[325] IBM, "Ibm | internet of things reference architecture," 2022. Accessed:2022-11-30.

[326] Toshiba, "'toshiba iot reference architecture" and "toshiba spinex"," 2022.

[327] Amazon, "'iot | aws architecture center"," 2022.

[328] University of Bristol, "Jean golding institute," 2022. Accessed:2022-11-30.

[329] C. Catlett, P. Beckman, N. Ferrier, H. Nusbaum, M. E. Papka, M. G. Berman, and R. Sankaran, "Measuring cities with software-defined sensors," *Journal of Social Computing*, vol. 1, no. 1, pp. 14–27, 2020.

[330] M. B. Ribeiro and K. R. Braghetto, "A data integration architecture for smart cities," in *Anais do XXXVI Simpósio Brasileiro de Bancos de Dados (SBBD)*, 2021.

[331] DAFNI, "Dafni facility roadshows and events," 2021.
[Online; accessed 4-December-2022].

[332] B. Matthews, S. Chorlton, P. Oliver, R. Fowler, and E. Yang, "Designing dafni: a national facility for modelling infrastructure," in *9th Conference on Adding value and preserving data*, pp. 172–176, 2018.

[333] T. Tryfonas and P. Tully, "Bristol infrastructure collaboratory," 2022.
[Online; accessed 4-Nov-2022].

[334] L. Leite, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles, "A survey of devops concepts and challenges," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–35, 2019.

[335] M. Treveil, N. Omont, C. Stenac, K. Lefevre, D. Phan, J. Zentici, A. Lavoillotte, M. Miyazaki, and L. Heidmann, *Introducing MLOps*.
O'Reilly Media, 2020.

[336] S. Sagiroglu and D. Sinanc, "Big data: A review," in *2013 international conference on collaboration technologies and systems (CTS)*, pp. 42–47, IEEE, 2013.

[337] Y. Brikman, *Terraform: Up and Running*.
" O'Reilly Media, Inc.", 2022.

[338] S. Pandya and R. Guha Thakurta, "Hands-on infrastructure as code with puppet," in *Introduction to Infrastructure as Code*, pp. 135–163, Springer, 2022.

[339] M. Howard, "Helm–what it can do and where is it going?," *arXiv preprint arXiv:2206.07093*, 2022.

[340] J. Dobies and J. Wood, *Kubernetes operators: Automating the container orchestration platform*.
O'Reilly Media, 2020.

[341] A. Vazquez, "Starting with freeipa," in *Practical LPIC-3 300*, pp. 555–577, Springer, 2019.

[342] G. Camposo, "Securing web services with keycloak," in *Cloud Native Integration with Apache Camel*, pp. 77–115, Springer, 2021.

[343] A. O'grady, *GitLab Quick Start Guide: Migrate to GitLab for all your repository management solutions*.
Packt Publishing Ltd, 2018.

[344] M. Sharif and G. Lueckemeyer, "Coaas: Continuous integration and delivery framework for hpc using gitlab-runner," in *Proceedings of the 2020 the 4th International Conference on Big Data and Internet of Things*, pp. 54–58, 2020.

[345] E. Utami, H. Al Fatta, *et al.*, "Analysis on the use of declarative and pull-based deployment models on gitops using argo cd," in *2021 4th International Conference on Information and Communications Technology (ICOIACT)*, pp. 186–191, IEEE, 2021.

[346] B. Burns, J. Beda, K. Hightower, and L. Evenson, *Kubernetes: up and running*.
" O'Reilly Media, Inc.", 2022.

[347] V.-C. Le and M. Yoo, "Lightweight k3s tool for internet of things environments," vol. 46, no. 11, pp. 1958–1964, 2021.

[348] Y. Xiong, Y. Sun, L. Xing, and Y. Huang, "Extend cloud to edge with kubeedge," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 373–377, IEEE, 2018.

[349] OpenYurt, "Openyurt | an open platform that extends upstream kubernetes to edge," 2022.
Accessed:2022-11-30.

[350] SuperEdge, "Superedge," 2022.
Accessed:2022-11-30.

[351] NVIDIA, "Nvidia device plugin for kubernetes," 2022.
Accessed:2022-11-30.

[352] NVIDIA, "Nvidia gpu feature discovery," 2022.
Accessed:2022-11-30.

[353] LFEdge, "Fledge," 2022.
Accessed:2022-11-30.

[354] rook, "rook | open-source, cloud-native storage for kubernetes," 2022.
Accessed:2022-11-30.

[355] RedHat, "Red hat ceph storage," 2022.
Accessed:2022-11-30.

[356] J. Salamero, "Kubernetes runtime security with falco and sysdig," 2019.

[357] N. Sabharwal, S. Pandey, and P. Pandey, "Getting started with vault," in *Infrastructure-as-Code Automation Using Terraform, Packer, Vault, Nomad and Consul*, pp. 131–150, Springer, 2021.

[358] J. A. Donenfeld, "Wireguard: next generation kernel network tunnel.," in *NDSS*, pp. 1–12, 2017.

[359] M. Feilner, *OpenVPN: Building and integrating virtual private networks*. Packt Publishing Ltd, 2006.

[360] Aquasec, "Starboard," 2022. Accessed:2022-11-30.

[361] C. Gormley and Z. Tong, *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.", 2015.

[362] E. Silva, "Fluentd: a high performance unified logging layer," 2016.

[363] V. Sharma, "Getting started with kibana," in *Beginning Elastic Stack*, pp. 29–44, Springer, 2016.

[364] J. Turnbull, *Monitoring with Prometheus*. Turnbull Press, 2018.

[365] I. Tien, "Why we made mattermost an open-source slack alternative," 2015.

[366] OperatorHub.io, "Database | operatorhub.io | the registry for kubernetes operators," 2022. [Online; accessed 4-Nov-2022].

[367] Strimzi, "Strimzi - apache kafka on kubernetes," 2022. [Online; accessed 4-Nov-2022].

[368] OperatorHub.io, "Streaming and messaging | operatorhub.io | the registry for kubernetes operators," 2022. [Online; accessed 4-Nov-2022].

[369] Seldon, "Seldon core - the open-source framework for easily and quickly deploying models and experiments at scale.," 2022. [Online; accessed 4-Nov-2022].

[370] L. Fernández, R. Andersson, H. Hagenrud, T. Korhonen, E. Laface, B. Zupanc, *et al.*, "Jupyterhub at the ess. an interactive python computing environment for scientists and engineers," in *This conference*, 2016.

[371] S. Venkatramulu, M. S. B. Phridviraj, C. Srinivas, and V. C. S. Rao, "Implementation of grafana as open source visualization and query processing platform for data scientists and researchers," *Materials Today: Proceedings*, 2021.

[372] Knowage, "Knowage | open source analytics and business intelligence," 2022. Accessed:2022-11-30.

[373] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng, and E. Kovacs, "A standard-based open source iot platform: Fiware," *IEEE Internet of Things Magazine*, vol. 2, no. 3, pp. 12–18, 2019.

[374] Fiware, "Fiware | helm repository for generic enablers," 2022. [Online; accessed 4-Nov-2022].

[375] Red Hat, "Managing idm users, groups, hosts, and access control rules," 2022. [Online; accessed 4-Nov-2022].

[376] Phase Two, Inc, "Organizations for keycloak - single realm, multi-tenancy for saas apps," 2022. [Online; accessed 4-Nov-2022].

[377] Gitlab, "Gitlab - set up your organization," 2022. [Online; accessed 4-Nov-2022].

[378] Gitlab, "Projects - argo cd - declarative gitops cd for kubernetes," 2022. [Online; accessed 4-Nov-2022].

[379] Kubernetes, "Multi-tenancy," 2022. [Online; accessed 4-Nov-2022].

[380] InfluxDB, "Manage organisations in influxdb," 2022. [Online; accessed 4-Nov-2022].

[381] Strimzi, "Using open policy agent with strimzi and apache kafka," 2022. [Online; accessed 4-Nov-2022].

[382] iterative.ai, "Get started with cml on gitlab," 2022. [Online; accessed 4-Nov-2022].

[383] Grafana Labs, "Manage organizations," 2022. [Online; accessed 4-Nov-2022].

[384] JupyterHub, "Jupyterhub," 2022. [Online; accessed 4-Nov-2022].

[385] R. Kumar, L. Adwani, S. Kumawat, and S. K. Jangir, "Opennebula: Open source iaas cloud computing software platforms," in *National Conference on Computational and Mathematical Sciences (COMPUTATIA-IV), Technically Sponsored By: ISITA and RAOPS, Jaipur*, 2014.

[386] M. Azure, "Microsoft azure," *línea]. Available: https:/ /docs. microsoft. com/es-es/azure/virtual-machines/linux/quick-createportal.[Último acceso: 10 Diciembre 2017]*, 2016.

[387] BigClouT, "Bigclout | big data meeting cloud and iot for empowering the citizen clout in smart cities," 2022.
Accessed:2022-11-30.

[388] P. Fremantle *et al.*, "A reference architecture for the internet of things," *WSO2 White paper*, pp. 02–04, 2015.

[389] S. T. Hospitals, "Perfect Patient Pathway Test Bed Overview Report 2," 2016.

[390] U. A. Raymond, S. Vadgama, S. Crowe, S. Morris, and M. Utley, "Evaluation of the nhs england innovation test bed at care city. accessed: 2020-06-30," 2019.

[391] P. Yadav, V. Safronov, and R. Mortier, "Enforcing accountability in smart built-in iot environment using mud," in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys'19, (New York, NY, USA), p. 368–369, Association for Computing Machinery, 2019.

[392] J. Rubio-Aparicio, F. Cerdan-Cartagena, J. Suardiaz-Muro, and J. Ybarra-Moreno, "Design and implementation of a mixed IoT LPWAN network architecture," *Sensors (Switzerland)*, vol. 19, no. 3, 2019.

[393] Pycom, "Fipy experiment board," 2020.
Accessed: 2020-06-30.

[394] Semtech, "LoRa and LoRaWAN," *Semtech Technique Paper*, no. December 2019, pp. 1–17, 2020.

[395] Alliot Technologies, "Robeau lorawan water flow meter," 2020.
Accessed:2020-06-30.

[396] LoRa Alliance Technical Commitee, "LoRaWAN 1.1 Specification," *LoRaWAN 1.1 Specification*, no. 1.1, p. 101, 2017.

[397] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019.

[398] Technical Marketing Workgroup, "A technical overview of LoRa® and LoRaWAN," no. November, 2015.

[399] L. D. Portal, "An in-depth look at lorawan® class a devices," 2020.
Accessed:2020-06-30.

[400] Sigfox, "Letterbox sensor." https://partners.sigfox.com/products/letterbox-sensor, 2020.
Accessed:2020-06-30.

[401] Sigfox, "Sigfox Technical Overview," vol. 1, no. May, p. 26, 2017.

[402] GSMA, "Lte-m deployment guide to basic feature set requirements," 2020.
Accessed:2020-06-30.

[403] S. Øyvann, "Internet of sheep: Why world's biggest nb-iot pilot is roping in woolly helpers." https://www.zdnet.com/article/internet-of-sheep-why-worlds-biggest-nb-iot-pilot-is-roping-in-woolly-helpers/, 2020.
Accessed:2020-06-30.

[404] Arkessa, "Cellular iot solution guide," 2019.
Accessed:2020-06-30.

[405] GSM, "LTE-M Commercialisation. Case study how AT&T and Telstra connect million more IoT devices.," pp. 1–10, 2019.

[406] GSMA, "NB-IoT Deployment guide to basic feature set requirements," *Gsma*, vol. Release 3, no. June, pp. 1–80, 2019.

[407] J. Harbin, A. Burns, R. I. Davis, L. S. Indrusiak, I. Bate, and D. Griffin, "The AirTight Protocol for Mixed Criticality Wireless CPS," *ACM Trans. Cyber-Phys. Syst.*, vol. 4, no. 2, 2019.

[408] I. Correia, L. Gouveia, and F. Saldanha-da Gama, "Solving the variable size bin packing problem with discretized formulations," *Computers & OR*, vol. 35, pp. 2103–2113, June 2008.

[409] Raspberry Pi, "Raspberry pi 4 model," 2020.
Accessed:2020-06-30.

[410] Pycom, "Fipy datasheet." https://docs.pycom.io/gitbook/assets/specsheets/Pycom_002_Specsheets_FiPy_v2.pdf, 2020.
Accessed:2020-06-30.

[411] ARM, “Smarter device manager,” 2022.
[Online; accessed 4-Nov-2022].

[412] “Fipy: How to send data via simultaneous connected with both wi-fi and lte (nb-iot/ cat-m).” https://forum.pycom.io/topic/6204/fipy-how-to-send-data-via-simultaneous-connected-with-both-wi-fi-and-lte-nb-iot-c 2020.
2020-06-30.

[413] LoRa™ Alliance, “LoRaWAN™ Regional Parameters v1.1rA,” *LoRaWAN™ 1.1 Specif.*, p. 56, 2017.

[414] The Things Network, “Lorawan airtime calculator,” 2020.
Accessed:2020-06-30.

[415] The Things Network, “Duty cycle,” 2020.
Accessed:2020-06-30.

[416] The Things Network, “Best practices to limit application payloads,” 2020.
Accessed:2020-06-30.

[417] Sigfox Support, “Link quality: general knowledge,” 2020.
Accessed:2020-06-30.

[418] R. L. Bras, “What is the difference in data throughput between lte-m/nb-iot and 3g or 4g?,” 2020.
Accessed:2020-06-30.

[419] PyCom, “Vodafone nb-iot prepaid subscription,” 2020.
Accessed:2020-06-30.

[420] shawwwn | Github Gist, “μping: Ping library for micropython,” 2020.
Accessed:2020-06-30.

[421] shawwwn, “uping usage - micropython forum,” 2020.
Accessed:2020-06-30.

[422] The Things Network, “Frequency plans,” 2020.
Accessed:2020-06-30.

[423] “Esp32 wi-fi throughput,” 2020.
Accessed:2020-06-30.

[424] D. George, “uiperf3 | pure python, iperf3-compatible network performance test tool.,” 2020.
Accessed:2020-06-30.

[425] Smart Citizen Kit, "Smart citizen kit docs." https://docs.smartcitizen.me/Smart%20Citizen%20Kit/, 2020.
Accessed:2020-06-30.

[426] "Event based programming/ interrupts | fipy | network (wifi/lte/lora) disconnection." https://forum.pycom.io/topic/6341/event-based-programming-interrupts-fipy-network-wifi-lte-lora-disconnection, 2020.
2020-06-30.

[427] TELTONIKA, "Teltonika industrial cellular modem with multiple lpwan connectivity options (trm250)." https://www.wifi-stock.com/details/teltonika-industrial-cellular-modem-trm250.html, 2020.
Accessed:2020-06-30.

[428] MultiTech, "Multitech brochure product lpwa solutions for industrial iot." https://www.multitech.com/documents/publications/brochures/MT_Brochure_Product_LPWA_2019-10-15.pdf, 2020.
Accessed:2020-06-30.

[429] A. Rullo, E. Serra, and J. Lobo, "Redundancy as a measure of fault-tolerance for the internet of things: A review," *Policy-Based Autonomic Data Governance*, pp. 202–226, 2019.

[430] S. M. Oteafy and H. S. Hassanein, "Resilient iot architectures over dynamic sensor networks with adaptive components," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 474–483, 2017.

[431] B. S. Chaudhari, M. Zennaro, and S. Borkar, "Lpwan technologies: Emerging application characteristics, requirements, and design considerations," *Future Internet*, vol. 12, no. 3, 2020.

[432] M. I. Hossain and J. I. Markendahl, "Comparison of lpwan technologies: Cost structure and scalability," *Wireless Personal Communications*, vol. 121, pp. 887–903, Nov. 2021.

[433] J. Santos, P. Leroux, T. Wauters, B. Volckaert, and F. De Turck, "Anomaly detection for Smart City applications over 5G low power wide area networks," *IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018*, pp. 1–9, 2018.

[434] G. Roque and V. S. Padilla, "LPWAN Based IoT Surveillance System for Outdoor Fire Detection," *IEEE Access*, vol. 8, pp. 114900–114909, 2020.

[435] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Towards end-to-end resource provisioning in Fog Computing over Low Power Wide Area Networks," *Journal of Network and Computer Applications*, vol. 175, no. August 2020, 2021.

[436] M. M. Tajiki, S. H. G. Petroudi, S. Salsano, S. Uhlig, and I. Castro, "Optimal estimation of link delays based on end-to-end active measurements," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4730–4743, 2021.

[437] H. Kim, E. Kang, D. Broman, and E. A. Lee, "Resilient Authentication and Authorization for the Internet of Things (IoT) Using Edge Computing," *ACM Transactions on Internet of Things*, vol. 1, no. 1, pp. 1–27, 2020.

[438] A. Orive, A. Agirre, H.-L. Truong, I. Sarachaga, and M. Marcos, "Quality of service aware orchestration for cloud-edge continuum applications," *Sensors (Basel)*, vol. 22, Feb. 2022.

[439] C. Cicconetti, M. Conti, and A. Passarella, "In-network computing with function as a service at the edge," *Computer*, vol. 55, pp. 65–73, Sept. 2022.

[440] S. Guo, K. Zhang, B. Gong, W. He, and X. Qiu, "A delay-sensitive resource allocation algorithm for container cluster in edge computing environment," *Computer Communications*, vol. 170, pp. 144–150, 2021.

[441] I. Pelle, M. Szalay, J. Czentye, B. Sonkoly, and L. Toka, "Cost and latency optimized edge computing platform," *Electronics*, vol. 11, no. 4, 2022.

[442] T. Pfandzelter and D. Bermbach, "tinyfaas: A lightweight faas platform for edge environments," in *2020 IEEE International Conference on Fog Computing (ICFC)*, pp. 17–24, 2020.

[443] Z. Qin, G. Denker, C. Talcott, and N. Venkatasubramanian, "Achieving resilience of heterogeneous networks through predictive, formal analysis," *HiCoNS 2013 - Proceedings of the 2nd ACM International Conference on High Confidence Networked Systems, Part of CPSWeek 2013*, pp. 85–92, 2013.

[444] Z. Qin, L. Iannario, C. Giannelli, P. Bellavista, G. Denker, and N. Venkatasubramanian, "MINA: A reflective middleware for managing dynamic multinetwork environments," *IEEE/IFIP NOMS 2014 - IEEE/IFIP Network Operations and Management Symposium: Management in a Software Defined World*, pp. 3–6, 2014.

[445] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A software defined networking architecture for the internet-of-things," *IEEE/IFIP NOMS 2014 - IEEE/IFIP Network Operations and Management Symposium: Management in a Software Defined World*, 2014.

[446] M. Y. S. Uddin, A. Nelson, K. Benson, G. Wang, Q. Zhu, Q. Han, N. Alhassoun, P. Chakravarthi, J. Stamatakis, D. Hoffman, L. Darcy, and N. Venkatasubramanian, "The Scale2 Multi-Network Architecture for IoT-Based Resilient Communities," *2016 IEEE International Conference on Smart Computing, SMARTCOMP 2016*, 2016.

[447] A. Modarresi and J. Symons, "Resilience and technological diversity in smart homes: A graph-theoretic approach to modeling IoT systems with integrated heterogeneous networks," *Journal of Ambient Intelligence and Humanized Computing*, no. Alliance 2008, 2020.

[448] S. Chaterji, P. Naghizadeh, M. A. Alam, S. Bagchi, M. Chiang, D. Corman, B. Henz, S. Jana, N. Li, S. Mou, M. Oishi, C. Peng, T. Rompf, A. Sabharwal, S. Sundaram, J. Weimer, and J. Weller, "Resilient Cyberphysical Systems and their Application Drivers: A Technology Roadmap," pp. 1–36, 2019.

[449] Y. Harchol, A. Mushtaq, V. Fang, J. McCauley, A. Panda, and S. Shenker, "Making edge-computing resilient," in *Proceedings of the 11th ACM Symposium on Cloud Computing*, SoCC'20, (New York, NY, USA), p. 253–266, Association for Computing Machinery, 2020.

[450] D. F. Carvalho, P. Ferrari, E. Sisinni, and A. Flammini, "Improving Redundancy in LoRaWAN for Mixed-Criticality Scenarios," *IEEE Systems Journal*, pp. 1–10, 2020.

[451] D. Hodgson, "Helping doctoral students understand phd thesis examination expectations: A framework and a tool for supervision," *Active learning in higher education*, vol. 21, no. 1, pp. 51–63, 2020.