UNIVERSIDADE FEDERAL DE OURO PRETO

USING BLOCKCHAIN AND LOW POWER IN SMART CITIES TO INTERNET OF THIGS APPLICATIONS: A FOG COMPUTING APPROACH

Celio Marcio Soares Ferreira

Advisors: Ricardo Augusto Rabelo Oliveira
Jorge Sá Silva

Ouro Preto
November 2022

UNIVERSIDADE FEDERAL DE OURO PRETO

UFOP

decom departamento de computação

USING BLOCKCHAIN AND LOW POWER IN SMART CITIES TO INTERNET OF THIGS APPLICATIONS: A FOG COMPUTING APPROACH

Celio Marcio Soares Ferreira

Doctoral Thesis presented to the Postgraduate Program in Computer Science, of the Universidade Federal de Ouro Preto, as part of the requirements for obtaining the title of Doctor in Computer Science.

Advisors: Ricardo Augusto Rabelo Oliveira
Jorge Sá Silva

Ouro Preto
November 2022

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

F383u Ferreira, Celio Marcio Soares.
Using Blockchain and Low Power in Smart Cities to Internet of Thigs Applications [manuscrito]: A Fog Computing Approach. / Celio Marcio Soares Ferreira. - 2022.
176 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Ricardo Augusto Rabelo Oliveira.
Coorientador: Prof. Dr. Jorge Sa Silva.
Tese (Doutorado). Universidade Federal de Ouro Preto. Departamento de Computação. Programa de Pós-Graduação em Ciência da Computação.
Área de Concentração: Ciência da Computação.

1. Ethereum. 2. Blockchain. 3. BLE. 4. LoRa. 5. WPAN. 6. LPWAN. 7. Smart Cities. 8. Semantic Web. 9. Smart Contract. I. Oliveira, Ricardo Augusto Rabelo. II. Silva, Jorge Sa. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 004

Bibliotecário(a) Responsável: Luciana De Oliveira - SIAPE: 1.937.800

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO
PROGRAMA DE POS-GRADUACAO EM CIENCIA DA COMPUTACAO

FOLHA DE APROVAÇÃO

Célio Márcio Soares Ferreira

Usando Blockchain e redes Low Power em Smart Cities para aplicações internet das coisas: uma abordagem Fog Coputing

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Doutor em Ciência da Computação

Aprovada em 19 de agosto de 2022

Membros da banca

Prof. Dr. Ricardo Augusto Rabelo Oliveira - Orientador - Universidade Federal de Ouro Preto
Prof. Dr. Andre Luiz Lins de Aquino - Universidade Federal de Alagoas
Prof. Dr. Carlos Frederico Marcelo da Cunha Cavalcanti - Universidade Federal de Ouro Preto
Prof. Dr. Heitor Soares Ramos Filho - Universidade Federal de Minas Gerais
Prof. Dr. Luiz Henrique Andrade Correia - Universidade Federal de Lavras
Prof. Dr. Saul Emanuel Delabrida Silva - Universidade Federal de Ouro Preto
Prof. Dr. Jorge Miguel Sá Silva - Universidade de Coimbra

Prof. Dr. Ricardo Augusto Rabelo Oliveira, orientador do trabalho, aprovou a versão final e autorizou seu depósito no Repositório Institucional da UFOP em 20/10/2022

Documento assinado eletronicamente por **Gladston Juliano Prates Moreira**, **COORDENADOR(A) DE CURSO DE PÓS-GRADUACÃO EM CIÊNCIA DA COMPUTAÇÃO**, em 20/10/2022, às 16:33, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0414978** e o código CRC **930FCFF1**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: (31)3559-1641 - www.ufop.br

"The love for all living creatures is the most noble attribute of man." - Charles Darwin.

I dedicate this work to

My son, for the notion of importance and responsibility towards future generations. My mother, for her constant and eternal struggle to motivate and nurture me in the most challenging moments of my journey. My late father, who did not have the opportunity to experience this point, reached in my training. To my supervisors and teachers for their patience and dedication to my not always straight-line search for knowledge. To my co-workers who kept the work in quality and performance, helping me with competence and promptness in my moments of absence. To my friends and lab colleagues, who made this journey smoother and richer in information.

Thanks to FAPEMIG, CNPq, CAPES and UFOP (Federal University of Ouro Preto).

Abstract of Thesis presented to UFOP as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

USING BLOCKCHAIN AND LOW POWER IN SMART CITIES TO INTERNET OF THIGS APPLICATIONS: A FOG COMPUTING APPROACH

Celio Marcio Soares Ferreira

November/2022

Advisors: Ricardo Augusto Rabelo Oliveira
Jorge Sá Silva

Department: Computer Science

With the advent and popularization of Internet of Things (IoT) devices, new possibilities for applications that use data extracted from the things we use in everyday life arise. Cars, wearables, health sensors, and home appliances will generate unprecedented amounts of data and bring insights that will revolutionize our daily routines. A potential scenario significantly impacted is Smart Cities (SC), which uses devices spread out on a large scale in an urban environment to extract traffic, weather, and equipment maintenance data to obtain insights acting on city management and disaster prevention. The network infrastructure currently available for these network applications uses proprietary communication technologies and is dependent on mobile phone companies. Their systems are proprietary, centralized, isolated from other databases, and constantly exposed to Single Point of Failure (SPOF). IoT applications are still primarily embryonic and do not provide reliable verification of the data source at the edge, as in the case of IoT devices, often with outdated firmware. Our work investigates the use in SC of a composition of Low Power Wide Area Networks (LPWAN) and the popular Personal Area Networks (PAN), independence of mobile network providers, and Low Power consumption. For this, we used development kits with LoRa and BLE to verify the feasibility and possible problems in this integration, and we evaluated the scalability of LoRa using a simulator. Security gaps in IoT Apps in Smart Cities mainly come from the difficulty of knowing and trusting edge devices. The problem of standardizing and updating these devices during their lifetime justifies our search for using tools that support transparency, scalability, reliability, resilience, and implicit requirements of decentralized Blockchain networks that support Smart Contracts. For

this, we present a network architecture using Fog Computing and Smart Contracts Blockchain, which, through API gateways, authorizes and authenticates edge communication from IoT devices previously known by their metadata and firmware. To provide standard and link data from Blockchain with existing Web datasets, we use and add new components to ontologies that model Ethereum entities. This approach allows us to use the semantic web for data consumption and linking, which exposes data from Ethereum networks in soft-realtime through middleware. This work investigates the potential use of Fog Computing in SC in Low Power networks, strategies to identify and authenticate IoT devices at the edges using Blockchain and Smart Contract, and consumption and data link of Blockchain with the current web using the Semantic web. The set of these resources used in Fog computing allows searching for a composition of independent SC network infrastructures, Low Power, with reliable information coming from the edges and integrable with other pre-existing data sets. As the main results, we show the limits of the LoRa network, using a simulator in single-gateway and multi-gateway scenarios. We present scenarios of mixed use of traditional using Blockchain as authentication and validation background, by API gateway in Fog Computing architecture, and we present the times in transactions per second of this approach considering signatures and validation of payloads using Ethereum Blockchain. We present a middleware to expose Ethereum data in soft-realtime using ontologies that model Ethereum in the literature and extended by our EthExtras ontology, providing classes and properties for links and queries.The main advances of this work are the models using the Fog Computing paradigm for Smart Cities, where we present its use as a mixing point of LoRa and BLE and the Blockchain API Gateway to validate data from IoT devices. In addition to our Middleware for extracting and consuming Ethereum data in soft real-time using our EthExtras and EthOn vocabulary.

Resumo da Tese apresentada à UFOP como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

USANDO BLOCKCHAIN E REDES LOW POWER EM SMART CITIES PARA APLICAÇÕES INTERNET DAS COISAS: UMA ABORDAGEM FOG COMPUTING

Celio Marcio Soares Ferreira

Novembro/2022

Orientadores: Ricardo Augusto Rabelo Oliveira
Jorge Sá Silva

Programa: Ciência da Computação

Com o advento e popularização dos dispositivos da Internet das Coisas (IoT), surgem novas possibilidades de aplicações que utilizam dados extraídos das coisas que usamos no dia a dia. Carros, wearables, sensores de saúde e eletrodomésticos gerarão quantidades sem precedentes de dados e trarão insights que revolucionarão nossas rotinas diárias. Um cenário potencial impactado significativamente é o Smart Cities (SC), que utiliza dispositivos espalhados em grande escala em um ambiente urbano para extrair dados de tráfego, clima e manutenção de equipamentos, para obter insights que atuam na gestão da cidade e prevenção de desastres. A infraestrutura de rede atualmente disponível para esses aplicativos de rede usa tecnologias de comunicação proprietárias e depende das empresas de telefonia móvel. Seus sistemas são proprietários, centralizados, isolados de outros bancos de dados e constantemente expostos a Single Point of Failure (SPOF). Os aplicativos IoT ainda são principalmente embrionários e não fornecem verificação confiável da fonte de dados na borda, como no caso de dispositivos IoT, muitas vezes com firmware desatualizado. Nosso trabalho investiga o uso em SC de uma composição de Low Power Wide Area Networks (LPWAN) e as populares Personal Area Networks (PAN), buscando independência de provedores de rede móvel e baixo consumo de energia. Para isso, utilizamos kits de desenvolvimento com LoRa e BLE para verificar a viabilidade e possíveis problemas nesta integração, e avaliamos a escalabilidade do LoRa utilizando um simulador. As lacunas de segurança em aplicativos de IoT em cidades inteligentes vêm principalmente da dificuldade de conhecer e confiar em dispositivos de borda. O problema de padronizar e atualizar esses dispositivos durante sua vida útil justifica o uso de

ferramentas que suportam transparência, escalabilidade, confiabilidade, resiliência e requisitos implícitos de redes Blockchain descentralizadas que suportam Smart Contracts. Para isso, apresentamos uma arquitetura de rede utilizando Fog Computing e Smart Contracts Blockchain, que, por meio de API gateways, autoriza e autentica a comunicação de borda de dispositivos IoT anteriormente conhecidos por seus metadados e firmware. Para fornecer dados padrão e de link do Blockchain com conjuntos de dados da Web existentes, usamos e adicionamos novos componentes a ontologias que modelam entidades Ethereum. Essa abordagem nos permite usar a web semântica para consumo e link de dados, expondo dados de redes Ethereum em soft-realtime por meio de um middleware. Este trabalho investiga o potencial de uso de Fog Computing em SC em redes Low Power, e estratégias para identificar e autenticar dispositivos IoT nas bordas utilizando Blockchain e Smart Contract, provendo consumo e enlace de dados de Blockchain com a web atual utilizando a web semântica. O conjunto desses recursos utilizados na Fog computing permite buscar uma composição de infraestruturas de rede SC independentes, Low Power, com informações confiáveis provenientes das bordas e integráveis com outros conjuntos de dados pré-existentes. Como principais resultados, mostramos os limites da rede LoRa, usando simulador em cenarios de um gateway e multigateway. Apresentamos cenarios de uso hibrido de aploicacoes tradicionais e Blockchain, usado API gateway em arquitetura Fog Computing, e aprsentamos os tempos em transacoes por segundo desta abordagem considerando assintauras e validação dos payloads usando Blockchain Ethreum. Apresentamos um midleware para expor dados do Ethreum em soft-realtime usando ontologias que modelam o Ethereum na literatura e extendidda pela nossa ontologia EthExtras, fornecendo classes e proriedades para links e consultas. Os principais avanços deste trabalho são os modelos utilizando o paradigma Fog Computing para Smart Cities, onde apresentamos seu uso como ponto de mistura de LoRa e BLE e o Blockchain API Gateway para validar dados provenientes de dispositivos IoT. Alem de nosso Middleware para extração e consumo de dados Ethereum em soft real-time usando nosso vocabulário EthExtras e EthOn.

Contents

List of Figures

List of Tables

Acronyms

ABI Application binary interface. 8, 17, 21

ABP Activation by Personalization. 52, 124

ADEPT Autonomous Decentralized. Peer-to-Peer Telemetry. 37

ADR Adaptive Data Rate. 53

AI Artificial Intelligence. 119, 121, 147, 148

API Application Programming Interface. 3, 5–10, 12–14, 23, 25, 30, 70, 71, 79–82, 84–86, 90, 92, 93, 95, 101, 102, 109, 116, 120, 122, 124, 125, 132, 134–138, 140, 144, 145, 149–151

ASPE Asymmetric Scalar Product Preservation. 121

BLE Bluetooth Low Energy. 5, 6, 11, 28, 29, 43–47, 51, 52, 61, 119, 124–126, 139, 148

BOL Bills of Lading. 37

COT The Chain of Things. 37

CPS Cyber-Physical Systems. 113, 121

DApp Decentralized Application. 7, 8, 10, 12, 17, 21–23, 31, 32, 93, 101, 103–105, 109, 111, 112, 114, 132–134, 136–138, 150

DDF Decentralized Development Fund. 38

DDoS Distributed Denial of Service. 6, 130, 139, 140

DeFi Decentralized Finance. 105, 109, 111, 130, 133, 143

DER Data Extraction Rate. 29, 53, 58

DNS Domain Name System. 22, 111, 132

ECDSA Elliptic Curve Digital Signature Algorithm. 18

ENS Ethereum Name Service. 22, 111, 116, 132, 149, 150

ERC Ethereum Request for Comments. 22, 149

ETH Ether. 21, 22, 104, 133, 134, 150

EVM Ethereum Virtual Machine. 21, 120, 132

GA-DT Genetic Algorithm-based Decision Tree. 120, 121

GA-SVM Genetic Algorithm-based Support Vector Machine. 120

GATT Generic Attribute. 46, 47, 51, 61, 139

GEth GoEthereum. 22, 70–72

I4.0 Industry 4.0. 31, 111, 119–121, 149, 151

IIoT Industrial IoT. 31, 35, 121

IoHT Internet of Healthcare Things. 120

IoM Internet of Money. 103

IoST Internet of Sensor Things. 120

IoT Internet of Things. 2–13, 17, 19, 21, 24, 25, 27–32, 36–41, 43–48, 50, 53, 58, 60, 62–72, 74–76, 78–87, 90, 92, 93, 96, 101, 102, 111–115, 118–124, 128–130, 132, 134–138, 143–149, 151

IoTApp IoT Application. 47, 48, 51, 52

IPFS Interplanetary File System. 8, 22, 23, 40, 41, 101, 111, 113, 116, 132, 133, 149, 150

IPNS Inter-Planetary Name System. 41

JSON JavaScript Object Notation. 32, 70

JSON-LD JavaScript Object Notation for Linking Data. 109

JWT Jason Web Token. 86

LOD Linked Open Data. 24, 149

LoRa Long Range. 4–6, 11, 27–29, 44–48, 50–53, 58, 60, 61, 118, 119, 123, 125–127, 129, 139, 146–148

LoRaWAN Long Range Wide Area Network. 28, 51, 52, 118, 123, 124, 126

LPWAN Low Power Wide Area Network. 2, 4–6, 11, 28, 29, 44–47, 51, 53, 60, 119, 123, 127, 129, 139, 142, 143, 148

M2M Machine-to-Machine. 120

MPN Maintenance and Payment Notices. 40

MQTT Message Queuing Telemetry Transport. 47, 48, 52, 74–76, 118, 124

NB-IoT Narrow Band Internet of Things. 4, 118

NEC Network Energy Consumption. 29

NFT Non-Fungible Tokens. 22, 41, 66, 104, 105, 111, 114, 116, 133, 134, 136–138, 149

OTAA Over The Air Activation. 124

OWL Ontology Web Language. 23, 33, 104–106, 138

P2P Peer-to-Peer. 7, 8, 17, 18, 37, 63, 65–67, 69, 83, 103, 112, 115, 132

PER Packet Error Rate. 28

PoA Proof of Authority. 19, 22, 68, 120, 121

PoC Proof of Capacity. 68, 129

PoS Prove of Stake. 19, 67–69, 101, 140, 150

PoW Proof of Work. 19, 22, 67–69, 71, 74, 121, 129, 140, 150

Pub-Sub Publish-Subscribe. 74

QoS Quality of Service. 28

RDF Resource Description Framework. 13, 23, 24, 32, 33, 104–106, 109, 111, 113, 116, 124, 126, 138, 148, 149

RPC Remote Procedure Call. 70

RSSI Received Signal Strength Indicator. 28

SC Smart Cities. 2–14, 17, 24, 25, 27–31, 38, 43–46, 50, 53, 60, 62–71, 74, 78–84, 86, 95, 96, 101, 102, 111–115, 118–123, 125, 127–130, 132, 134–137, 139, 140, 144, 146–149, 151

SDK Software Development Kit. 38

SF Scatter Factor. 118

SLA Service Level Agreement. 65

SNR Signal-to-Noise Ratio. 28

SPOF Single Point of Failure. 121

SSN Semantic Sensor Network. 10, 113, 148

UML Unified Modeling Language. 31, 32

URI Universal Resource Identifier. 13, 23, 33, 105, 109, 138

USB Universal Serial Bus. 50

UUID Universally Unique IDentifier. 47

W3C World Wide Web Consortium. 23

WoT Web of Things. 9, 113

WPAN Wireless Personal Area Networks. 5, 6, 11, 28, 29, 44–47, 51, 60, 148

XML EXtensible Markup Language. 109

Part I

Initial Considerations

Chapter 1

Introduction

Internet of Things (IoT) devices allow objects to continuously produce all types of data. This information that we made throughout each day will soon exceed the number of findings on the WEB. This long-awaited volume of data will motivate the emergence of a new generation of applications that will use data from our daily lives to generate knowledge and change how we interact with the environments in which we live [3].

This problem leads us to a continuous need to search for new solutions capable of supporting the long-awaited mass adoption of IoT devices, mainly promoted by the arrival of technologies developed for the IoT world, such as 5G [4, 5].

To App extract data from an urban area, for example, using IoT devices are highlighted, as they can receive data from the most diverse sources and environments with simplicity and flexibility [6], [7].

Smart Cities (SC) are a fertile field for these applications IoT, where this urbanized space is used to obtain management information, generate insights, and prevent urban problems.

In a vast application landscape of an SC, we have, for example, urban areas, places where it is expensive to deploy, or there is no energy or network infrastructure. These locations are often inaccessible, and it is hard to reach them often for a simple battery change. We can include water and sewage pipes monitoring, trash cans, fire sensors inside a forest, temperature, humidity, pressure sensors deployed on buildings or power towers, train track sensors, river and rain flood alerts, mountain rockslide control, and snow avalanche sensors. The Low Power Wide Area Network (LPWAN) networks have adherence to these applications because the technologies used have characteristics of access to devices IoT located at a long distance, communication without sight, and low energy consumption [8] [9].

The processing and bandwidth constraints, SC IoT Apps are required to use some architectures with disruptive security and network features. One such architecture is Fog Computing, which makes the decisions and processing activities in a network

point close to the edge node and close to the backbone in contact with the Cloud, that is, the gateways. This network architecture is a potential architecture in a scenario where the nodes at the network's edge do not have enough computational capacity or throughput at the edges. It can be used as a concentration point and data filter of devices, in addition to security rules [10].

SC management Applications that receive data from IoT in the cloud cannot always rely on edge devices. These may not have been provided or installed by the application managers, may be out of standard, and have outdated or modified firmware. These devices do not always have authentication and validation of data sent to API or use traditional centralized strategies using JSON Web Token (JWT). These applications with these characteristics can benefit from security gateways API in the Fog Computing architecture. Validate data, device, and firmware before sending to a management API sc, using a decentralized security manager like Blockchain without the need to trust centralized management of the organism.

Data stored in management API generally depends on proprietary and non-standard protocol, extracting data and linking between systems complex to deploy. Structured extraction models and linking between datasets, such as Semantic Web, can provide a standard query, access, and external link to data in a standardized way.

The presence of these communication, security, and data access standardization capabilities could benefit in a possible future massive adoption of IoT deployed in a SC [11].

Therefore, some new propositions can be used by SC IoT Apps classes that have these requirements:

- Network infrastructure (cost, accessibility, long-range coverage, low power consumption);
- Security (trust in the data, even unknown and legacy devices, guarantee of data origin);
- Standard Data access (Integration with other databases).

Our problem was to investigate the architectures and composition of technologies in order to propose solutions that meet this set of requirements.

Disruptive technologies such as Low Power networks and Blockchain networks have potential characteristics that give flexibility and security to awaited new SC IoT Apps.

Using fully Low Power networks makes possible an SC network with coverage and independence from external stakeholders as mobile companies, using gateways

in Fog Computing to provide contact with points of high-speed networks in a hybrid way.

Cryptocurrency technologies like Blockchain can be used as a background for the security and identity of IoT devices in an SC, using Fog Computing to interact with the off-chain world. Semantic Web can be used to integrate and consume and link data generated by SC IoT Apps and stored on Blockchain in a standardized way and as a graph.

The possibility of Blockchain and Smart Contract transactions in SC scenario provides the possibility of having a decentralized infrastructure uncoupled with a traditional network. Smart Contract is an efficient process of routine automation between IoT devices, eliminating human intervention in dangerous or unacceptable actions, improving this security, and using the traceability and reliability of Blockchain.

This work show some results achieved in SC scenarios with new paradigms in Blockchain Network Chapter 7 and 8, and Fog and Edge Computing's with Low Power network paradigms in Chapter 5.

1.0.1 Network infrastructure

SC IoT Apps deployed in regions without adequate coverage of the mobile operators' network, such as the interiors of forests, deserts, and points on the high seas, are potential users of Low Power Wide Area Network (LPWAN) to become eligible for transmission of IoT data .

Therefore, these Apps SC IoT have as characteristics a hard-to-reach and long-distance communication. It demands to be quickly viable and affordable, for example, a wireless network infrastructure that communicates over long distances with low power consumption to avoid frequent battery changes, preferably independent of mobile network infrastructure and competitive installation cost.

We have mobile network operators, satellite operators, and Wi-FI networks when evaluating available long-range network options. Technologies like 4G for example have prohibitive power consumption for SC IoT Apps located in hard-to-reach places [12] without wall power. The 5G and Narrow Band Internet of Things (NB-IoT) have the expected network coverage and power consumption requirements. However, infrastructure costs are high compared to Wi-Fi and Long Range (LoRa), depending on stakeholders, such as mobile phone companies being proprietary networks. In some countries, the fees charged for their use can be prohibitive to maintain scenarios with many devices.

A low-cost network infrastructure option using unlicensed frequency is Wi-Fi. However, it has high energy consumption, making its use unfeasible in difficult-to-

access scenarios such as avalanche monitoring, is also restricted to SC IoT Apps with wall power or an easy battery change.

Traditional SC network architectures generally use gateways connected to fiber optic backhaul or high-throughput wireless connections. These gateways in networks with IoT applications are intermediaries for security and integration with cloud applications. The expected scenario in most SC applications is the hybrid scenario in which edge applications use different communication technologies and a gateway to deliver data to the data management API. However, in scenarios when the backhaul is unavailable, or the cost of deployment or energy is prohibitive, low-power, long-range network deployments gain relevance.

These low-power networks typically follow the Low Power Wide Area Network (LPWAN) and Wireless Personal Area Networks (WPAN) architectures. The LPWAN [13] are networks that aim to connect devices powered mostly by small batteries and transmit long-range messages with low power consumption. One of these recent LPWAN technologies is Long Range (LoRa) [14], which promises in its specifications links of up to 45 km, allowing transmission rates between 0.3 and 50 kbps in non-licensed. These networks meet the needs of scenarios that send short and preferential messages with a frequency of a few times a day SC IoT.

In an ideal Long Range Low Power architecture scenario SC IoT we would only need an interface LPWAN like LoRa on the device, but this is still expanding. Most currently manufactured IoT devices do not yet have an LPWAN interface as standard, but ethernet, Wi-fi and WPAN interface as Bluetooth Low Energy (BLE).

Therefore, a mix of LPWAN LoRa and WPAN BLE network could be a SC communication solution to compose this Low Power network end to end to the backhaul allowing. Its network characteristics are an investment with affordable infrastructure costs, with independence from a mobile network operator for long-distance communication to the network's edge.

The independence of this network is due to the use of unlicensed frequency, open protocol standards, and technical features of LoRa and BLE.

Some SC IoT applications may require business logic or decisions close to IoT devices at the network edge and have a low tolerance for communication delays with external servers or real-time execution.

As in the LoRa and BLE network mix, the transmission rates are low compared to 4G, 5G, and Wi-fi networks, architectures such as Fog Computing and Edge Computing [15] can contribute to reducing the limits of this mix of networks with Low Power characteristics. Fog computing and Edge computing are network architecture paradigms that aim to allow processes or services to be managed closer to the network's edge. It reduces the amount of data transmitted to the cloud, brings intelligence closer to the edge application, and improves the efficiency of the network

used by the IoT. This Apps in this architecture send only consolidated data to the cloud and have data transport efficiency even with low transfer rates.

To study this potential SC scenario, we investigate a low-energy architecture independent of the mobile network infrastructure, allowing the use of SC IoT Apps, using LPWAN and WPAN. The LoRa and BLE are respectively, LPWAN and WPAN technologies with use unlicensed frequencies. With a mix of these technologies, the rise of new applications transmits long-range data (up to 45 km) with low power consumption, manufacturing cost, and simplified implementation as characteristics. We implemented a testbed mixing LoRa-BLE and simulated LoRa to evaluate its scalability in scenarios of huge numbers of nodes as the SC.

The results of our investigation of this problem can serve as a motivator and provoke reflections on the use of new network paradigms for use cases that do not require frequent data transmission or cloud streaming, in which case sending only consolidated data is required. This information is enough for management and decision information. Use cases such as managing water and sewage networks and sliding rocks and snow on a mountain that have devices deployed in hard access places would not need to send status to each reading variation but an already consolidated set of relevant information and decisions taken at the edge.

1.0.2 Security

Some public services require the identification of a citizen, such as passes to use public hospitals or transport, had a significant percentage of data characterized as sensitive for dealing with data from citizens and public bodies, making data security and transparency a prerequisite [16].

However, these public services, by including IoT for interaction at the edges of the application, can exchange data in an outdated and insecure way, working with devices already with outdated firmware and with API exposed in the cloud in a centralized way exposed to cyber attack as DDoS [17].

Extracting reliable data from unknown or outdated devices and sending them to external API requires new propositions that mitigate the risk of fraud. It can provide security to this scenario, especially considering the expected mass adoption of IoT devices in a universe of applications SC [18][19].

The expected volume of accesses and transactions provided by these new IoT applications in SC scenarios that have dimensions and the number of users in a population can make centralized network models inefficient. It can be limited to handling many simultaneous transactions, need robust data redundancy infrastructure, and should be a complex and efficient security feature to be one point of concentrated cyber attack.

Technologies such as Blockchain have made it possible to propose decentralized infrastructure solutions without needing a trustable or reliable central intermediary. It uses the Peer-to-Peer (P2P) network with a data structure in chained data blocks and signed transactions with strong cryptography that allows data to be stored and consulted with reliability, scalability, and immutability [20].

Some Blockchain features, such as Smart Contracts, enable automation, improving security by eliminating human intervention in unacceptable actions, combined with Blockchain traceability and reliability. Smart Contracts can be powerful in the elaboration of immutable routines. An example of a Smart Contract in IoT is a called maintenance action from of maintenance with defect, such as a sewer density reader, for example. When a defect is identified, a call to a Smart Contract triggers a maintenance company and, if necessary, could generate credits in virtual currencies for the service provider. By having the characteristic of immutability and distribution and transparency routines that generate rewards and payments can be handled automatically by these Smart Contracts in a safe way as well as already operational in cryptocurrency networks.

We propose to use Blockchain and Smart Contract as a security manager background in SC IoT Apps network infrastructure can provide most of the necessary security requirements, as mitigates some security and privacy issues sending data to external cloud data management SC API.

An Open Source Blockchain is Ethereum, one of the leading cryptocurrencies in popularity and trading volume. However, it is a powerfull and disruptive development platform for developing decentralized applications Decentralized Application (DApp) using Smart Contracts [21]. Ethreum can be used as private or public Blockchain with a relevant community of developers; many open source projects, libraries, and tools. These projects aimed at decentralized development promise to change the next generation of applications paradigm in what the community calls Web 3.0 [22][23].

We chose Ethereum as a decentralized security authenticator manager of IoT devices in Fog Computing architecture paradigm. For this, we develop API gateways that call Smart Contracts deployed in Ethereum the rules to verify the originality of the IoT device, validating its firmware, payload, and metadata such as installation location, device owner, serial number numbers, etc. One metadata is the endpoint of the data's external API destination. This authorization and authetication stategy using Blockchain and Smart Contract set to mitigate some fraud risks while sending information from unknown devices setup in a SC to its data management repository in the cloud.

The API gateways in Fog Computing compose a hybrid architecture with centralized and decentralized features. The central components of the architecture are

our daemon designed to be an API gateway in the Fog Computing Network, the API data management SC in the Cloud. And if it is installed in the cloud, the web application responsible for registering the characteristics of IoT would be in the context of centralization. To bring scalability and availability of API gateways, because they are stateless, it is possible to implement them in architectures providing autoscaling. The decentralized components are linked to the Ethreum Blockchain which already has this native feature.

The calls to Smart Contracts are made by API gateways for data verification done through an Ethereum transaction. The Smart Contracts are Turing complete immutable programs that, using the security features of Blockchain, are too special accounts accessed by a network address that can send transactions and have a balance of cryptocurrencies. Smart Contracts run without no infrastructure dependencies, being resilient and security-enhanced. They are implemented in a Block by a transaction. The Smart Contracts are written in a language such the Solidity, compiled, and deployed in Ethereum. The address of deployed Smart Contract is used by the DApp for interaction using its Application binary interface (ABI) interface Blocks are the fundamental entity of Ethereum Blockchain, responsible for being the repository of transactions. Its structure is a chain of blocks linked by the previous block's hash. Without a central trusted intermediary authority, all transactions are signed and replicated between network nodes in a P2P architecture. All nodes share this same data structure, and any changes are public to all nodes, with changes validated by a consensus mechanism. Moreover, any change to any of the previous blocks in the block network violates the consensus rules and invalidates the entire chain. The primary control is done by accounts that use their cryptocurrency balance to send transactions on the network. Transactions are signed messages sent by this account responsible for changing the state of the Blockchain, using its private key to sign the transaction. Once validated, transactions wait to be validated by the Ethereum Blockchain network until they are permanently included in a block. The transactions can be verified anywhere, or anybody is possible because it is stored in the block to maintain an immutable history.

The web application responsible for the registry is a DApp with stateless feature and its deployment can even be thought of in decentralized servers such as IPFS. However, we did not get to test this possibility, leaving this task for the future as a search for an architecture with more decentralized components.

Applications that currently generating IoT data and send to API on the cloud. Could use our proposal of Authentication and Authorization using Blockchain to verify data authenticity by signature, firmware, and data origin in gateways in Fog Computing before allocating the same API in the cloud. In the example of sewage control, the devices before sending the sensing information on the density and tem-

perature of the waste to the management API. Add your ID, message signature, the hash of your firmware, and HTTP endpoint of the API in the cloud to the payload. This information is validated in a API gateway on the Fog and, if validated, forwarded to a protected API in the cloud. This case addresses a hybrid use of centralized and decentralized strategies in applications, including legacy applications, partially mitigating the risk of having outdated and fraudulent firmware, in addition to fraudulent sending of information from the edge of the network.

We can enumerate some limitations of our proposition and architecture. The IoT devices must support an operating system capable of installing node.js and the web3.js libraries responsible for interacting with Ethereum, such as a RaspberryPi. Our Edge Gateway API IoT executes signatures on the payloads it receives before submitting it to another API Gateway, which takes a certain amount of processing power.

1.0.3 Standard Data Access

Many of the IoT devices already found in the industry have protocols, Application Programming Interface (API), proprietary infrastructure, and there is no standard or consensus among manufacturers on the way to expose their data for consumption. The exchange of data without a standard makes it complex to manage and link the data generated by these devices in large volumes.

Born as a proposal to standardize the query, link, and use by computers of Web data, the Semantic Web allows the creation of standardized data sets on the Web using ontologies and rules for the interoperability of these data. It can be a proposal for standardizing access to data produced by the increasing number of IoT devices found in scenarios like SC. The Semantic Web applied to the IoT context can help link device data using an ontology of sensors to be queried and integrated with other Web databases, enabling new insights for applications [24] [25].

Ontologies provide data access abstractions, which allow data to be searched as a graph to be accessed by a query language such as a database in calls from languages such as SPARQL. A IoT data expose using Semantic Web have external dataset link, data extraction, filtering, and aggregating features in a standardized way; this approach is called Web of Things (WoT)[26].

Blockchain is used in this work as a management and security tool for SC IoT App devices. Consuming data from its structure using standardized proposals such as the Semantic Web allows new opportunities to use this data produced by SC IoT Apps. It brings new possibilities for integrating information from transactions, blocks, accounts, and Smart Contracts with other web databases, creating new opportunities and insights.

In a scenario of IoT devices in waste management applications submitting transactions on the Blockchain to change their state in the pipes. The transactions submitted in Blockchain using Smart Contract generate access history when stored on the Blockchain in log format. It allows queries and links with other web datasets, asking questions by writing SPARQL sentences using ontologies made for sensors and Blockchain.

The possibility of querying data interconnected by SPARQL makes it possible to use external datasets for new and consolidated insights. An example in our context of SC would be waste management applications being able to cross-reference density data in water with external weather datasets exposed and annotated using ontologies Semantic Sensor Network (SSN) [27]. The possibility of having historical data of ambient temperature and rainfall linked with IoT consolidated data, for example, could enrich the analysis of density and increase of residues in repositories and pipes.

Applications using Blockchains with a Smart Contract and an ecosystem of decentralized projects already in the process of rapid popularization, such as Ethereum. Currently, applications and interfaces API of devices and applications SC do not have a standard query and data integration, making relevant new ontologies propositions and standardized ways of exposing its data for consultation and integration. This approach of providing an ontology for Ethereum and a middleware that exposes its endpoints was explored in this research and is a contraposition of the most current proprietary and isolated API SC and IoT.

A popular Decentralized Application (DApp) has emerged to access real-world services and applications outside the decentralized Blockchain ecosystem that use Smart Contracts to provide off-chain interaction with external real-world services and is called upon by the Oracles community. Considering the future demand for data consumption from a Blockchain, we can consider these Oracles one of the potential entities to consume and consult external data linked to Ethereum data to generate richer insights intensively [28].

Consuming Ethereum data gains relevance considering the growth of decentralized projects and decentralized organizations. The oracles responsible for interacting with the off-chain world use Ethereum data to interact with the real world. SC IoT Apps that use Ethereum as a background for security management or data and logs repository may need to consume this data in a standardized way, enabling the linking of this data with other datasets to enrich queries and generate new business insights.

In order to expose the data in a standardized way, we propose an extension to the EthOn ontology of the literature that represents the entities of the Ethereum data model, EthExtras. EthExtras adds some objects that expose properties in a complementary way to transactions, receipts, blocks, and accounts. EthExtras also

inserts the external relationship with the DBpedia dataset as an example of a link between datasets.

Using EthOn and EthExtras, we developed a middleware that exposes data from Ethereum networks in soft real-time. Entities such as Blocks, transactions, and accounts are exposed in web URI endpoints as a graph in RDF format, making it possible to query their triples through tools such as SPARQL.

1.1 Contributions

We propose the IoT LowPower and Blockchain scenarios apply Fog / Edge computing paradigm, Chapter 5, autentichate and authorize IoT devices using Blockchain and Smart Contract as background, Chapter 7, and using web semantic to extract data of Blockchain data using new ontologies, Chapter 8.

1.1.1 A Fog/Edge Computing LowPower Network

This work proposes a low-energy SC network architecture, with competitive cost of deployment, using unlicensed frequency. In this architecture, the IoT devices until the backhaul network use high rates, in an alternative of mobile network infrastructure. To this, we mix the Low Power networks LoRa an LPWAN and BLE a WPAN in a Fog/Edge computing.

Using a combination of network paradigms like Fog Computing, LPWAN networks can help in scenarios that require streaming data but don't have throughput from the edge of the network to the backhaul. In this paradigm, only consolidated data are sent, sufficient for managerial and decision-making information. Rock and snow slide statuses on a mountain with devices deployed in hard-to-reach places would not need to send status for each reading variation but rather an already consolidated set of relevant information and decisions made at the edge.

We implemented test-beds using LoRa and BLE interfaces with message exchange in a Fog / Edge Computing architecture.

To evaluate the capacity of the LoRa network, we simulate an SC IoT network scenario of thousands of devices using LoRaSim, sending messages in different periods. To verify the range of the LoRa specification, we up links in a metropolitan area, where the results obtained with a development kit were satisfactory, showing that LoRa presents an ideal potential for network coverage of a city.

This Long Range approach is suitable for use cases of SC IoT Apps where the devices are located in remote locations from the base and difficult to access due to difficult access. We can include water and sewage pipes monitoring, trash cans, fire sensors inside a forest, temperature, humidity, and pressure sensors deployed on

buildings or power towers, train track sensors, river and rain flood alerts, mountain rockslide control, and snow avalanche sensors.

And the feature combined with BLE networks allows Low Power networks to use devices already found on the market.

This contribution is published in [29].

1.1.2 IoT authorization and identification using BlockChain

IoT devices installed in urban areas, such as pipes, power sources, sewage, and trash cans, the temperature can be installed by unknown stakeholders and with hardware often from different and unknown manufacturers. In this scenario, the prior knowledge and registration of a device, its metadata, its firmware, and the data destination API address can increase security and confidence in data coming from the edges.

In this scenario, the security features provided are often ineffective. For these applications, our proposal presents an additional layer of security to receive messages from IoT devices more reliably using as a security background infrastructure in decentralized paradigms such as Blockchain.

We presenting API gateways **IoT Edge API Gateway** and **Blockchain API Gateway**. These API gateways run in Edge and Fog Computing, and they sign and verify the IoT message's authenticity, using Smart Contracts deployed in Blockchain.

The **Blockchain API Gateway** call the Smart Contracts of the project IoT Device Management [30] a DApp. We deploy a testbed using real devices running the **IoT Edge API Gateway** to validate messages before sending them to a server running the project IoT Framework Engine [31]. This objective of the testbed is to represent a typical SC IoT App scenario.

How to contribute, we can list:

- The **IoT Edge API Gateway** project is a daemon running on an IoT device, which is responsible for assuming two sensors receiving messages and preparing a payload containing assurance and metadata that are used to aid in the verification of authenticity;

- the **Blockchain API Gateway** project is a daemon running in bastion that protects the application network. The daemon receives the **IoT Edge API Gateway** payloads. If the sender's authenticity is verified, the message follows the protected application network's data management server address;

- a discussion about Blockchain and Smart Contract in SC Apps; and,

- a testbed using real devices to produce IoT data and send to a SC API as proof of concept.

This contribution is published in [32].

1.1.3 Extract Blockchain data using Semantic Web

As a contribution, we direct efforts in applying ontologies and semantic Web techniques to expose Ethereum data as a graph. This model would enable standardized use of Ethereum data in current and new decentralized applications.

This alternative can solve the integration needs between pre-existing datasets on the Web and Blockchain, allowing SPARQL queries on the linked datasets. For example, this feature can be helpful to recent applications that interact with Blockchain and the world off-chain, the Oracles.

Oracles services provide guaranteed and secure communication of Smart Contracts with the off-chain world. Its services propose to query, verify and authenticate external data sources, potential beneficiaries of a Semantic Web-based dataset integration. An example of calling an Oracle service to help an SC application would be the query to external data of services used in an urban environment, such as product prices, credit information on a financial institution, and payment verification on credit cards, among others. In other words, a dataset modeled as the data exposed by our Midelware using Semantic Web tools can be the potential to help this bridge between the Blockchain and Smart Contract world and the outside world. Blockchains do not access data from outside the network (off-chain), Our middleware is not designed specifically for IoT SC Apps data consumption and can receive new classes added by EthExtras to be applied to any Apps class that uses Ethereum and Smart Contract as a base, exposing their logs as URI to be consulted by SPARQL

Our contributions to Web Semantic research are:

- The EthExtras, a new ontology that adds components to EthOn, add auxiliary classes that extends some relationship between Ethereum entities.
- A Web application that uses EthOn and EthExtras ontologies extracts data from Ethereum in production and convert it into RDF, making these entities available as Universal Resource Identifier (URI) for visualization and query.

This contribution is published in [33].

1.1.4 General Contribution

Our general contribution proposal shows that the Fog Computing architecture is a powerful option when it is applied to scenarios of devices with low computational

capacity and multiple characteristics. When we apply it to the context of SC, we see the need to apply advanced security in a scenario dominated by the diversity of technologies. We seek to investigate a use case in Blockchain that already has security features in its design; it can be a path to be followed in applications where it is not always possible to trust or know the edge.

This work made possible the following publications,

[29] **Low-Energy Smart Cities Network with LoRa and Bluetooth**, 7th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2019.

[34] **Blockchain for Machine to Machine Interaction in Industry 4.0**, Blockchain Technology for Industry 4.0: Secure, Decentralized, Distributed and Trusted Industry Environment, 2020.

[32] **IoT Registration and Authentication in Smart City Applications with Blockchain**, Sensors, 2021

[33], **A middleware for systems consumes Ethereum data in soft real-time: a Semantic Web approach**, SBESC, 2021

1.2 Structure of the thesis

The rest of the text is structured as follows

Part II, the Theoretical References;

– Chapter 2, summarize the main technologies and concepts;

– Chapter 3, show the related works;

– Chapter 4, presents applications, issues and challenges of using Blockchain, Smart Contract and IoT;

Part III, the Case Studies;

– Chapter 5, addresses our Low Power Fog Computing network proposal for Smart Cities (SC);

– Chapter 6, discuss Smart Cities (SC) Scenarios and present a Fog Blockchain and Smart Contract IoT and challenges to adoption;

– Chapter 7, address our Smart Cities (SC) Authentication and Authorization IoT using API Gateways calling Smart Contract in Ethreum Blockchain;

– Chapter 8 addresses our model for consumption, linking, and use of data from Ethereum Blockchain networks in a standardized way using Semantic Web and ontologies;

Part IV, the Final Considerations;

– Chapter 9, present the works of literature that discuss and that reference the main contributions of this work;

– Chapter 10, describes the hypothetical join scenario of our use cases.

– Chapter 11, the discussion and conclusions around the main themes raised and addressed in the research

– Chapter 12, the final conclusions and the future works

Part II

Theoretical Reference

Chapter 2

Tecnologies and Protocols

In this research, Blockchain is the platform for Smart Contracts and data responsible for writing state and identifying and validating IoT devices. To better understand our contributions, this chapter describes the Blockchain components, a base technology used in this research, and Web Semantic main topics. The objective is an overview of Blockchain-related characteristics, technologies, and approaches.

2.1 The Blockchain entities and concept

Blockchain is a distributed ledger being all transactions are signed and replicated between network nodes in a P2P architecture. The network does not need a central trusted intermediary authority, and its structure is a chain of blocks linked by the previous block's hash. Nodes share this data structure, and any changes are public to all nodes, with changes validated by a consensus mechanism. Figure 2.1.

It is a recent technology and has not yet been tested in all scenarios. It is commonly found as a solution in financial services and cryptocurrencies. Its decentralized feature makes cok that features the scalability and immutability required by many modern applications such as SC IoT Apps.

To write routines and programs in Blockchain, we have Smart Contracts, a concept introduced by Nick Szabo in 1994 [35]. These Smart Contracts are the basis of decentralized programs commonly called Decentralized Application (DApp) and have no infrastructure dependencies, being resilient and security-enhanced. They are implemented in a Block by a transaction and called by an interface Application binary interface (ABI) located in a Blockchain address, making it possible to build Turing complete routines and immutable, using the security features of Blockchain [36].

The block, account, and transactions are the main Blockchain entities in this section and are referenced during our research and used to model Ethereum Blockchain's ontologies to soft-real-time data exposure middleware.

Blocks

Blocks are the fundamental entity of Blockchain, responsible for being the repository of transactions. It has the previous block's hash, which is used to chain. The hashes are calculated based on the block data, and any change to any of the previous blocks in the block network violates the consensus rules and invalidates the entire chain.

The first block is called Genesis Block, and to add a new it is necessary to mine, being a consensus algorithm responsible for verifying the trust of this newly mined block and approving a new block in the chain.

Accounts

The primary control is done by a user or Smart Contract. A Blockchain account is an entity that, in non-permissioned networks such as Ethereum, uses its cryptocurrency balance to send transactions on the network. Smart Contracts are special accounts accessed by a network address that can send transactions and have a balance of cryptocurrencies.

When a user account is created, a private key consisting of 64 hexadecimal characters is generated, and the public key is calculated from this private key with Elliptic Curve Digital Signature Algorithm (ECDSA). The account's public address is the last 20 bytes of the public key hash, adding 0x to the beginning.

Transactions

Transactions are signed messages sent by a user or Smart Contract and are responsible for changing the state of the Blockchain. The user needs the private key to sign the message and a destination public account address for a valid transaction.

Once validated, transactions wait to be confirmed by the Blockchain network in a temporary structure called a mem-pool until they are permanently included in a block. The transaction can be verified by anywhere or anybody is possible, because it is stored in the block to maintain an immutable history. This security feature is the foundation of Blockchain's robustness, allowing reliable, auditable data storage without being altered or deleted, decreasing the possibility of transaction fraud.

2.1.1 Consensus Algorithms

Blockchain has a distributed and decentralized network paradigm with P2P nodes of a highly scalable nature. The cryptography base uses an elliptic curve that allows the nodes to store the transaction in blocks without trust. Strong security is based on the consensus algorithm, which is often computationally expensive.

In the consensus algorithm Proof of Work (PoW), the miner's nodes make excessive computational use to find a solution and create a new block, which is an algorithm used by Bitcoin. The solution after PoW must satisfy criteria, which the other network nodes can easily verify. The block is mined and validated by all nodes of the network and addiction to the chain.

However, nowadays, the PoW is a problem; PoW miners make much computational effort to a resolution, being an inefficient, slow, and highly energetic consensus process. An energy-saving alternative is Prove of Stake (PoS). Instead of requiring users to find a nonce by brute force, the PoS requires people to prove ownership of the amount of money. The justification for this algorithm is that persons with more coins are less likely to attack the network.

Some critics of the PoS consider this selection based on the balance of the account quite unfair because it would give dominion of the net to the rich group of people. Compared to PoW, PoS has better energy efficiency by having many Blockchains adopting PoW and migrating to PoS gradually as the Ethereum.

Another approach is the Proof of Authority (PoA), whose transactions and Blocks are validated by authorized accounts, known as validators. Validators are responsible for inserting transactions into blocks. In the PoA, people earn the right to become validators and are encouraged to maintain their position, with a reputation given to validator identities. They are encouraged to maintain trust in the transaction process to prevent their identities from being associated with a security incident leading to a negative reputation. PoA is considered more robust than PoS. Blocks and transactions are verified by pre-approved participants, who act as system moderators.

2.1.2 Merkle Tree

Merkle trees are used in Blockchain networks to validate the data integrity of a transaction in the blocks. In our research proposal of using an API gateway for authentication and validation of payloads of an IoT, during a IoT registration, a Merkle root using its metadata is calculated and registered in the Blockchain for future validation.

Merkle trees are used to verify a Blockchain's content and data consistency, where transactions are used to generate a binary tree where each leaf node is the transaction hash, and the value of each non-leaf node is the hash of the transactions below it.

Merkle proofs verify if a transaction belongs to the tree. It uses a tree root hash and a "tree branch" with all hashes along the path from the leaf to the root. It is possible to verify that the resulting hash for that branch is consistent throughout the

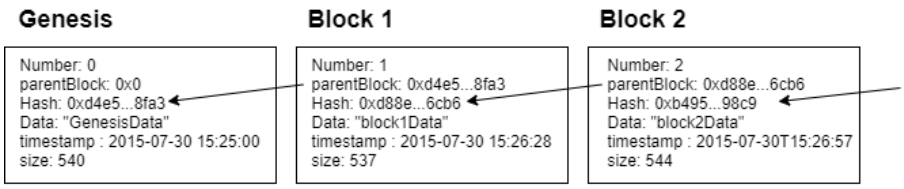


Figure 2.1: Blockchain Blocks

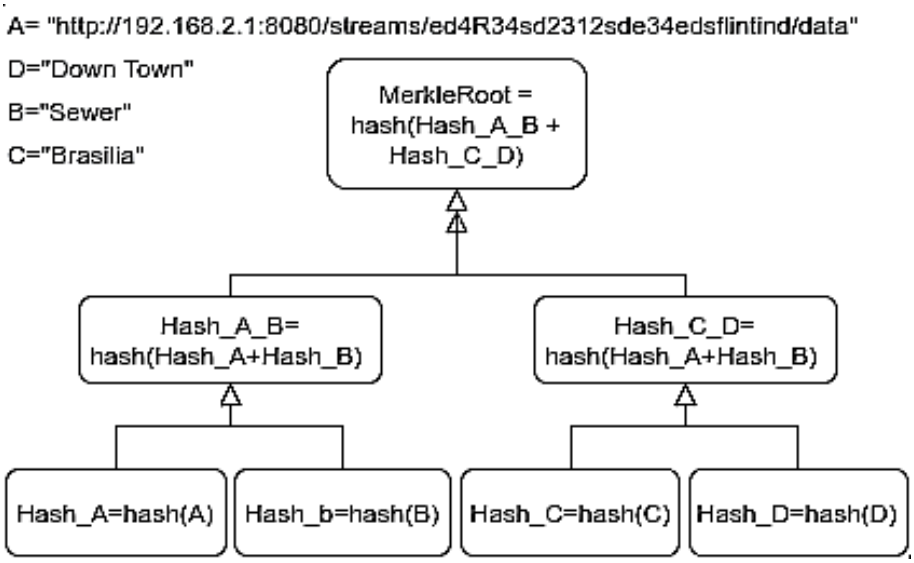


Figure 2.2: Mekle Tree

path by verifying that the data belongs and is in that position in the tree. Someone who needs to prove that a piece of data is in the tree does not need the entire tree.

An individual transaction on a Blockchain Block can be verified using Merkle Tree, a data structure that efficiently stores key-value pairs. This data structure is a complete binary tree of hashes. Each leaf node is the hash of an object, a transaction in the case of Blockchain. The value of each non-leaf node in the tree is the hash of the two nodes below it, continuing until the total number of hashes remaining is just one, the Merkle root, Figure 2.2.

The proofs of a Merkle tree are used to decide if a piece of data belongs to the tree and is consistent with a set of data without revealing it. For example, to verify data [C] in Merkle Root, we use the function hash [C], which results in Hash_C. Furthermore, to validate whether C belongs to the Merkle tree, it is unnecessary to reveal it. Hash_A_B is the Hash_A when hashed with Hash_B, Hash_C_D is the Hash_C when hashed with the hash of unknown D, Hash_D. Hash_C_D hashed with Hash_A_B result in the Merkle Root Hash_A_B_C_D.

We use Hash_D, Hash_C_D, and Hash_A_B without revealing C or any data to prove that the C data is present in the Merkle tree.

With these, we can obtain Hash_A_B_C_D, therefore proving that Hash_C was part of the Merkle tree, which implied that data C was, in fact, part of the universal data set [A, B, C, D]. The hash function generally uses SHA-2, although others can also be used. Ethereum uses Keccak-256 belonging to the SHA-3 family of hash functions.

2.1.3 Ethereum

The open-source Ethereum is the Blockchain network we use in this research. It has built-in attributes for deploying Smart Contracts. The contracts in this work deployed in Ethereum are used for the registry and validation of an IoT device. We use access to its data to extract soft-real time Blockchain data as a graph using Web Semantic.

Ethereum is currently the leading DApp development platform and the second most popular cryptocurrency, losing in financial transaction volume only to Bitcoin, the precursor to the Blockchain [37]. The Ethereum project is an Open Source Blockchain created to be the global decentralized computing platform for executing Smart Contracts programs. It uses Blockchain sync mechanisms to manage state changes, using its default cryptocurrency, the Ether (ETH).

Ethereum has become a popular platform for Blockchain applications, providing more features than Bitcoin because it includes the capacity to run programs or Smart Contracts; it significantly contributes to generating new application possibilities [37].

Ethereum follows the revolutionary idea of the internet as a free and collaborative network, in contrast to the current network increasingly controlled and regulated by a few centralized organizations. Furthermore, it is in this expectation of being the new internet for a new pool of totally decentralized applications, independent of the current internet. All this shows the relevance of new research on Ethereum that just as the Semantic Web is considered Web 3.0, and its client libraries are named like that, like Javascript, web3.js

In Ethereum, each Smart Contract has an address, and for a new transaction, submit is necessary to use its address and its ABI interface definition. The transaction is sent, and after consensus and block validation, the Smart Contract is executed in a secure environment, the Ethereum Virtual Machine (EVM).

Smart Contracts on Ethereum are Turing Complete language, written and compiled in languages like Solidity, stored as bytecode in a block, and have its execution in an EVM. In Ethereum, all transactions have rates measured in an internal unit called Gas. Thus, each Ethereum transaction must specify a maximum Gas fee limit used during a Smart Contract routine execution. This mechanism ensures some control over the costs of the routine run of a particular call to a Smart Contract, in

addition to preventing it from entering, for example, an infinite loop and executing it consistently [38].

Ethereum Request for Comments (ERC)

The Smart Contracts motivate the invention of a new programming language, Solidity, to write on Ethereum. They already include routines of software thinking in security. The ERC provides Smart Contracts standards already tested by the community, such as the ERC-20 and ERC-721, used by the developers to Tokens and Non-Fungible Tokens (NFT) projects. The existence of these Ethereum Request for Comments (ERC) standards motivates a current significant increase in projects of this nature using Ethereum as a basis.

Storage

Store great data quantity in an Ethereum Blockchain is not recommended should limits of architecture, time of a transaction, store limits, and mainly the Gas costs. DApp with needs to store digital media such as documents and digital media could benefit from decentralized storage projects. Some decentralized projects for deploying distributed stores are Interplanetary File System (IPFS) and Swarm [39]. The DApp uses only this file's reference in this platform and stores this in Ethereum.

Name Service in Ethereum

A project resolve names in Ethereum, Ethereum Name Service (ENS), Domain Name System (DNS) corresponding and is responsible for referred Ethereum references in a human-readable name, provides a complete decentralized architecture to DApp.

Public Ethereum Network

The OpenSources project implementations as GoEthereum (GEth) written in Go and Parity, written in Rust, could run a private Ethereum or participate as a node of public Ethereum MainNet or test networks (Ropsten, for example).

The MainNet is the production public Ethereum network, and to call transactions and deploy Smart Contracts in this network is necessary real ETH. The transactions and Smart Contract execution consume Gas, have the cost in ETH, and have real consequences.

To deploy test Smart Contract in public Ethereum network, the community maintains test networks Ethereum. The more popular available are Ropsten, a PoW network, and Kovan and Rinkeby, both PoA networks. It is possible to deploy Smart Contracts to test, and the ETH in these networks do not have anyone value and are distributed by faucets easily found on the internet. A popular service in Ethereum

and DApp ecosystem is Infura [40] is used to connect with public Ethereum networks and IPFS. Infura provides a Blockchain development suite and API and avoids the necessity of a local node setup.

2.2 Semantic Web

Semantic Web is the proposition to simplify Web data consumption, allowing integration and cooperation between man-machine, the Web 3.0. The WEB is the most significant data source, and HTTP is the dominant protocol for transactions and consumption of structured, semi-structured, and unstructured online sources. This effort aims to access WEB contents as a database, where data is accessed by a Universal Resource Identifier (URI) and related, shared, and queried over HTTP.

The Semantic Web is currently the World Wide Web Consortium (W3C) recommendation for integrating WEB data sources. Proposed in 2001 by Tim Berners-Lee, the Semantic Web mainly works on offering data models that can be an extension of the World Wide Web, allowing machines and humans to work together. For example, it links the meaning of words through ontologies.

2.2.1 Ontologies

Ontologies are data models representing a set of concepts within a domain and the relationships related to content published on the Internet.

To model ontologies, a standard format is Ontology Web Language (OWL), a language designed to represent things and relationships. OWL ontologies generally refer to other OWL documents, and their representation can be used by computer programs and humans to understand a set of objects and properties of a domain. The currently recommended format for storage and query data is the Resource Description Framework (RDF), its format representing a graph triple.

2.2.2 Resource Description Framework (RDF)

The RDF triple consists of:

- subjet, expressions that RDF uses to describe the resource
- predicate, a specific description of a resource can be an attribute or a relation between subject and object.
- object, a named property, and its value.

To retrieve information from an RDF Graph, we have SPARQL; this is the semantic query language recommended by W3C; it recovers and manipulates data in RDF format, returning an RDF Graph as a result.

A classic example of a SPARQL query to a web dataset is a search in the dataset represented by the endpoint `http://www.w3.org/People/Berners-Lee/card` [41].

When we submit the SPARQL query, 2.1 using the FOAF ontology returns "Timothy Berners-Lee." FOAF is declared as a prefix, the ontology responsible for describing a person, its activities, and relations.

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
    ?person foaf:name ?name .
}
```

Listing 2.1: SPARQL query

Today we already observe interconnected Semantic Web communities such as the Linked Open Data (LOD)[42], which has over a thousand connected and open-access datasets.

Figure 2.3 illustrates a Graph using EthOn Ontology, our EthExtras ontology proposed in this work, and DBpedia. They present Classes and their properties representing the Ethereum Blockchain, its networks, its Genesis Block, and link as an external DBpedia dataset. DBpedia is a project that uses Web Semantica extracts and exposes structured content from Wikipedia.

2.2.3 IoT and Smart Cities Web Semantic models

This work's contribution has focused on Blockchain data extraction. However, Semantic Web has potential use in specific applications such as SC and IoT to provide a link with pre-existing datasets. Allowing interaction between these data generated by pre-existing devices and sensors and datasets such as those existing in LOD [42] forming a Web of Things (WOT) [26].

In the hypothetical SC case of a standardized data extraction generated by IoT temperature sensors and exposed by ontologies and Semantic Web models. It Provides rich queries such as "what is the average monthly temperature of the central public library region compared to the number of bicycle users at the library station." Queries like this using the database generated by sensors and crossed with volumes of public transport users can bring insights into the influence of temperature on public transport and the impact on public prediction visitors, once linked datasets and available in Semantic Web models and ontology. Compared to central-

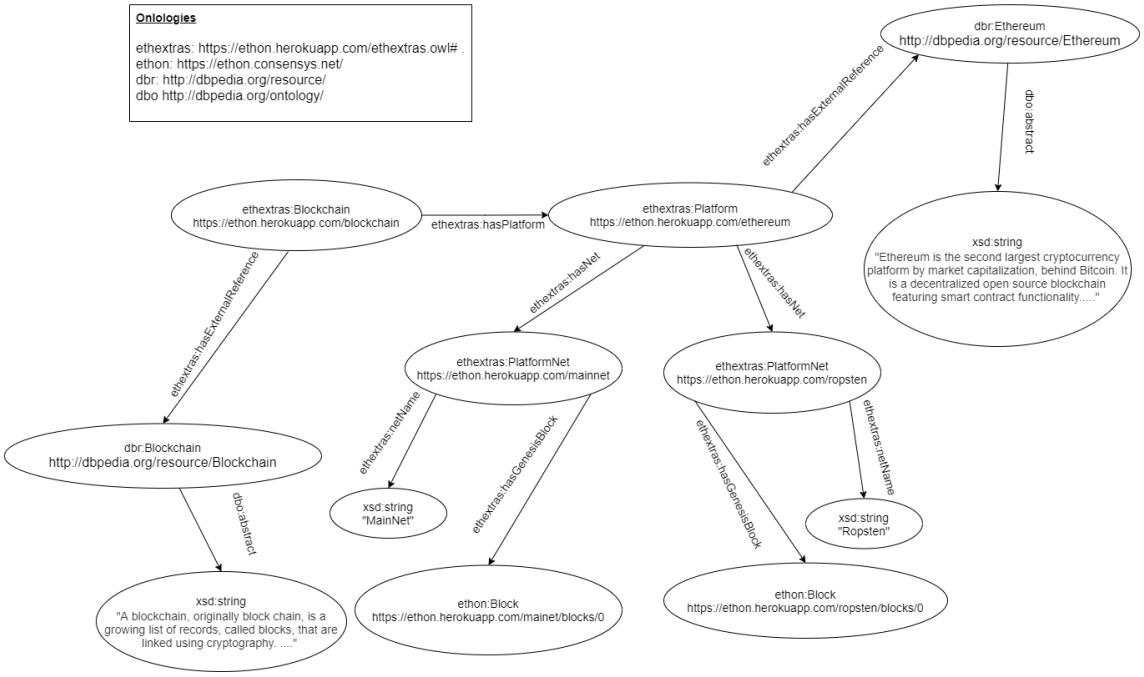


Figure 2.3: A Ethereum description using RDF Graph

ized databases, the maintenance and integration of these by API and the integration of heterogeneous databases of the various use cases and sectors of responsibility of a city may not be so simple.

Some propositions for extracting data from IoT that annotate and expose semantic absorption models can already be found in the literature; one of them is the XGSN. It implements a virtual sensor layer to visualize data using the SSN (Semantic Sensor Network) [43] ontology.

OpenIoT [44] is an example of a framework that annotates these data in graph databases using the RDF (Resource Description Framework) format to extract them and link them with other datasets to enrich the queries.

OpenIoT uses XGSN, which implements the concept of a virtual sensor to interact with the IoT device by abstracting its API.

SSN ontology is an ontology for describing actuators and sensors, covering their observations, procedures, characteristics, and observed properties. as well as the actuators. SSN includes a lightweight but independent core ontology called SOSA (Sensor, Observation, Sample, and Actuator) [45] for its classes and fundamental properties. SSN and SOSA can support various applications and use cases, including satellite imagery, large-scale scientific monitoring, industrial and home infrastructure, and SC.

As SC ontology, we found SCO (Smart City Ontology) in literature [46]. It is composed of building blocks (physical, institutional and digital space), functions (information gathering), learning, collaborative innovation, and information dissem-

ination.), Its superclasses describe the central physical and social elements, digital and functional cities, and urban districts.

2.3 Conclusion

This chapter addressed the concepts and technologies that served as tools for our research and testbeds using Blockchain and Semantic Web. Although several other concepts are discussed, we will deepen them as they appear in the chapters in which we present the use cases.

Chapter 3

Related Works

This chapter shows the main related works as a theoretical reference for our investigations in the field of Smart Cities (SC). We divided into three groups, the works related to Low Power networks, the researches that use Blockchain as background for Internet of Things (IoT) applications, and the works that address the extraction and use of Blockchain data using Semantic Web and ontologies from the literature.

In the literature, research on SC IoT and Blockchain security is limited. Most works talk about Blockchain and Smart Contract technology, their communication and security benefits, and the challenges of providing transactions on embedded IoT hardware. These works mainly focus on providing ownership and identity relationships, authentication and authorization, data governance, and privacy over IoT and Blockchain.

3.1 LowPower Studies

There are few studies involving LPWAN, and most relevant works are still in the simulation field. The Things Network [47] is one of the cases where a real-world implementation is presented but does not provide details on the actual limits of expansion of this network. We relate some of the articles that cover the technologies and concepts related to Low Power in the SC and IoT networks.

Survey [48], works describing and categorized apps related to practical applications using LoRa networks. Work [49] details the LowPower technologies available for IoT applications.

The article addresses the difficulty in collecting data in some city locations and acting in real-time. It emphasizes the need to investigate and incorporate new communication technologies to make data collection, analysis, and decision-making more efficient, enabling better forecasting and planning and impacting the population's quality of life. When analyzing SC, [50], it shows the impacts of the constant interaction of users in the urban environment, using extraction and data collection of

IoT devices offline for consolidation and future manipulation. Our proposition of an LPWAN network LoRa and Wireless Personal Area Networks (WPAN) Bluetooth Low Energy (BLE) makes some of the problems unresolved by this work can be bypassed using LoRa allowing efficient long range data collection.

In [51], are do performance tests on an LPWAN LoRa deployed in the city of Rennes on a protocol stack called LoRa FABIAN, to verify the QoS allowed by the network. The authors use real test-beds for generating and observing the traffic between the nodes IoT and LoRa IoT stations. This work long-range transmission technologies in unlicensed bands, challenging traditional applications using cellular networks as results provide performance information of a LoRa network, in metric of Packet Error Rate (PER), Received Signal Strength Indicator (RSSI) and Signal-to-Noise Ratio (SNR). This study shows the implementation challenges of LPWAN networks using only Lora devices. Our proposition uses Fog Computing and BLE at the network's edge to extend the LPWAN.

In the [9] are analyzed the performance and scalability evaluations of the LoRaWAN protocol. This document compares the technology limits and advantages of using LoRaWAN for indoor IoT communications compared to 5G networks, using LoRa devices in all solutions.

The work [52] studies the scalability of LoRa and show that the probability of coverage drops exponentially as the number of end devices grows due to interference in the scattering sequence.

The paper [13] investigates LPWAN with LoRa technology in health and wellness applications. In this paper, we study the performance of LoRa communication for indoor IoT Apps used to monitor the well-being of one of the researchers in their workplace. As a result, they conclude that LPWAN networks and LoRa technology have shown potential for monitoring patients in hospitals and at home. Our proposition when merging technologies such as Bluetooth and LoRa would allow the architecture proposed by this paper to use widely found BLE healthcare devices.

The work [53] discusses a proposal for an intelligent irrigation system, where IoT irrigation devices send data to the cloud through gateways and LoRa devices. The article shows the proposed system's transmission distance and energy consumption are reliable. Just like in SC applications, merging LoRa and BLE technologies could be helpful in Smart Agriculture use cases, making it possible to use today's most popular Bluetooth devices with more affordable prices.

In the study, [54] the LoRaSim tool based on SimPy, an environment for discrete event simulation based on Python, was used to simulate LoRa networks. They demonstrate the scalability of the LoRa network using simulation across multiple scenarios and the number of connected devices. Its results and parameters are compared and calibrated with a real LoRa network. To evaluate the scalability and

performance of a LoRa network obtained by the simulation results with LoRaSim, the authors defined two metrics, Data Extraction Rate (DER) and Network Energy Consumption (NEC). DER is the proportion of messages received and transmitted over a period, and Network Energy Consumption (NEC) is the amount of energy to extract one message. In our study, we used LoRaSim to verify LPWAN LoRa network scenarios in IoT Apps for SC, and DER is one of our results parameters.

Like our work, some studies use the Fog Computing approach in their IoT application architectures; they show the challenges of processing as much information as possible at the network's edge. This strategy allows sending consolidated data to the cloud, consuming little network bandwidth and increasing intelligence and efficiency to IoT Apps. In [55], the authors use the open source projects TensorFlow, Docker, and Kubernetes, to implement a distributed data analysis platform using the Fog Computing paradigm. The paper [56] present a broker MQTT in Fog Computing to activate and deactivate BLE nodes in WPAN, monitoring trajectory, energy consumption and performance. The work differs from our research because it only acts as the manager of the BLE devices and does not extract data from them.

3.2 Blockchain and IoT Works

Table 3.1: List of IoT and Blockchain related works.

Techniques	Problems Addressed	Contributions
Blockchain [57]	Survey	Concentrate works in information systems
Blockchain and IoT [58]	Database for IoT Apps	A Simple Mechanism to Blockchain based Database
Blockchain, Smart Contract and IoT [59]	Automate complex processes	Identify solutions and workarounds in the combination of Smart Contract, Blockchain and IoT
Blockchain, SC and IoT [60]	Information security and privacy of IoT	a security framework that integrates the blockchain technology with smart devices
Blockchain, SC and IoT [61]	Blockchain-assisted information distribution system for the IoT	Design of the system
Blockchain and Fog Computing [62]	Secures sensitive data with encryption, authentication	Ensure improved security features through Blockchain technology
Blockchain and Fog/Edge Computing [63]	Applications of Blockchain-enabled fog	unveils the working relationship of Blockchain and the fog/edge
Blockchain, IoT and Edge Computing [64]	Cooperation and collaboration of recurses	An incentive-based mechanism to offer a reward for the participant in the process using Blockchain
IoT, Kubernetes and Fog Computing [55]	Analytics applications without sending everything to the data centers	iA analysis platform in the Fog Computing using Kubernetes
Blockchain and IoT [65]	Tracking and revocation of malicious users	Blockchain access control scheme with traceability and revocability in IIoT for smart factories
Fog Computing and IoT [56]	Availability of application-layer	MQTT-driven IoT-Fog integration
SC, IoT and Edge Computing [66]	Artificial Intelligence (AI) in Edge	PnP-AI and its impact in the SC
Blockchain, IoT and Edge Computing [67]	Integrates IoT with Edge Computing and Blockchain	Proposed a model designed for a scalable and controllable IoT system
Blockchain, Smart Contract and IoT [30]	IoT identity, security and interoperability	Systems users, entities, register devices using Smart Contract with and control information in a web interface

The majority of Blockchain works are in applications related to cryptocurrency and digital finance applications. Table 3.1 summarizes the techniques, problems, and contributions of the Blockchain studies using SC and IoT.

Blockchain can be applied in various application domains and sectors of our society, covering almost all aspects of business, industry, finance, and governance, among others. Surveys, such as [57], concentrate on the works that address Blockchain applied to information systems and their security aspects. The work [58] studied the challenges and opportunities of using Blockchain as a database for IoT Apps.

We can find a pros and cons analysis of Blockchain integration's possibilities with IoT in [59]. The work combines Blockchains and IoT and emphasizes the power of this union of technologies. It can be powerful, where Blockchain provides resilient and truly distributed peer-to-peer systems and the ability to interact reliably without auditing. By approaching Smart Contracts, the work shows that it is possible to automate complex processes, with IoT devices being the contact points with the physical world. In the article, the combination of Smart Contract, Blockchain, and IoT is a breakthrough in automating workflows in new and unique ways, which enables cost-effective and time-saving cryptographic verifiability. Its conclusion estimates that the integration of Blockchains into IoT Apps causes significant transformations in various sectors of the economy, bringing new business models and rethinking systems and process implementation.

As in our proposition, article [60] discusses the implementation of a SC that is integrated with the Blockchain providing IoT devices with a secure communication platform. The article proposes a framework that securely integrates the physical layers, IoT communication, and application interface. Differently, our work focuses on investigating the impacts of using Blockchain in sending the message to an SC API after registration, identification, and recognition of IoT devices.

This article presents a prototype in which all operations related to Blockchain use an API gateway. The work [61] addresses the security requirements for an IoT Blockchain network and discusses how these can be satisfied through Smart Contracts. The paper addresses the main challenges associated with IoT devices' security and trust in Blockchains, presenting a design of a Global IoT information distribution system using Blockchain.

Some studies use the Fog Computing approach in their IoT Apps architectures and our research. They show the challenges of processing as much information as possible at the edge of the network [62, 63]. Use this paradigm. It allows for sending consolidated data to the cloud, isolating the network's segments with bandwidth economy, an essential feature to efficient IoT Apps.

Edge computing based on cooperation and collaboration is proposed in [64] to share resources and deliver services. An incentive-based mechanism is adopted to

offer a reward for the participant in the process using Blockchain.

The research [67] discusses data privacy and the benefits of applying Edge Computing and Blockchain in Industrial IoT (IIoT) scenarios, implementing experiments in Ethereum to evaluate security, performance, and energy efficiency.

The Blockchain is used in security management to block and revoke access to malicious users, responsible for identity authentication, public keys, user attribute sets, and revocation lists. The work [65] approaches the security and revocation of data and access for Smart Factory services in IIoT. The work proposed an attribute-based access control scheme and protocol for the Smart factory supporting traceability and revocation over a Bilinear Diffie–Hellman assumption. This result of work is schemes that optimize the size and overhead during the public key generation, data encryption, and data decryption stages.

The work [30] develops a system based on Ethereum to identify and authenticate IoT devices through Smart Contracts. The DApp [68] published on GitHub uses a web interface for registration and authentication of devices and Ethereum Smart Contracts for device identity, assigning Payload and Metadata Validate using Merkle Tree. We use its platform and Smart Contracts as a library and background for **Blockchain API Gateway** and **IoT Edge API Gateway** development proposition.

3.3 Semantic Web and Blockchain Related Works

Blockchain enables a secure and immutable database. These characteristics, together with Semantic Web, which can give web services the possibility to consume and obtain knowledge as a graph, can give new resources and functionalities.

Research and Implementation exploring the technical aspects of the Blockchain and applications linked to problem domains such as Industry 4.0 (I4.0),IoT, and SC are already widely studied in the literature. The works with Semantic web, Blockchain is not extensively covered and only recently found few relevant investigations. In this session, we present the main results found related to the focus of our research.

The article [69] goes deeper into the benefits of the Semantic Web and Blockchain, providing an overview of scenarios that benefit from the union of these approaches and analyzing its advantages and disadvantages.

The work [70] shows the results of a project aiming at developing the conceptual schema of Ethereum using Unified Modeling Language (UML). The motivation for this was that most Ethereum literature is found from a technical or an economic perspective. Developers, researchers, and students need a simple model to understand the deep foundations of Ethereum, in contraposition to having that search

details of this technology and its objects relationships in many books, papers, and web references. This research differs from our work because it does not use Semantic Web for modeling but UML.

The research [71] describes the BLONDiE ontology, representing the semantics of structures related to relevant Blockchain projects: Bitcoin, Ethereum, and Hyperledger. This ontology focuses on integrating the standard data formats of the different Blockchain platforms, separate from our proposition that uses EthOn to deepen searches in Ethereum.

The article [72] presents the creation of a linked data index implemented on Ethereum. Unlike our work that applies the EthOn ontology, this work focuses its results on the BLONDiE ontology. The work implements a semantic index for the platform and exposes the data as Linked Data, indexed at block and transaction-level according to the BLONDiE ontology.

The SANSA Semantic Web tool in the [73] poster is used in an Ethereum Decentralized Application (DApp), the CryptoKitties game, an online game based on Smart Contracts that allows players to trade characters securely, virtual pets. The Project uses Alethio [74] as an Ethereum analytics tool, providing transactions and logs in Resource Description Framework (RDF) modeled on the EthOn ontology and consolidating some results. Like our work, this research focuses on the EthOn ontology's results to make Ethereum data more digestible for end users.

The work [75] investigates item description and discovery in a Blockchain to the Supply Chains using Semantic Web approaches, proposing a framework that provides an object discovery layer in resume object discovery, registration, and selection. The core of this proposition is to use Hyperledger Blockchain with an ontology of object and product discovery. Unlike our approach concentrate on Ethereum, it focuses its approach on Blockchain Hyperleger technology [76].

The research [77] presents futuristic scenarios and ideas and industrial scenarios, using Blockchain networks for data feed and Semantic Web for data interconnection. The work presents the DeSCA prototype that records the interactions of the participants of a Supply Chain system in BlockChain and replicates this data in RDF format using the BLONDiE ontology as a basis. As well as our approach, the motivation of this research is to show that the Semantic Web principles can be used in a decentralized internet using Blockchain as Background to compose futuristic DApp scenarios.

An empirical analysis of store RDF triples in Blockchain compared with JavaScript Object Notation (JSON) storage are presented in [78]. The data in the proof of concept is produced by the IoT devices and the Ethereum Gas costs of these operations and its effectiveness in querying the database using SPARQL. Our proposal, different from this work, uses a web middleware that, through Web3.py [79], the Python

library to speak with Ethereum Blockchain and on the fly generates the triple RDF of the Ethereum entity called via its Universal Resource Identifier (URI) link.

The work [80] presents a distributed database compatible with Blockchain named GraphChain that exposes the data with RDF graphs in a semantic model. GraphChain, to define its semantics, uses its own Ontology Web Language (OWL) ontology to define structural entities. Its graphs can be published as web-accessible linked data objects using HTTP and can be queried by SPARQL. Some prototypes using Java, C, and JavaScript are used to demonstrate the dynamic of this Blockchain. This work uses the RDF graphs as part of the structure of an unprecedented Blockchain. Different from our middleware propose that uses the current Blockchain Ethereum networks to extract the graphs exposing this data in a semantic model and using it with the available web ontologies and we EthOn extension the EthExtras ontology.

3.4 Conclusion

In this chapter we show the main related works from the literature that we use as a theoretical reference in our investigations into the SC problems addressed. We seek to cover research on the main topics that we will cover during our testing and presentation of contributions, Low Power networks, Blockchain, Semantic Web and ontologies.

Table 3.2: Table of related work by meets

Work	SC Network	IoT App	Low Power Network	Fog Computing	IoT Security	Blockchain	Smart Contract	Semantic Web
[47]	LoraWan real network	Collaborative Solution for IoT connectivity	LoRaWan	LoRAWan Gateway				
[81]	LoraWan real network using Blockchain	Decentralized wireless infrastructure for IoT connectivity	LoRaWan	Blockchain Hotspots Miner		Helium Proof of Coverage (PoC)		
[48]		LoRa Survey	LoRaWan					
[49]		LowPower Technologies for IoT App	Large scale Wireless Wensor Networks (WSNs)					
[50]	Big Data Analytical Approach	Sensor BigData and Hadoop						
[51]	LoRaWan real Implementation	Stations Using LORa FABIAN Protocol	LoRaWan					
[9]		In-door Stations	LoRAWan					
[52]			Numerical Simulation of LoRa					
[13]	In-door Campus University	LoRa Mote to monitoring well-being	LoRa					
[53]	Agriculture Smart Irrigation	Irrigation Nodes	LoRaWan	LoRaWan Gateway				
[54]	ioT network simulation using LoRaSim	LoRa						
[56]		BLE energy management	WPAN	MQTT Driven node discovery				
[55]				Distributed Analytics				
[55]					Blockchain for Information Systems Management and Security	access control, prevent fraud, verification, trasaction integrity	blockchain-based smart contracts for security	
[58]		Blockchain Database for IoT Apps			Blockchain as a Database			
[59]			marketplace of services between devices, store the hash of the latest firmware update, rent their house or car, buy and sell energy automatically, supply chain			Blockchain and Smart Contract as IoT Apps Platform	automation of multi-step processes.	
[60]	Blockchain based parking example					Blockchain and Smart Contract as IoT Apps security Platform	Security Framework	
[61]		IoT Authentication and Access control				A secure way to identify and locate Things		
[62]	Applications of the Internet of Everything (IoE)	Devices are clustered according to their location and functionality, reducing energy consumption, throughput, cost, and time overheads	ZigBee, and Bluetooth	Blockchain Fog-based Architecture Network (BFAN)	The architecture secures sensitive data with encryption, authentication using Blockchain	Repository of the data received from the smart sensors	Executed based on conditions met	
[64]		Service composition solution through volunteer computing		Ensure secure communication and service delivery for the participants	Blockchains are formed whenever a service request is initiated	Blockchain enabled resource sharing and service composition solution through volunteer computing		

Table 3.3: Table of related work by meets (cont.)

Work	SC Network	IoT App	Low Power Network	Fog Computing	IoT Security	Blockchain	Smart Contract	Semantic Web
[67]		Integrates IoT with edge computing and blockchain, which is called Blockchain-based Internet-of-Edge (BIoE)			Edge computing and Blockchain to establish a privacy-preserving mechanism	Experiment evaluations running on Ethereum	Transfer tasks to Server to process them between nodes	
[65]		IIoT, secure storage and sharing of smart factory data			Framework for secure and privacy			
[30]		Authentication and Authorization of IoT Devices			Smart Contract for Validation of Payload, Device Identification and Metadata Verify	Ethereum	Payload Validation, Authorization Devices	
[69]						Agnostic Blockchain		RDF Content, Virtual RDF, Semantic Meta-Data, External Points
[70]						Ethereum		conceptual schema in UML
[71]						Bitcoin, Ethereum, and Hyperledger		BLONDiE ONtology
[72]						Ethereum		Semantic index to the Ethereum Blockchain platform
[73]						Ethereum	CryptoKitties game	Web framework SANSA applied in the CryptoKitties use case, EthOn Ontology.
[75]	Supply Chain	Object Discovery				Hyperledger	Item tracking operations	
[77]						Bitcoin and Ethereum		BLONDiE Ontology
[78]		sensors that store their data into Blockchains				Ethereum		Aaalysis of the cost of storing into Blockchain solving SPARQL queries reading directly the RDF or using a virtualiser fed RDF
[80]						Blockchain compliant distributed database exposes data using Semantic Web		Own OWL-compliant ontology

Chapter 4

Models using Blockchain, Smart Contract and IoT

This chapter has content published in the book Chapter [34], and covers relevant models found in the literature and industry, using IoT, Blockchain.

4.1 Blockchain and IoT propositions

There are still few applications using Blockchain as a security background or data for IoT applications. Those in the prototype or start of production stages started their work recently. The union of these two technologies gives security and robustness to some scenarios [82].

Several projects propose using Blockchain and IoT, and we list some of the most relevant ones that illustrate the integration of these two technologies.

4.1.1 Chronicled

Projects like Chronicled [83] are committed to providing security for IoT devices and integration with the most popular Blockchain networks. They are using Smart Contracts for registration and identity verification.

4.1.2 AEROToken

The AEROToken [84] project proposes to be a drone road infrastructure within the United States, coordinated by Smart Contracts on the Ethereum Blockchain. In some countries and the United States, drone operators need low altitudes over private properties. In addition to other restrictions for security and privacy, the project faces these challenges and constraints of commercial drone service demands

[85]. AeroToken enables property owners to authorize drone flight, notifying and logging via Blockchain that their airspace is available.

4.1.3 The Chain of Things

The Chain of Things (COT) [86] develops innovative and futuristic IoT solutions using Blockchain, the ChainofSecurity project, which provides security features for IoT; the ElectricChain, which provides solar power generation data publicly in near real-time, and the Chain of Shipping. Chain of Shipping aims to mitigate fraud and process management inefficiencies that use Bills of Lading (BOL), which, even when used digitally, can be duplicated or hacked. This project proposes the creation of a Smart BoL, eliminating paper forms and centralized databases, ending fraud through Blockchain and SmartContract. Using a Smart contract would require executing the steps provided for in the BOL between participants and stakeholders, generating alerts in case of non-compliance with the routine contained in the contract.

COT also works on conceptual projects, such as Liquidstar and Blockpass; Liquidstar is a network with intelligent batteries of decentralized management using Blockchain and Smart Contract. Portable features the so-called "Solar Buckets." replace expensive traditional networks and liquid fuels with Mobile Virtual Networks. Its purpose is to be the future of energy for more than 1 billion people globally, who live in places without electricity grid coverage and many others who do not have a reliable source of energy. Blockpass is intended to be an identification application using Blockchain, its primary use case being industrial applications. It provides an identity layer, a protocol that allows the interaction between identity profiles with devices allowing the creation of applications with reliable interaction between different entities.

4.1.4 ADEPT

Autonomous Decentralized. Peer-to-Peer Telemetry (ADEPT) is an IBM project in partnership with Samsung that uses Blockchain to build a distributed network of IoT devices in a decentralized way [87]. The project is based on BitTorrent for sharing files, Ethereum for Smart Contract, and TeleHash for P2P messaging. ADEPT has propositions for use in domestic environments, creating a network of autonomous devices. In this network, appliances signal operational problems and make software updates on their own, communicating with other nearby devices to control and guarantee energy efficiency[88].

In an ADEPT proof of concept, an innovative washing machine uses a Smart Contract that provides the rules for buying detergents and choosing retailers. Demonstrating how to make the IoT device manage its supplies, carry out purchases and

maintenance, and act in conjunction with other devices to optimize the energy of the domestic environment, taking place without a central controller orchestrating or mediating [89] operations.

For this purpose, the Samsung W9000 washing machine was used using Smart Contracts to request new supplies from the detergent dealer, pay for the order itself and receive notification from the seller that the detergent was correctly paid for shipped, notifying the washing machine owner's smartphone. Wash, as this device is under intelligent home management. One of the main challenges faced by this solution is the scalability strategy, in order to incorporate the large number of IoT devices expected to come into operation in the coming years, which should surpass the billions of devices [90].

4.1.5 MyBit

MyBit Network uses Blockchain and Smart Contract to manage assets [91] using Ethereum and MyBit Software Development Kit (SDK). MyBit is proposing a Decentralized Development Fund (DDF), offering services that divide proportional earnings and revenues between groups of IoT asset owners. Drones, scooters, bicycles, cars, machines can be shared without the need for a centralized manager because Smart Contracts govern the rules for financial gains.

4.1.6 Slock.it

Slock.it [92] develop tools for Smart Locks using Ethereum Blockchain, IoT, and Smart Contract. The idea is to use a device IoT to control, for example, rental, sale, or seasonal use of a property. In this example, the Smart Contract defines the business parameters as deposit and rent value, being possible for the lessee to open and close the padlock and access the property when making a transaction to the Ethereum Smart Contract. Rent collection, refunds, and discounts are now managed without interference from third parties using a centralized model [93].

Using the Slock.it solution would allow current models of shared property use to be operated without companies. Slock.it provides these intelligent locks that can be used at various points of a SC, such as cars, bicycles, scooters, and gondolas. Cars, for example, can be anywhere in the city until the next customer, located with a phone app, can unlock their lock and use it. They also work in transport solutions to provide charging infrastructure for electric vehicles and control the use and payment of the service through Blockchain and Smart Contracts. This business model made up of intelligent locks could threaten large shared businesses like Uber and Airbnb, and Smart Contracts already have agreements between participants, making them irrelevant.

4.2 Smart Contract Scenarios

The junction of decentralized finance strategies and IoT devices brings new possibilities of use and forms of remuneration for services using cryptocurrency [94]. The application scenarios for sharing economy are potential users of Blockchain, Smart Contract, and IoT, which guarantees the required security standards. In common, they use Smart Contracts that implement the promises established between the parties, guaranteeing the previously agreed financial and operational parameters.

Figure 4.1 shows scenarios of using Smart Contract and Blockchain for fuel payments at a gas station, Drones paying toll to fly in properties and houses with Smart Locks.

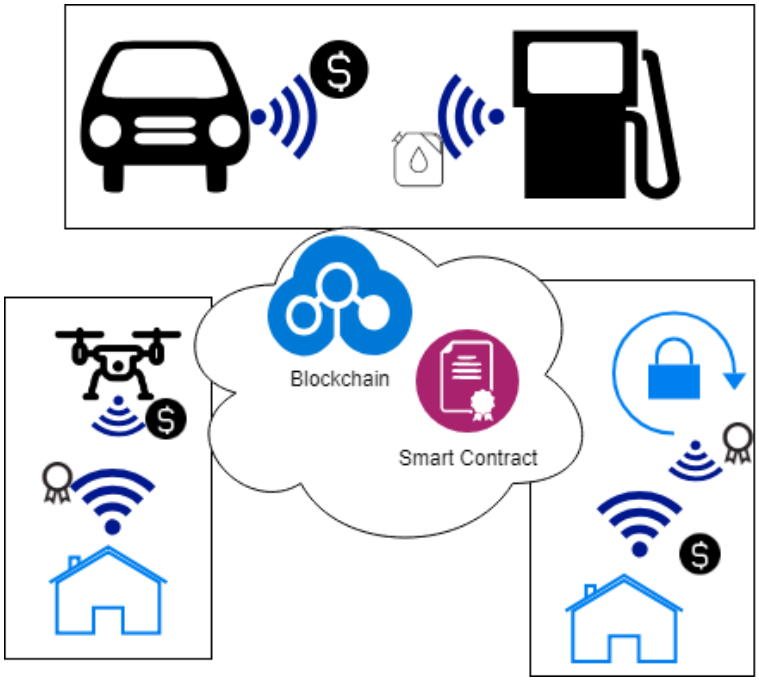


Figure 4.1: IoT in a communication using Blockchain and Smart Contract

Without a central and autonomous control entity in private urban transport, a payment system can become viable and safe. An agreement established by Smart Contract makes it possible for a passenger's smartphone to communicate with the car and automatically make payment autonomously, offering transparency, security, and confidence in using the service.

Another scenario of autonomous payment and financial movement without intermediaries is the payment of fuel for a car at a gas station. In a traditional scenario, the application stores the user's credit card and exchanges information between card machines and financial companies to release credit for purchasing fuel and payment to the gas station. A centralized entity is not necessary for a scenario using only Smart Contract applications.

The vehicle runs an application that accesses the public Blockchain. It sends cryptocurrencies to a Smart Contract that already agrees with values previously established between the parties and releases the fuel at the bomb. The gas station can directly interact with the Blockchain through its app to determine if a vehicle has paid and record how much gas it has purchased.

In addition to gas payments, the decentralized application can monitor the vehicle's autonomous fuel quantity, anticipate refueling needs, and automatically propose a convenient route to a gas station that contains better price adjustments, forecasts, and notifying each transaction balances.

In another example of a standalone autonomous payment scenario., the application receives a message informing that based on its daily shopping list and home devices inventory. Was identified local retailers with the best prices according to the agreed budget using the anticipated agreement of the Smart Contract among the participants, bought what was missing, and arranged for its delivery at the right time.

Smart Contracts may have to be parameterized by profile; an example of this would be a vehicle where the car owner is allowed to pay for fuel or park in his office when driving. However, his child is not, and the driver's identification is required before autonomous transactions.

Smart Contracts may have different agreements per user profile; an example of this would be a vehicle where the car owner can pay for fuel or park in their office while driving. However, the owner's child does not, with the identification of the car's driver being required before autonomous transactions.

The project Aigang, for instance, has Smart Contracts on Ethereum to contract and process insurance service requests such as Maintenance and Payment Notices (MPN) automatically between IoT devices [95]. It uses AIX currency to trade investment opportunities in its products, using Smart Contract to agree on different levels of risk and reward. Aigang includes claims handling, fraud detection, renewal, and payment in addition to accurate insurance pricing, reducing administration costs and delays in service requests.

4.3 Blockchain Storage propositions

An important project that proposes to be a distributed object storage service is the Interplanetary File System (IPFS) [96]. It serves files decentralized and is outside the control of companies, governments, content tracking agencies. Each file receives a unique fingerprint represented by a cryptographic hash, ensuring its uniqueness and immutability. By relying on collaboration and node maintainers, a framework using Smart Contract and Blockchain can create monetary incentives and rewards

for keeping copies of files at the edge of the network.

An important feature when using IPFS is Inter-Planetary Name System (IPNS). The IPNS is one of the proposals for naming using names for digital assets in IPFS. In short, IPNS is a hash associated with a record containing information about the IPFS hash to which it is linked.

Digital media industries benefit from IPFS resources, like books, music, photos, movies, among others, when stored, cannot be duplicated on the network, and it is even possible to trade them based on Smart Contract. An example of monetization of these assets is the Non-Fungible Tokens (NFT) that generate value to tangible assets to trade and guarantee their authenticity and ownership, generating a new way of trading and valuing assets.

4.4 Conclusion

In this chapter, we show some cases of relevant uses of IoT already found in the literature and the industry, using the set of resources provided by Blockchain and Smart Contract. Some of the projects are currently disruptive references in adopting projects using a decentralized paradigm.

Part III

Case Studies

Chapter 5

The Low Power Network

In this chapter, we approach the scenarios SC IoT using Low Power Networks as Lora and BLE in Fog Computing architect paradigm. This chapter has a significant part of its content already published in the work [29].

5.1 SC IoT App Network

SC networks are a potential scenario for using IoT in network architectures using Low Power technology. Urban environments are environments where we find users and applications of different interests and requirements, each city having a different capacity in basic infrastructure. The study [97] proposes a mix of Low Power, Zigbee technologies for sensors and Wi-fi to access the backbone in order to provide vehicle management and traffic monitoring in profile SC IoT App. The intelligence must be achieved with a minor human intervention as an essential point of SC IoT App, Machine-to-machine communication requires reliable networks, Low Power devices, and minimal infrastructure structure before being considered "smart."

The [98] study of a review of wireless technologies in the SC shows a comparison and lists the problems that difficult coexistence among them. The reason is that most popular wireless technologies available today employ ISM frequencies. WiMax is very popular in providing long-distance wireless communication, such as in rural areas, saturating and increasing interference between 2.4 GHz ISM networks. (Industrial, Scientific, and Medical). ISM frequency becomes a problem due to the growth of technologies that use this range, such as Wi-Fi, Bluetooth, and ZigBee. Home automation, telemedicine, and healthcare applications already employ Wi-Fi on a large scale.

An network infrastructure using Wi-Fi has high power consumption, making some SC IoT scenarios unfeasible. The Wi-Fi stays restricted to situations SC with wall power or easy battery charging. Often the proposes of SC IoT implementations use gateways connected to a fiber optic backhaul or high-throughput wireless

connections, which act as intermediaries of the IoT network to the Cloud or local infrastructure. However, when we do not have any available backhaul or the cost of infrastructure and energy prohibitive.

Given all these problems and limitations of other technologies in these scenarios, Low Power networks become a relevant option. The Low Power network options are LPWAN [99] [13] and WPAN. The popular technologies of this networks are Long Range (LoRa) and Bluetooth Low Energy (BLE), both using unlicensed frequency.

The LPWAN are networks formed by connecting battery-powered devices that send payloads by a long range and using Low Power Consumption. They are associated with networks of sensors and devices that communicate in environments of relevant geographic distance in applications that transmit only little data with low throughput. This throughput is sufficient to send small messages a few times a day [100].

In a SC IoT App used to collect information on water supply and sewage collection networks can provide reliable information for management, making it possible to predict critical situations such as; amount of infiltration, detection of blockages, flood forecasting insights, and water pollution control. Flexible, low-cost monitoring approaches are increasingly relevant and challenging, given the growing need for management, control, and performance in this scenario.

Sewage monitoring networks have specific requirements to be deployed. Devices must be resistant to unfavorable conditions and often subject to interferences and aggressive nature effects coupled with sensing and data transmission with low power consumption to use long battery cycles and reduce service maintenance in difficult-to-access locations. Using gateways that depend on wall power in these scenarios is a limitation. In addition to the hardware cost, this environment's physical access and management cost increase proportionally [101]. Sewage monitoring is an example of SC IoT Applications SC IoT that send few and small loads during the day and that are often installed in environments with restricted physical access can benefit from the LPWAN features. Pipes, power sources, dumps, high towers, and mountains are other examples. LPWAN has requirements that meet this IoT applications [8]. They operate at ultra-low power, allowing a long battery life or even working without using them, with clean energy. This is advantageous as there is little exchange. Important in applications where economic restrictions on the adoption of thousands of devices are relevant in addition to reducing environmental impact. As the devices are low cost and widely accepted, there is no need for a SIM card or equivalent; simple installation and minimal maintenance. The communication activities of the devices vary from application to application. However, for networks in LPWAN networks, the power consumption limit must demand that the object wakes up the minimum necessary to send or receive data, rejecting synchronized mesh-type

networks, making ALOHA the preferred non-cellular star network architecture. The LPWAN technology LoRa [14] for example, promises in its specifications links of up to 45 km, using rates between 0.3 and 50 kbps in unlicensed frequency.

Comparing the 5G networks with LPWAN, we can observe that 5G has the technological requirements for most SC IoT Apps. Such as high bandwidth and data traffic capacity, being prepared and designed for mass connection, wide area coverage, and low latency [102]. 5G is considered one of the disruptive technologies capable of revolutionizing SC, nevertheless, despite the characteristics capable of this task, problems such as power supply to hudge quantity of devices and challenges such as managing the wide distribution of devices, especially devices deployed in remote or inaccessible areas and the high cost of building and maintaining the infrastructure of these networks [103]. Avalanche control applications such as [104] in support of radars could use the long-range features of LPWAN as communication technology auxiliary of sensors and tools deployed in roads or places that are often difficult to access in winter.

These challenges limit realizing 5G in SC, driving research like ours that develops new approaches, methods, techniques, and tools. The performance of 5G communication depends on many variables, such as spectrum availability, bandwidth, and the number of cells [105]. Despite this, the spectrum may not be sufficient to support the explosion of devices and traffic to which the network will be subjected, leading to communication problems in scenarios of areas with a high density of devices [106].

5G is not meeting some of the requirements of the simplicity of deployment and infrastructure and cost required by the sewage applications, for example. Using LPWAN LoRa provide network infrastructures SC, can mitigate some of these problems and limitations of 5G networks and can bring independence to some stakeholders in this process, such as mobile operators.

The Edge devices in a SC IoT Apps can be equipped with a WPAN hardware interface. WPAN is a Low Power network connecting devices near the user. BLE is the most popular WPAN technology available for shipment, be able to link devices in ranges of 10 m to 1.5 Km[107], using unlicensed frequency for communication.

Some SC applications may require business logic or decisions close to IoT devices due to a low tolerance for communication delays with external servers or edge device computing limitations.

A networking paradigm such as Fog Computing allows processes or services to be managed and run close to the edge of the network, as the internet gateway, reducing the amount of data transmitted to the Cloud and improving application efficiency. Networks with low rates, such as LoRa links, can benefit from architectures such as Fog Computing [108],[15]. Fog Computing can be utilized as an efficient architecture to reduce delays and enhance the energy efficiency of the SC IoT Apps [109].

We propose A merge of Low Power Wide Area Network (LPWAN) LoRa and Wireless Personal Area Networks (WPAN) BLE networks as SC IoT network infrastructures SC , technologies that have long range access and device popularity features, different from Wi-fi and Zigbee.

Propositions using LPWAN LoRa together with the popular WPAN BLE can bring flexibility to SC IoT Apps that need to use the technological resources of both networks, as well as in the initial proposition of this chapter on Wi-fi and ZigBee composition. This network LoRa and BLE has, by definition, Low Power characteristics and has attribute adherence in use cases where energy resources are not available at the edges.

BLE was proposed to be agnostic, allowing connection between different devices with a focus on low-power applications, with reduced hardware costs. acrshortble is an independent standard of the "classic" Bluetooth, and it has a communication and data exposition structure called Generic Attribute (GATT) that manages and stores information using services and their characteristics. Compared to other wireless technologies, its protocol was designed for low consumption and mobility devices, having an access standard, built-in security, and simplicity in data extraction.GATT and its simplicity as a protocol was a decisive factor in choosing BLE for this research over other radiofrequency technologies with a mobility profile. Bluetooth technology is versatile and virtually ubiquitous in the mobile devices and sensors used in real-world service applications. The work [110], uses Blockchain as a security gateway for IoT BLE Devices.

5.2 Extending a Smart City LPWAN LoRa using WPAN BLE

Extending an LPWAN using WPAN devices with currently more significant popularity in the industry, we use the low-power network technologies LoRa and BLE, this network maintains the characteristics of low energy consumption considering the technological characteristics.

The figure 5.7 represents a network SC using our proposed architecture. LPWAN. In it, we can see edge gateways that we call LoraEdge and their respective links LoRa with the gateway of contact with the Cloud network, LoRaFog. The LoRa links are long-distance in a star topology, with LoRaFog being the central point of contact for edge devices in contact with LoRaEdge.

The WPAN in the edge network is formed by the LoRaEdge gateway and its BLE links in a star topology. The essential function of the LoRaEdge device is to extract data from the BLE devices. In Figure 5.7, we see some examples of applications,

established links for health devices, and vehicle traffic control.

In LoraFog, we use a publish/subscribe protocol like Message Queuing Telemetry Transport (MQTT), used globally for IoT applications, designed to be light on message transport and useful for connections where we have small messages at low baud rates.

For the LPWAN and WPAN integration, we propose two algorithms, to be used in LoRaEdge and LoRaFog.

5.2.1 LoRaEdge algorithm

The algorithm proposed for LoRaEdge, Algorithm 2, describes the routine to extract information from GATT from BLE devices and sends the extracted features to LoRaFog using the LoRa interface.

The call to *GetIoTApp*, IoT Application (IoTApp) represents an app installed by end users on LoraEdge, just as the IoT app store. The idea of IoTApp is to receive the data extracted from the device and other LoRaFog, and aggregate, generate insights, perform Machine Learning algorithms, and filter or consolidate data. This implementation satisfies the Fog Computing architecture that aims to have information processing closer to the edge and before contact with the Cloud. For example, in a hypothetical fire detection application, edge data such as temperature and smoke could predict fire risk.

The *LocalSense* function reads data from sensors installed in LoRaEdge, and this can be interesting because using local sensors in the gateways can bring data outside the area of activity of the edge sensors. The *LoraSocket* function represents communication through the LoRa device interface, and the *Received* function receives data from the LoRaFog device.

Bluetooth.Scan fetches devices BLE, *Bluetooth.GetAdv* fetches the properties of a device. *Bluetooth.Connect* function to attempt a connection to the BLE device found. *Services* retrieve the available GATT services from the device.

For each characteristic of a service GATT, the Universally Unique IDentifier (UUID) *char.Uuid* is extracted, and its value is read by *char.Read*.

The data received from the LoraFog device, values from the local sensors, and the extracted features are sent for processing by IoTApp in the *iotapp.Process* function.

The result of this processing is sent to the LoraFog device by the *Send* function. Finally, BLE connections are interrupted by *Bluetooth.Disconnect*, waited for a period, *RandonSleep*, and *Bluetooth.Scan* start again.

5.2.2 LoRaFog gateway algorithm

The proposed algorithm for the LoRaFog gateway, Algorithm 1, receives data from LoRaEdge via LoRa and publishes the results to a topic MQTT in the cloud.

The IoTApp abstracted by *GetIoTApp* in LoraFog represents the same abstraction found in LoRaEdge. However, in the execution context of a gateway, it can aggregate data coming from the edges of various LoRaEdge.

LoraSocket represents the LoRa interface communication. *Mqtt.Connection* is the initialization call to the Publish/Subscribe broker server using the MQTT protocol. *Subscribe* is the function with subscribing to a topic, getting IoTApp by *GetTopicURL*.

Data received from LoRaEdge devices by *Received* is processed by IoTApp using *Process* and published to a topic MQTT by *Publish*.

CheckMessages() checks for new messages from the subscribed MQTT topic, *Put*() sends messages to IoTApp and *Send* to LoRaEdge devices. The algorithm ends with *RandonSleep*.

Algorithm 1: LoRaFoG algorithm

```
1  iotapp = GetIoTApp()
2  socket = LoRaSocket()
3  topic = iotapp.GetTopicURL()
4  mqtt = Mqtt.Connect(mqtthost, user, password)
5  subscribe = mqtt.Subscribe(topic)
6  while True do
7      edgedata = socket.Received
8      if edgedata then
9          fogdata=iotapp.Process(edgedata)
10         mqtt.Publish(mqttTopic,fogData)
11         msg=mqtt.CheckMessages()
12         if msg then
13             iotapp.Put(msg)
14             socket.Send(msg)
15         end
16         randonSleep()
17     end
18 end
```

5.3 The Testbed

For our experiments, we opted for the Pycon-Lopy4 kit. Lopy4 is an IoT development board with Quad MicroPython support (LoRa, Sigfox, WiFi, Bluetooth), equipped with ESP32 DualCore 8MB flash memory, 4MB RAM.

Algorithm 2: LoRaEdge algorithm

```
1  iotapp = GetIoTApp();
2  localsenses = LocalSenses()
3  socket = LoRaSocket()
4  Bluetooth.Scan()
5  while True do
6      fogdata = socket.Received
7      adv = Bluetooth.GetAdv()
8      if adv then
9          connection=Bluetooth.Connect(adv.mac)
10         gattcs=connection.Services()
11         for gattcs in service do
12             chars = service.Characteristics()
13             for char in chars do
14                 uuid=char.Uuid()
15                 char=char.Read()
16                 hw=localsenses.Get()
17                 edgedata=iotapp.Process(fogdata,hw,uuid,char)
18                 socket.Send(edgedata)
19             end
20         end
21         Bluetooth.Disconnect()
22         RandonSleep()
23         Bluetooth.Scan()
24     end
25 end
```

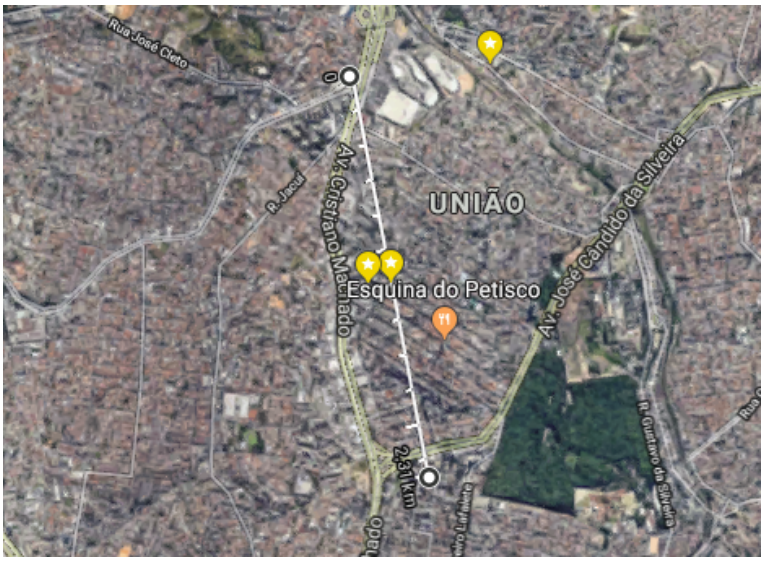


Figure 5.1: A Line of sight 3.2 Km LoRa link in Belo Horizonte, Brazil

For the LoRaFog device, we use the Extension Board 3. For the LoRaEdge device, we use Pysense, which has an ambient light sensor, barometric pressure sensor, humidity sensor, 3 accelerometer 12-bit axis, temperature, and USB connection. Each board has a 900 Mhz antenna for LoRa communication. The code used can be found in [111].

5.3.1 The range

In a first test-bed setup to understand the range limits of LoRa on a SC, we configured the LoraFog device with the PyR Antenna Kit LoRa (868MHz / 915MHz) at a height of 20 meters.

Using varying distances, we set up the 1 km LoRa links in a line of sight and shadow (underground and tunnels). We use the LoRaFog device for the base station and, as the remote, the LoRaEdge device. These devices send and receive payloads at a random time using Raw LoRa or LoRa-Mac.

In all these scenarios, the device running LoRaEdge succeeded in sending the messages to LoraFog base.

We were able to send messages on LoRa links up to 3 km on the line of sight, the remote shadow points communicated on links up to 1.5 km, and underground, we were able to send messages with links up to 1 km.

Figure 5.1 shows a 2.3km line of sight link pos'sible with this communication LoRa experiment in Belo Horizonte, Brazi

Considering the distances of LoRa links reached, we can observe that this technology has a potential adherence for use SC IoT Apps that need to communicate over a long range. The distances reached using this development kit could, for

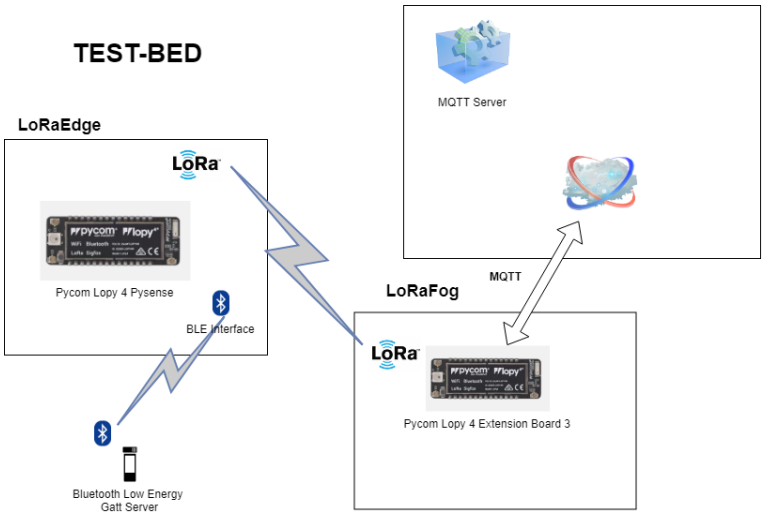


Figure 5.2: The test-bed using LoRa and BLE

example, cover the diameter of several cities on the planet.

Based on this observation, we believe that devices and antennas with a advanced setup could reach greater distances compatible with the profile of large metropolises.

5.3.2 Scanning the BLE devices

To validate and verify the feasibility of extending an LPWAN using WPAN, we generate a testbed applying the routines of the LoRaFog and LoRaEdge algorithms, using the network architecture of Figure5.7.

This experiment aims to send data extracted from the BLE device of the LoRaEdge device to the LoRaFog device using the LoRa link.

We use the MI Band Xiamoi as BLE device and extract one of its GATT read characteristics using the LoRaEdge.

Figure 5.2 represent the setup of this test-bed using LoRa and BLE, with LoRaFog and LoRaEdge algorithms.

We have been able to extract messages from BLE devices and send this information through LoRa Link.

Some side effects have appeared and need further investigation. During the scan phase, messages from multiple BLE advisor devices were received in range, causing delays and problems during the target characteristic extraction. To LoRaEdge receive payloads of LoRaFog is necessary, a window of receive implemented. In this testbed, we use Raw LoRa links, and this implementation has no guarantee of messaging or security. This communication and problems and security gap need to be addressed in future studies. In this experiment, we did not implement any IoTApp feature or its dependencies as *localsenses* , leaving this task for future work.

In a scenario using LPWAN and WPAN with Long Range Wide Area Network

(LoRaWAN) and BLE, some components of the Raw LoRa scenario need to be modified. The LoRaFog component is being replaced by the LoRaWan Gateway, responsible for the primary connection with the edge LoRa devices. Figure 5.8 represents this architecture.

Unlike our first scenario, in order to be able to send a message using LoRaWAN, it is necessary to compose the protocol backend, such as the network server LoRa and the application server LoRa.

We uploaded a test-bed with LoRaWAN, using the network architecture of Figure5.8 as a reference, with only the LoRAEdge component being configured. As in the test-bed with Raw LoRa, we implement no concept of IoTApp and its dependencies, leaving this task for future investigations.

Different from the previous test-bed using LoRa Raw, the LoRaWAN protocol has the guarantee of messaging or security implemented in its features.

In this experiment, we used the LoRa Server [112] project on containers installed in a desktop Linux representing the Cloud Network of architecture. This project proves the network servers, applications, and MQTT required for the LoRaWAN protocol.

To LoRa Gateways installed in we LoPy4 in a Pycom Expansion Board 3 running LoRaWAN Nano Gateway provide by Pycom [113].

For the LoRaEdge sensor device, we use the LoPy4 and Pysense, which has an Ambient light sensor, Barometric pressure sensor, humidity sensor, 3 axis 12-bit accelerometer, temperature sensor.

In this configuration, the security layer of LoraWan had to be set, requiring the use of Activation by Personalization (ABP) in LoRaEdge, requiring the DevAddr and session keys assigned during a procedure called activation.

We have been able to extract messages from BLE device and send the payloadn using LoRa Link, but some side effects have appeared and are an object of a future investigation.

The link from LoRaEdge to LoRa Gateway using AUS915 frequencies presented significant packet losses. It was not yet possible to interpret the adequate cause of this behavior, which varies according to the chosen data rate. The better results are in the communication setup with the LoRaEdge using DR 5 e LoRa Gateway SF7 / 125 kHz.

We do not integrate with MQTT and external API, and we only evaluate the arrival of the payload by the management tools of the LoRa Server.

5.4 Analyzing the scalability of a LoRa network

To evaluate the capacity of the LPWAN LoRa network, we simulate an SC IoT network scenario of thousands of devices with simulator LoRaSim [114], sending messages in different periods.

To evaluate the LoRa network scenario in an SC, we used LoRaSim using typical values of SC IoT Apps. LoRaSim uses the environment for discrete event simulation based on Python, the SimPy. We use it to simulate IoT communication in LoRa networks and analyze their scalability. The points in the graphs are the results of the averages of the values obtained in the simulation instances. The log files of simulation instances used in this chapter can be found in [115].

Figure 5.4 represents this scenario where all IoT motes send messages once a day and at the same gateway. For this simulation, LoRaSim received the parameters:

- number of nodes, representing the volume of IoT Motes
- number of gateways 1
- full collision detected, to be sensitive to collisions
- to create an Adaptive Data Rate (ADR) style communication, we used the LoRaSim parameter that optimizes the setting per node, based on the distance to the gateway.
- We use a range of 15 km

Scenarios were simulated with 100, 1500, 2000,3000,4000,5000,6000,7000,7500,8000 nodes, being collected 10 simulation instances using LoRaSim's *loraDir.py* script to each of these node volumes.

A simulation instance using one base communicating with 7500 nodes sending a message on average every hour simulating a day would run:

```
loraDir.py 7500 3600000 3 86400000 1
```

Scenarios were simulated with 5000,8000,10000,20000,30000 nodes, being collected 10 simulation instances using LoRaSim's *loraDirMulBS.py* script to each of these node volumes. A simulation instance using 3 bases communicating with 20000 nodes sending a message on average every hour simulating a day would run:

```
loraDirMulBS.py 20000 3600000 4 86400000 3 1
```

We can observe that when nodes send messages at the same gateway interval, we increase the number of collisions and decrease Data Extraction Rate (DER).

DER is a metric used by [54] to evaluate the simulation results of LoRa networks, which describes the proportion of messages received and transmitted in a period,

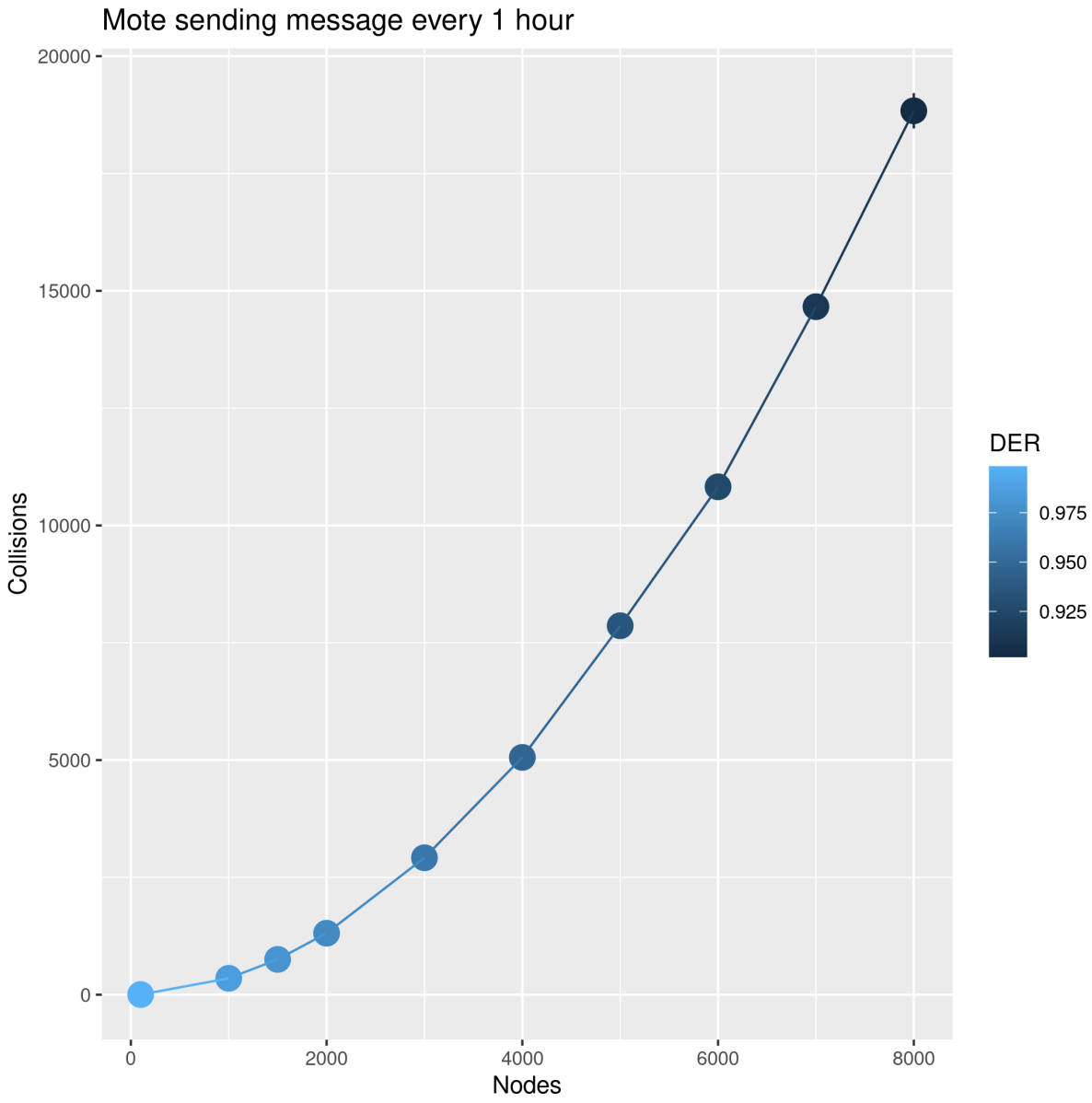


Figure 5.3: The Lora Motes sending messages to a gateway every 1 hour

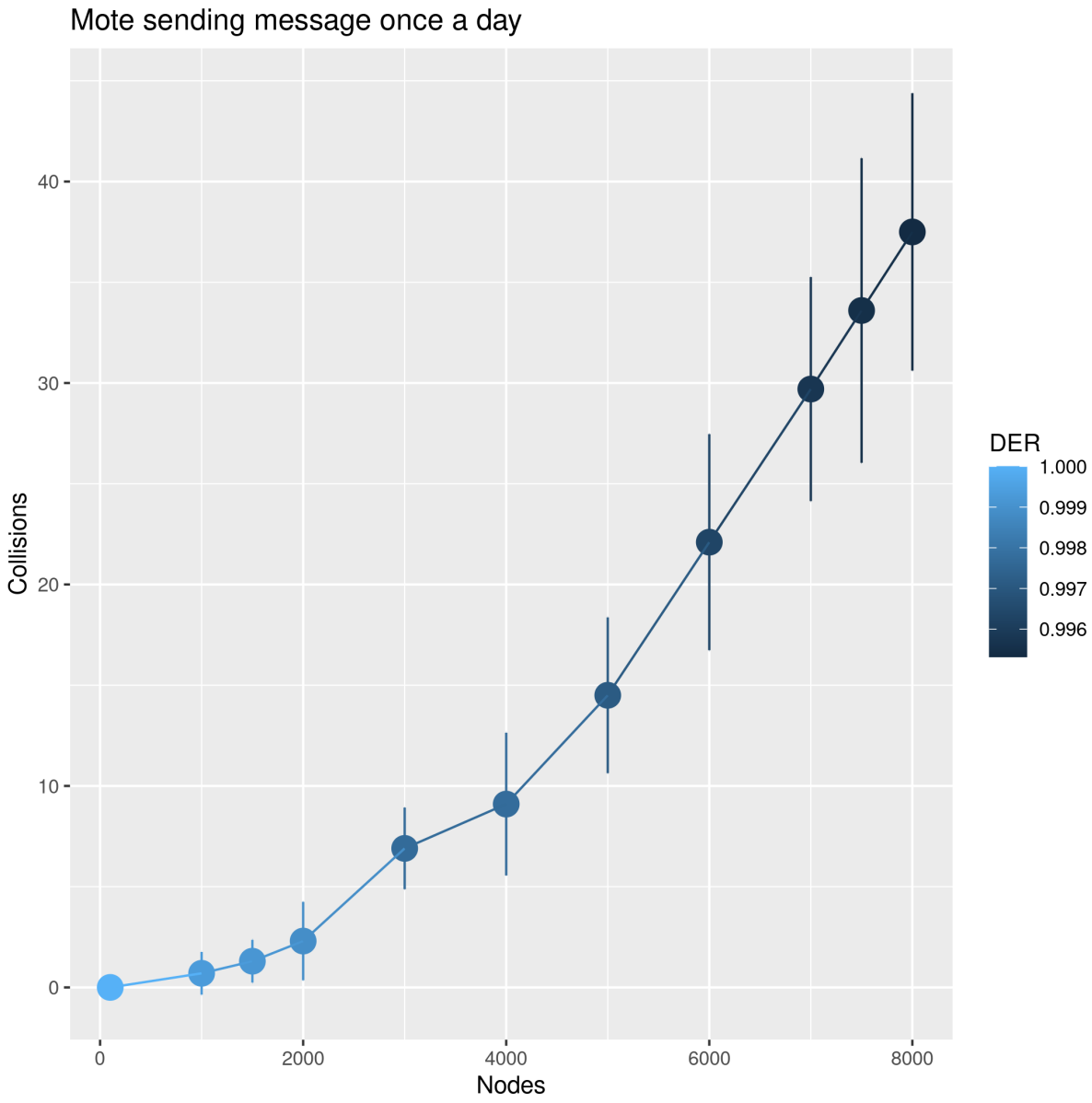


Figure 5.4: The Lora Motes sending messages once a day

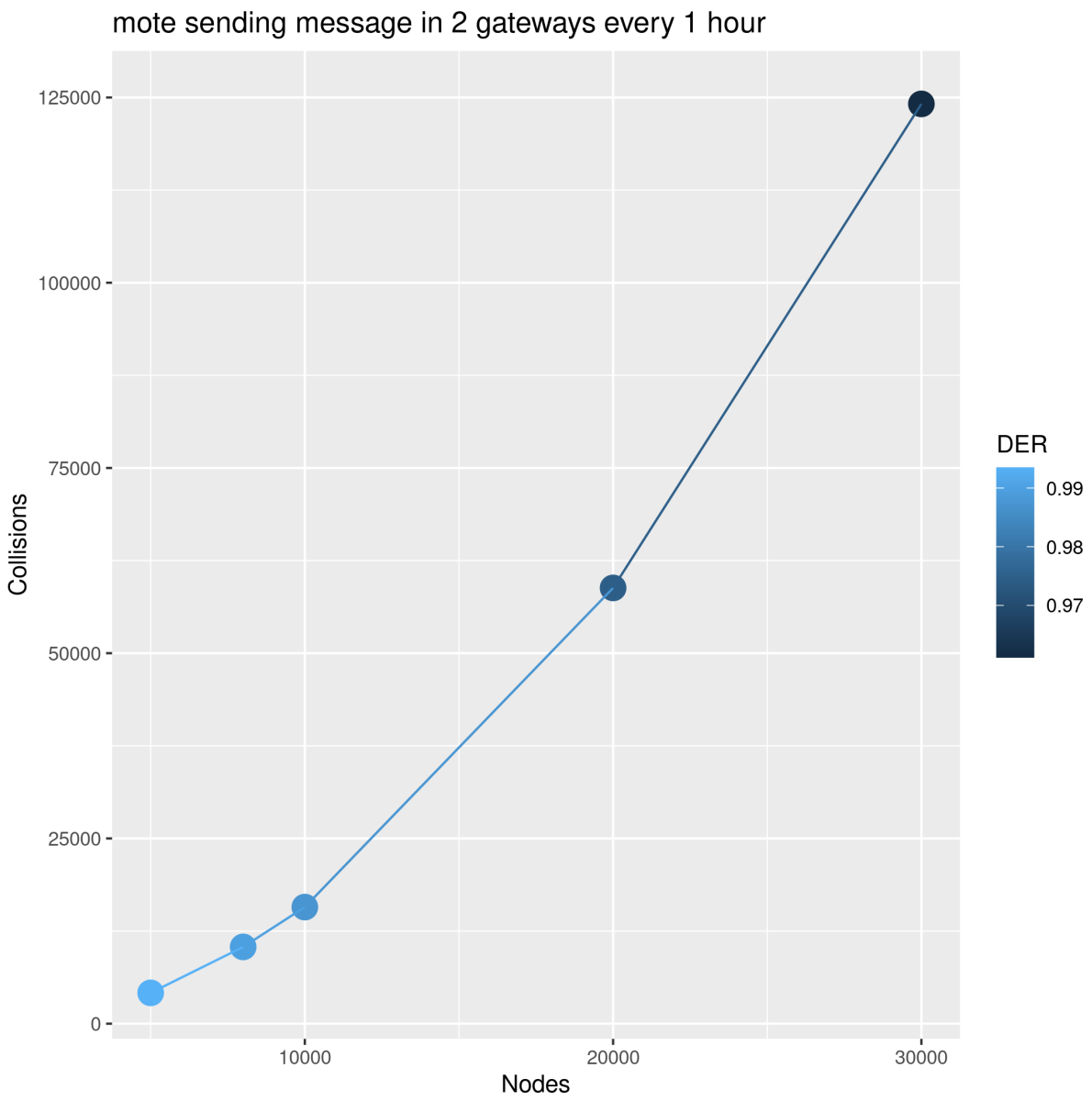


Figure 5.5: The LoRa motes sending messages every 1 hour in a multigateway scenario using 2 bases

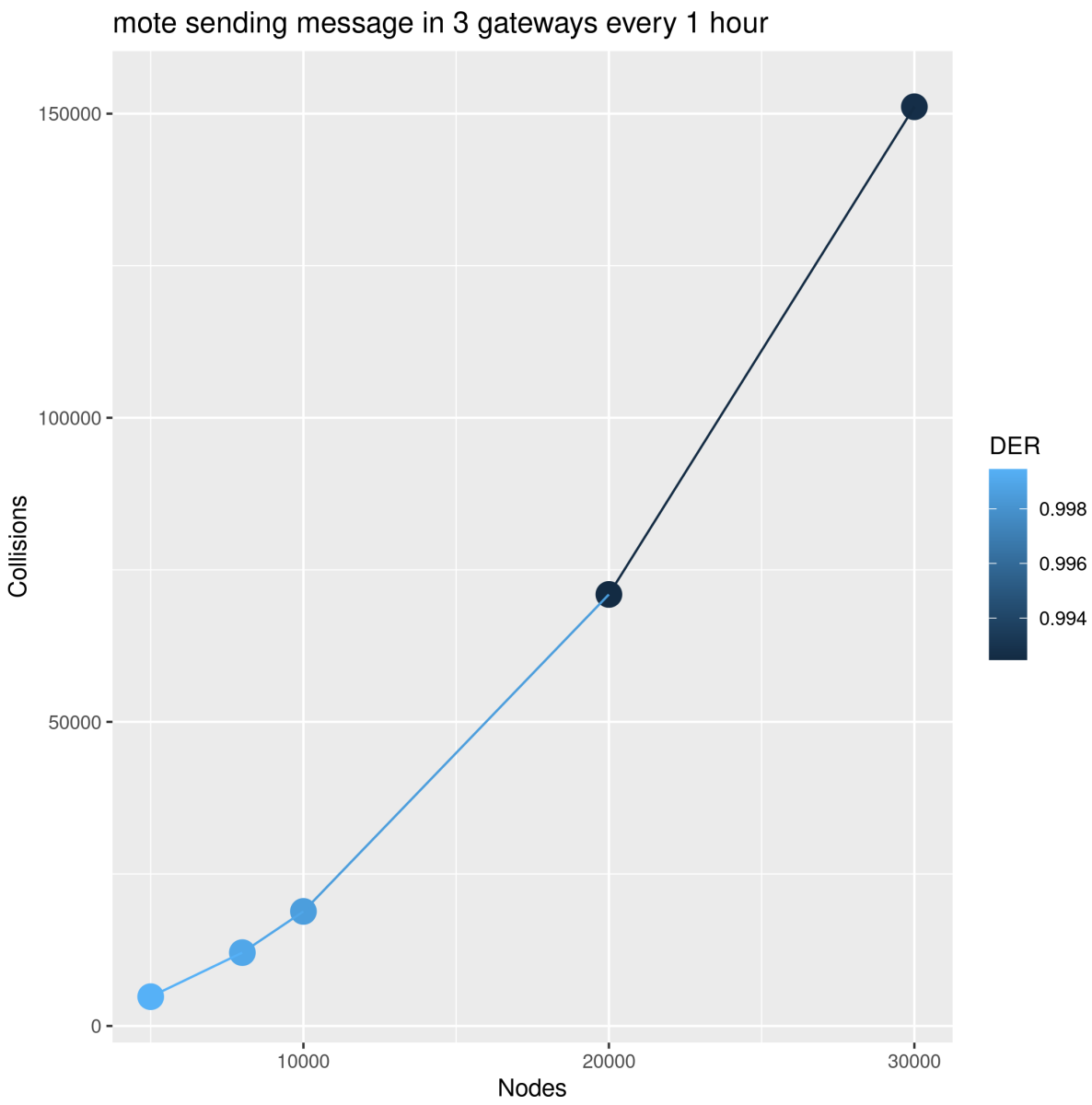


Figure 5.6: The LoRa motes sending messages every 1 hour in a multigateway scenario using 3 bases

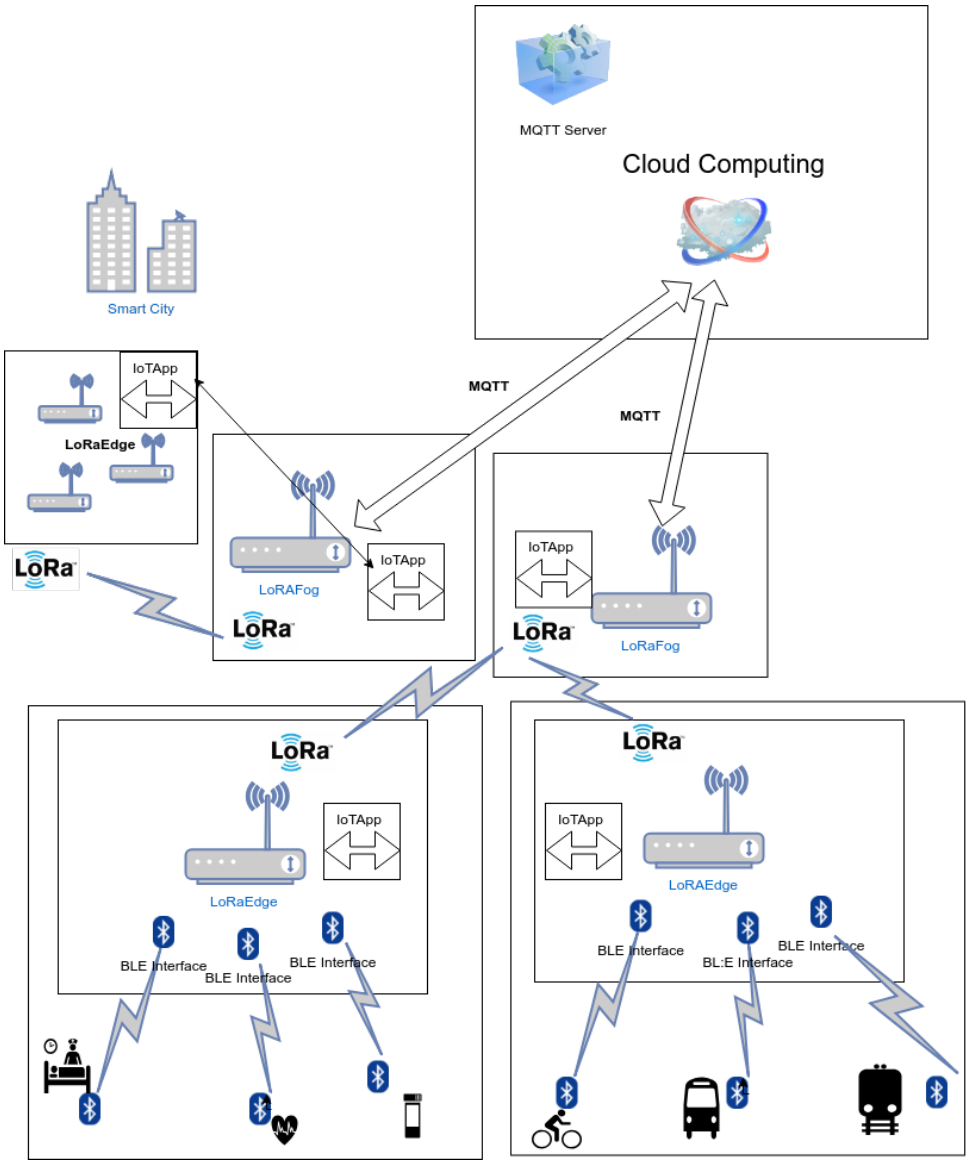


Figure 5.7: A hybrid LPWAN LoRa and WPAN BLE

according to the study, values below 0.95 means a saturation. Considering we can see in our graph that LoRa networks using a single 1 gateway and sending messages once have little communication loss, even for values above 8000 motes.

The Figure 5.3 is the result of simulation where all IoT nodes send messages to 1 gateway every 1 hour for 1 day. LoRaSim received the same parameters as the first simulation scenario for this simulation.

We can see that sending messages to the same gateway every 1 hour, and we have a significant increase in the number of collisions and a decrease in the DER compared to the previous scenario. Also, we already have DER values below 0.95, indicating a probability of saturation at values above 4000 nodes.

Figure 5.5, 5.6 are results of simulation using scenario multi-gateway, where all IoT nodes send messages to gateways every 1 hour for 1 day. LoRaSim received the

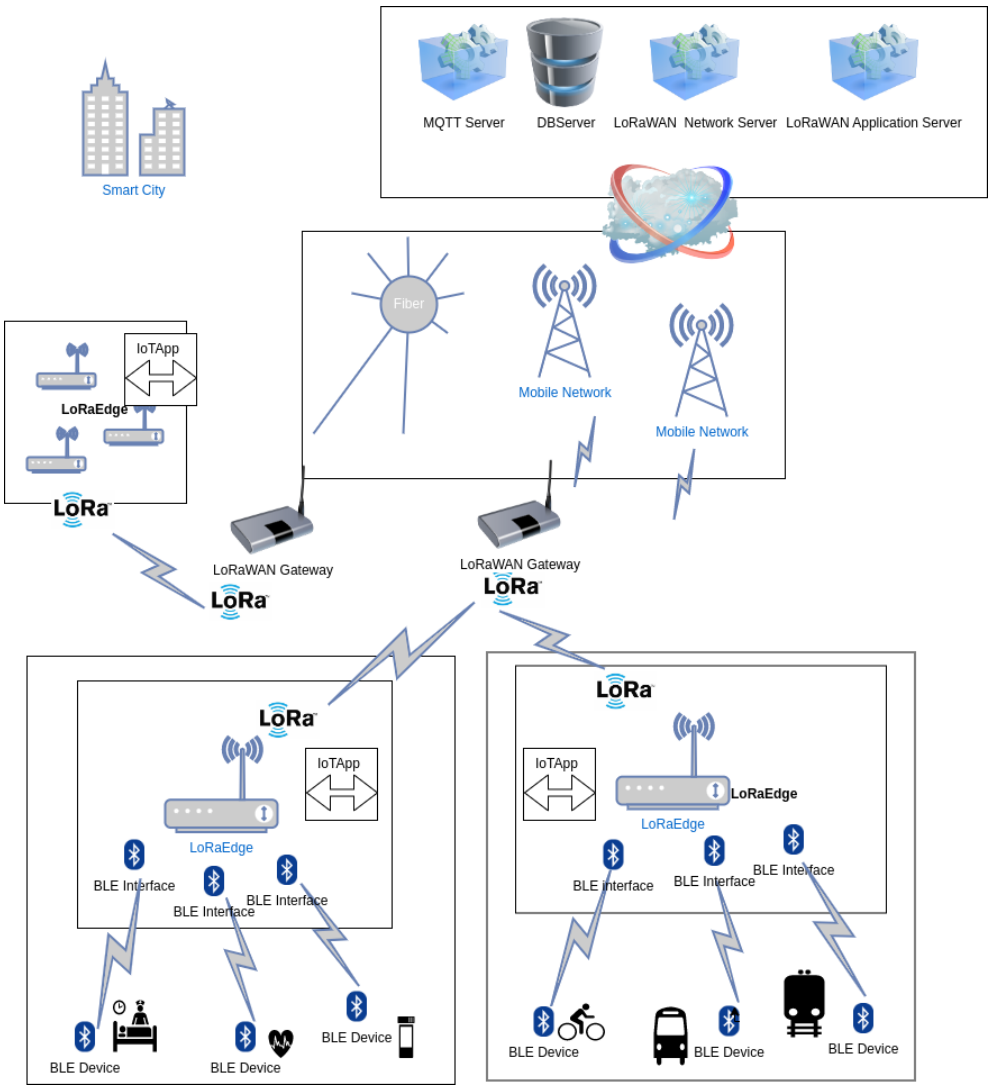


Figure 5.8: A hybrid LPWAN LoRaWAN and WPAN BLE

same parameters as the first and second simulation scenarios for this simulation.

We simulated sending messages in a multi-gateway scenario using 2 and 3 bases, receiving a message at most every 1 hour. We can observe in both scenarios that it is possible to DER above 0.95. This simulation results show that using multiple gateways sharing the load, it is possible to reach connection numbers of up to $20,000$ nodes for an application that sends a message within an hour, an adequate capacity for applications that send the status of their conditions for management information and monitoring, such as waste, water and sewage management.

One of the aspects that we can observe in communication scenarios with a base is that the proportion of the number of collisions per the number of motes in the network increases progressively. In the multi-base scenario, when adding more bases, this proportion decreases significantly.

Observing the growth of collisions in these single-gateway and multi-gateway simulation scenarios is relevant because, according to [116], an article that proposes the CARA (Collision Avoidance Resource Allocation for LoRaWAN networks) to improve network capacity by employing an intelligent algorithm for allocation of resources. One of the main limitations of LoRaWAN is the reduction of communication capacity and scalability due to collisions caused by the channel access mechanism using ALOHA, which leads to a degradation of the overall network performance. There are also capacity limitations due to duty cycles, which regulate the transmission times for each device and the number of collisions produced when using low Data Rates (DR). Coordinating resources across ALOHA channels is challenging to reach the maximum possible capacity.

Based on our observations, it is possible to perceive the potential of LoRa's range and scalability in environments with many devices communicating with SC. For it to have scalability, a scenario with multiple gateways is necessary, which would allow a reduction in the number of collisions and a reduction in message losses.

5.5 Conclusion

Regarding SC IoT App networks, we do not have a single technology or protocol that will cover all scenarios and use cases, each having its advantages and disadvantages in terms of range, cost, and energy consumption.

During our research, we could generate a potential in serving use cases that do not need bandwidth and frequent messaging, and we showed potential characteristics in this scenario.

In this chapter we show some challenges to be overcome when forming hybrid networks with LPWAN and WPAN, using low-power technologies LoRa and Bluetooth in SC application scenarios. These networks combine long-range and short-range

connectivity to maximize efficiency in a low-cost infrastructure compared to 5G networks that rely on different and interdependent technologies often controlled by stakeholders such as mobile network operators.

Recent propositions of this hybrid LoRa and BLE composition for IoT networks are already seen in the industry [117]. The New IoT module has also been able to leverage LoRa hardware to run the BLE physical layer, removing the need for multiple transceivers and allowing a very integrated and low-cost module.

We do not have a detailed analysis of energy consumption. The BLE device scanning phase of the LoRaEdge Algorithm, responsible for extracting information from edge devices, can be a power bottleneck. A network scenario with an excessive number of BLE edge devices and a high frequency of extracting messages from GATT can consume much energy. It should be the next focus of investigation in future works.

Using LoRa network simulation, we can see limits on the number of devices communicating at frequencies up to once an hour and up to once a day. In a future scenario, possibly when LoRa reaches Wi-Fi popularity, we may experience premature saturation, provoking discussions around some strategy or rules for sharing this network spectrum.

A potential scenario to be investigated is vehicular applications, which require technologies that enable mobility and geolocation. In this new scenario, a GPS integrated into LoRaEdge would allow the prototyping of these applications and would use LoRa to send the vehicle telemetry information and driver data.

In a future study, we apply the implementation of semantic queries, making LoRaFog devices a data query source for BLE devices via virtual sensors [118].

Chapter 6

Smart City and IoT Scenaries

This chapter has part of its content published in the book Chapter [34], addressing discussions relevant to the adoption of Blockchain and SC, and presenting the challenges of use in IoT and Fog Computing scenarios. We present in its closing the testbed using Ethereum clients.

6.1 Why Blockchain for Smart Cities IoT Apps ?

The SC application classes that deal with sensitive data, are strong initial candidates to use Blockchain as a technology to support their routines. In a public city environment, many applications will transfer data from registrations and the personal characteristics of citizens. They may even be responsible for paying fees, entrance tickets, and transport vouchers. These applications for the use of the population could benefit from security when doing transactions in a signed form. When we bring this to the edge environment like IoT, the lack of standardization of devices and ignorance of the environment where they are deployed, the Blockchain in purposes like the one discussed in this authentication and authorization work become potential solutions.

The Smart Contracts are Native features of some Blockchain, such as Ethereum, which allow the automation of iteration routines by the pre-defined and immutable IoT on the network without human intervention, combined with the traceability and reliability of Blockchain. IoT devices can use Blockchain features such as security background, information storage, and logs. The messages are stored in immutable blocks, can be verified anywhere and anytime, and the transactions are auditable, having built strong security based on cryptography and signature, making the possibility of alteration or falsification remote [119].

Blockchain has characteristics that make it a potential tool for Smart Cities (SC) [36] Apps, and many of these Apps have scalability and security requirements, which make it an indispensable prerequisite [60].

- Decentralized infrastructure, in an SC Apps, the participants or devices of an application are not known; one of the characteristics of the Blockchain is that it has been designed for networks without the need for trust. Blockchain is a decentralized, Peer-to-Peer (P2P) public lager in which transactions are replicated and known among all nodes on the network. Blockchain allows the existence of an untrusted network, and the network nodes do not need central trusted intermediaries to exchange messages with each other. Consensus algorithms are used to verify transactions.

- Resilient and scalable, due to its decentralized and P2P characteristics, Blockchain because it is a P2P network, it is highly scalable and resilient to failures and interruptions, as it does not have a central point of fragility. Being an immutable and durable ledger, once the transactions are registered in the Blockchain, they cannot be changed or deleted. This feature inhibits or eliminates unwanted side effects of public policy applications, such as fraud and corruption.

- Auditable and secure, SC will often deal with node participants that are not trusted. On Blockchain, transactions are created using a private key and strong encryption until they are registered in the public lager, with transparency, security, and auditing being essential.

- Standalone, each Internet of Things (IoT) device in use in an SC Apps, has a Blockchain account to make autonomous transactions and interactions using calls to Smart Contracts on the Blockchain; this enables interactions between devices without needing a trusted third party [59].

Currently, most SC communication proposals use the centralized model, such as cloud computing, to provide urban services on demand. In a centralized model, there is always a need for a reliable intermediary; Blockchain does not need this central point, with the business rules of the applications previously agreed through Smart Contracts. Decentralization in SC Apps with Blockchain adds transparency, security, and privacy to routines requiring automation [38].

6.2 Blokchain Smart Contracts in a SC IoT Apps

Current SC and IoT networks and Apps are commonly used to monitor urban conditions, city traffic, and predictive maintenance. Such networks have specific automation requirements; For this, a proposal to increase these applications' transparency, security, and resilience is to use Blockchain and Smart Contracts [120].

Urban Services: Some potential urban services scenarios are candidates for using Smart Contracts to automate and decentralize such applications. Some ex-

amples are public transport, health clinics, disasters, tourist information, access to public buildings, requests for services, and documents [6].

Public transport may have Smart Contracts that automate the use of public passes by the population, according to consumption rules and transport credits, in addition to validating their use and adding credits. Public health applications can use identity validation rules and drug acquisition profiles, recording their use and the use of public services, such as hospitals and exams. Monitoring and disaster sensors can subject their measurements to contracts that can trigger emergency groups following pre-established criteria and rules for care. Public buildings can release access to rooms and auditoriums following the rules in the contract according to the user's profile. This rule would also be helpful for public events where the contacts could coordinate credit and voucher access and consumption. Public services such as garbage collection can use contracts to pre-establish days, limits, and requests for garbage cans, all defined by limits already established in the contracts. Public documents and personal identification can be validated and issued following the rules of Smart contracts, facilitating their reissues and access using, for example, decentralized storage without needing dedicated dispatch and validation centers.

Preventive Service, diagnostics and maintenance: Preventive actions can be initiated after detecting any non-confirmation by asset monitoring or problems with operational routines. Some triggers can be implemented in Smart Contracts to provide intelligent diagnostics. An example is the exchange of assets and their replacement by service providers. In this case, all participants have already been agreed upon for terms and values by a Smart Contract [121].

Traceability: Smart Contracts can provide traceability and history of each phase in a public service provision process in SC [122]. With insights from these histories, it is possible to identify user patterns of use and behavior based on mobility to adjust volume, identify places of critical service demands, and ensure the location of assets. For example, in an SC IoT Apps, it would be possible to know who, where, and how long used a service [123].

Authenticity: To be trusted, unknown IoT devices deployed in an SC need to prove their authenticity. Information such as manufacturer name, installation location, technical characteristics, manufacturer, date of manufacture, obsolescence, and upcoming maintenance is potentially stored on Blockchain. When used by applications via Smart Contracts, these data enable an accurate inventory and allocation of resources in the urban area, reducing and even eliminating certificates and physical documents. The authenticity of a device increases the security and reliability of this SC Apps, making it possible to mitigate fraud and tampering in public services. [88].

Chained Requests: Interdependent and chained requests between more than

one public service can use Blockchain and Smart Contracts for coordination. An example of this is public parking lots and tolls, which can request ambulance and medical assistance during accidents and check the availability of these services [124].

Quality and Reputation: The Smart Contract may have quality parameters when contracting services or requesting new assets. Parameters such as delivery time and ratings provided by service participants can create this basis in addition to establishing an acceptable level of quality and Service Level Agreement (SLA) in the delivery of these services [110]. An example would be an autonomous request from a device that identified a failure and requests equipment maintenance based on reputation and quality parameters among available suppliers.

Inventory Control: Characteristics of an IoT Device, such as models, manufacturing dates, and others, can be stored in Blockchain, allowing other SC Apps to access this data via Smart Contract to audit quality, maintenance, and even request purchases of new ones [121].

6.3 Why Use Blockchain for SC Communication Security ?

Centralized structures are one of the natural candidates and focal points for attacks and security breaches. Current applications in the operation of SC IoT platforms are based on a centralized infrastructure, usually in cloud computing. Using Blockchain and Smart Contract for history and automation of routines would overcome many problems associated with a centralized approach. One of the main reasons is that a Blockchain network is P2P and has no single point of failure or vulnerability[125]. The expected scalability for SC Apps and the heterogeneity of IoT devices lead to the belief that infrastructures and applications based on traditional centralized models will not produce desirable results, mainly for the following reasons [124]:

- A good part of IoT devices does not have the computational power necessary to perform cryptographic operations or even perform calculations or algorithms that require computational power;
- In an urban environment, in most SC IoT Apps scenarios, devices are physically exposed and can be attempted fraud and malicious users attacks;
- Supporting and remotely maintaining IoT devices on an SC is not always possible as they may be in harsh environments or standby mode for energy savings.

The challenges of using IoT in an SC network become more evident when considering that these devices can be used in applications that collect personal data.

Moreover, control critical routines for the population such as water or energy distribution activation of components used in public safety and controls and road flows.

A centralized network architecture used by IoT applications has some classic security issues. The Denial of Service (DOS) attack is one of the most common attacks. This type of attack, if successful and coordinated, could result in service interruption by exploiting the weak central point of failure [126].

An SC infrastructure based on the traditional cloud or on-premises format creates infrastructure hubs for data storage. These data are often sensitive to users, such as health information, purchasing patterns, and behaviors. In these data storage strategies, the user also does not have complete control of how their data is used and by whom and is often managed by unreliable technical teams.

Despite the data legislation, centralized storage often does not give the transparency of how governance occurs, levels of responsibility, or traceability of these data. Some applications give data access to third parties with complete access to this data. This access leads to risks such as deletion and tampering without the user's explicit authorization. Many strategies based on centralized servers are not efficient enough to handle many communications from edge devices, typical of IoT solutions. This centralized approach has security and scalability issues and can impact the mass adoption of IoT solutions. Technologies such as Blockchain allow the formation of decentralized P2P networks used without a trusted intermediary.

SC IoT Apps using Blockchain as a background can store their data with reliability and scalability. In addition to static documents, transaction logs and historical records can use Blockchain. Asset trading routines such as Non-Fungible Tokens (NFT) can be used for trading and property transfer [127]. All encrypted and digitally signed.

These technologies integrated into the Blockchain, such as Smart Contracts, allowing the exploitation of automation features in IoT Apps using advanced security and encryption features. Smart Contrat, deployed with routines and automation of payments between applications and IoT devices, allows transactions and routines without human intervention or a centralized server, adding traceability, transparency, and reliability to the information transacted and stored.

The transactions that record immutable data on the Blockchain can be identified and verified anywhere and anytime. This feature of writing in the stone of decentralized ledgers such as Blockchain creates ground for the emergence of IoT Apps that can guarantee traceability and auditing, with the possibility of remote tampering with these transactions.

Immutable records mitigate fraud based on trusted information embedded by IoT devices and energy companies. A classic example would be an SC IoT Apps that provides power statistics based on users' consumption. This consumption can

be inspected and verified at any time.

By focusing on solving problems and proposing services to a city or government policy, SC IoT Apps need to be mainly transparent, secure, and traceable to allow citizens and public managers to verify costs, actions, and behavior of use of the applications. Some IoT devices that have a Long Life feature, that is, have a life expectancy above the standard obsolescence parameters. Many of these do not have an update cycle and periodic support without exposing them to vulnerabilities. [128]. This concern with vulnerabilities arising from lack of update is critical in devices that exchange goods or services and need trust, involving cost transactions. Blockchain requiring signature and advanced cryptography for each transaction can mitigate the risks between obsolete devices [129].

6.3.1 Blockchain and IoT, adoption Challenges

Bitcoin was the motivating project for creating Blockchain and its most famous user, being the most successful digital currency in the world. In short, it uses a distributed ledger to maintain transaction history in a P2P fashion across the network.

One of the foundations of Blockchain network security is consensus protocols; in this process, miners add blocks and fill them with transactions to ensure the integrity and state of the network and are rewarded with cryptocurrency for doing so. Bitcoin's initial approaches and algorithm are Proof of Work (PoW).

Therefore, the PoW approach has been criticized, mainly due to its consumption of energy necessary to generate a block [130]. Miners use the brute force search strategy to mine a block using PoW algorithms. As the problematic adjustments of this mining occur, more computing resources are needed for such resolution, which is inefficient, slow, and high energy-consuming [131].

The vital issue in adopting a consensus algorithm type is balancing security and efficiency. The most popular consensus algorithms currently seen in Blockchain networks still have many technical gaps, mostly related to performance problems in creating transactions [90].

In PoW, miners use much computational effort, and the Prove of Stake (PoS) algorithm is a much-discussed energy-saving alternative. Instead of requiring users to brute force find a nonce that satisfies the condition or challenge, PoS requires network participants to prove stake ownership of a certain amount of currency. PoS is based on the belief that this commitment of the participants inhibits sabotage of the network, as the losses are more significant than the gains. In this consensus, the creation of new Blocks and the writing of transactions are faster because they are familiar and rewarded by the nodes that participate in the stake without extreme computational effort [132].

The PoS critics consider this selection based on the account balance quite unfair because it would give the accounts with more coin dominance of the network. Compared to PoW, PoS has better energy efficiency. One of the recent cases is the Ethereum roadmap that gradually plans the migration to PoS.

Another approach is the Proof of Authority (PoA) consensus algorithm [133]. In Blockchain PoA networks, transactions and blocks are validated by previously authorized accounts known as validators that create and record transactions in blocks in an automated process. In the PoA, accounts based on reputation criteria merit becoming validators and receive incentives to maintain the achieved position.

As the search for a reputation generates rewards for an account, validators are encouraged to keep the transaction process intact and reputable, as they have losses associated with a negative reputation. PoA is considered more robust than PoS, as it only authorizes the creation of non-consecutive blocks by any validators, mitigating the risks of damage generated by a single authorizer. In the PoA model, the authority in this algorithm is based on the agreement between the parties. If one party is out of consensus, the other parties assume the assets and liabilities without affecting the network.

Miners must participate in the rewards in the Proof of Capacity (PoC) consensus algorithm. It is necessary to allocate ample hard disk space to mine a block [134]. Bitcoin created the concept of cryptocurrency in Blockchain, and Ethereum the Smart Contract. Smart Contracts allow the building of distributed applications using Blockchain's security and distribution features. The problems with running these applications on networks such as Ethereum are the GAS values for transactions. For these applications to become viable and famous, proposals for new and different consensus algorithms with less computational costs must be proposed. The work [135] presents a model for affordable transactions on the Blockchain.

Blockchain is a promising technology to solve some of the problems of an SC, but some gaps need to be addressed for massive adoption.

The Smart Contract gives Blockchain great power to build a new generation of decentralized applications and significantly impact how we consume and interact without needing a company, government, or financial institution to have central control of data and rules. Nevertheless, there are several technical challenges for this to be effectively feasible. Its scalability and ability to handle the volume of transactions are limited to the size and time of creating a new block. For example, bitcoin block size limits 1 MB being created every ten minutes, with 7 transactions per second. This feature would make it unfeasible, for example, to work as a data repository for an application that needs to publish data in streaming and has a high frequency of negotiations like some SC IoT Apps [89].

A solution to these limits would be to increase the size of the blocks, but this,

on the other hand, represents for the Blockchain network more storage space and more excellent propagation time in the network and an additional time for consensus formation between nodes, a classic problem of networks P2P [136].

Consensus algorithms like PoW and PoS face criticism. PoW consumes much energy, while PoS saves energy, centralizing decision-making power in the hands of a few rich accounts. So a balance between block size and security is necessary, which is the challenge.

Another critical problem for using Blockchain as a source of mass data storage is its lock time. For a new block to be created, we have a blocking time. This time is necessary for consensus and a new block to be inserted into the chain. This process ensures distributed ledger consistency [137].

For example, we have an Ethereum block time of 17 seconds, and Bitcoin has 10 minutes in PoW. This time prevents many applications such as SC IoT from being viable because there is no waiting tolerance in a transaction in some use cases. However, the security and robustness of Blockchain networks using PoW require this blocking time, and a shorter time would directly influence the decrease in security.

This lock time is determined by each consensus algorithm and is determined by the time it takes a node to confirm that the block is valid and insert it into the ledger. Moreover, for consistency, all nodes synchronize new blocks and their transactions.

Systems that perform communication with real-time communication requirements, where latencies cannot vary from 10–100ms are not applicable for storing data on Blockchain with consensus algorithms.

For instance, application execution calls using Ethereum Smart Contracts are made by submitting transactions to particular addresses. Considering this transaction's summed mining and replication times, we have a resulting unacceptable time for applications that rely on real-time. These times are not enough for Ethereum networks to be used as transaction repositories for SC IoT Apps that need real-time data from sensors [138].

Criticisms about the lack of regulation, the privacy of public transactions, and awareness of the Blockchain's limits affect the confidence of its adoption in other sectors outside Fintechs, such as SC Apps, in addition to discussions of legal issues and the actual applicability of Smart Contracts. Blockchain to become popular outside the cryptocurrency world, some of these challenges need to be resolved quickly.

6.3.2 The Fog Computing Blockchain and Smart Contract for IoT Scenarios

The business rules for IoT Apps using Blockchain are in the Smart Contract. The front-ends of these applications are client libraries that make API calls on the Blockchain [139].

Client API allows creating SC IoT Apps using new approaches. For instance, have each IoT node a Blockchain account, a transaction requesting access keys can be sent to a key manager informing the device account address. In possession of this key, the device and the application can make transactions on the Blockchain.

In the second case, the IoT device does not have an individualized account. In this case, the device acts as a verifier of the events created by Smart Contracts. It is passive and works as a sniffer, allowing a sensor to perform simple actions (e.g., turn on a relay) based on reading the values of variables (e.g., check relay variable values on Blockchain). This strategy can work with simplified security systems without storing or exposing critical private keys.

For these approaches to be a reality, some functionality and intelligence must be implemented in the devices. GoEthereum (GEth) implements the functions of a full node Ethereum Blockchain, with the entire protocol stack for managing blocks and transactions, monitoring, managing accounts, and mining, write em Go Language.

Some Blockchain API provide functions for signing and transmitting transactions for Blockchain, the javascript library for Ethereum, and web3.js [140]. Some functions, such as the transaction(), submit JSON-RPC transactions to Ethererum, and SignTransaction() signs.

In addition to creating transactions and interacting with the Blockchain, it is necessary to protect accounts and keys and allow signatures. The API available for accessing, for example, Ethereum, use JSON-RPC and one of the most popular libraries is web3.js. A feature of the applications that use these libraries is integrating Wallets. When taking these requirements to the world of Blockchain IoT Apps, it is necessary to propose unprecedented strategies that allow accounts to be managed by code without user interaction, and at the same time, guarantee security in this process [141, 142]. The main challenge in SC IoT Apps is avoiding private keys. A question still arises: What is the best way to keep the private key on IoT devices deployed in SC.

6.4 The Fog Computing Blockchain and Smart Contract for IoT Scenario

6.4.1 A Fog computing Blockchain

We set up a testbed using a private Ethereum Blockchain to investigate the use of Blockchain in a Fog Computing architecture. To evaluate Blockchain in Fog Computing, using an embedded device to be a JSON-RPC over HTTP API endpoint to receive the transactions that will be synchronized and mined on another server. We use GEth, a Free Ethereum client implemented in Go Language. It has full Ethereum features.

The main objective of this testbed was to understand the behavior and limits of using a Blockchain and its decentralized components in embedded hardware being used as a transaction gateway. This scenario is one of the possible architectures to be used in an SC that has IoT devices in applications that submit value transactions in private networks without direct contact with the outside world, Figure 6.1 describes the components testbed principals. An example of this application could be a ticket payment in public transport.

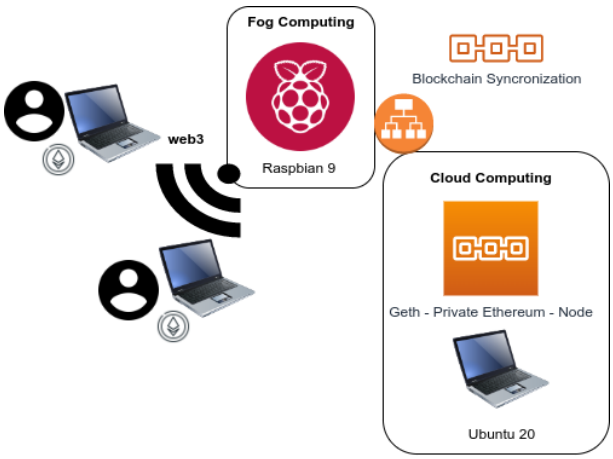


Figure 6.1: Testbed of a Ethereum Network in Fog Computing Scenario

We used as a payment gateway a Raspberry Pi 3 B+ with Cortex-A53 (ARMv8) 64-bit 1.4GHz and 1GB LPDDR2 SDRAM, installed with Raspbian 9. Version: 1.9.6-unstable - installed via snap. We set it up with a synchronization Ethereum node using GoEthereum (GEth) 1.9.6-unstable installed via snap, without the mining function due to sufficient computational resource limits to run the Ethash Proof of Work (PoW) consensus algorithm that we will use in the Blockchain network.

Representing the Ethereum cloud architecture, a mining instance was set up on another GEth server on an Intel I3 CPU desktop with 8 Gb of memory. The Listing 6.1 is used to start the private Ethereum network using PoW. The genesis

block difficulty has been set very low, $0x00001$ so that blocks are found quickly, a reasonable setup for private blockchains.

Listing 6.1: genesis.json

```
{
    "config": {
      "chainId": 12345,
      "homesteadBlock": 0,
      "eip150Block": 0,
      "eip155Block": 0,
      "eip158Block": 0,
      "byzantiumBlock": 0,
      "constantinopleBlock": 0,
      "petersburgBlock": 0,
      "istanbulBlock": 0,
      "berlinBlock": 0,
      "ethash": {}
    },
    "difficulty": "0x00001",
    "gasLimit": "8000000",
    "alloc": {
        "3590aca93338b0721966a8d0c96ebf2c4c87c544": {
"balance": "0x200000000000000000000000000000000000000000000000000000000000000000"
        },
        "8cc5a1a0802db41db826c2fcb72423744338dcb0": {
"balance": "0x200000000000000000000000000000000000000000000000000000000000000000"
        }

    }
}
```

To simulate IoT Ethereum transactions, we use a notebook with Ubuntu Linux using GEth, and as Peernode Ethereum, the Raspberry represents a device gateway in Fog Computing.

Raspberry, located in the Fog Computing area, acts as a payment gateway and interacts with both local and cloud network segments using its Wi-Fi and Ethernet interfaces. Private Ethereum allows users or devices on the local network to make new Ether transfer transactions between accounts. For this, we use the Metamask wallets to create these Ether transactions using notebooks connected to the WI-Fi network calling the Ethereum node HTTP RPC-API using the Raspberry Wi-Fi interface.

Below is the command to start the mining node and Raspberry node using GEth. In this setup are used two Ethereum accounts as an example; they are initing with ether in genesis.json and can be used as an ether base to receive mining rewards. This test account should never be used for production.

```
#Raspberry Pi Ethereum Node Fog Computing
geth init genesis.jason

geth --bootnodes "enode://$(minerEnode}@${minerIp}:30303"
--rpcapi "eth,web3,admin,personal,net"
--rpccorsdomain "*" --networkid 12345 --rpc --rpcaddr ${raspberryIp}
--rpcport 8545 --verbosity 4
--netrestrict ${localNet} --verbosity 3 --cache 2048

#Intel i3 Ethereum Miner Cloud Computing

geth init genesis.json

geth account import --password pass.txt private.txt

geth  --rpcapi "eth,web3,admin,personal,net,miner" --rpccorsdomain "*"
--networkid 12345 --rpc --rpcaddr ${minerIp}
--rpcport 8545 --verbosity 4 --netrestrict
${cloudNet} --nodiscover --mine --gasprice "0"
--etherbase "0x3590aca93338b0721966a8d0c96ebf2c4c87c544"
--miner.threads=1

#Test Accounts setup
0x8cc5a1a0802db41db826c2fcb72423744338dcb0
private.txt
df504d175ae63abf209bad9dda965310d99559620550e74521a6798a41215f46
pass.txt
pass

0x3590aca93338b0721966a8d0c96ebf2c4c87c544
private.txt
bc5b578e0dcb2dbf98dd6e5fe62cb5a28b84a55e15fc112d4ca88e1f62bd7c35
pass.txt
word
```

We made transactions by submitting Ether value inter-accounts using the Fog Node Raspberry Pi. Two scenarios were made collecting the time of duration of 10 transactions each, a scenario with a Metamask client making a transaction and one with two Metamask clients making the simultaneous transactions. In this scenario, we consider the latency of sending transactions to the Fog irrelevant because we are in a Wi-Fi network with high bandwidth capacity. The table 6.1 represents the average and standard deviation of the transaction times in the two scenarios.

It was possible to observe that the transaction log time is due to the mining time, consensus, and synchronization between the nodes. Even when generating simultaneous transactions from two customers, the transaction times observed are

	One Client	Two Clients
Avg	25 s	32 s
Sd	12 s	15 s

Table 6.1: Transaction Times

suitable for payment or validation applications, being very similar to the times of credit card transactions, but not suitable for applications that require real-time, as transactions were observed above 60s. The limits of this application are observed concerning the storage capacity of the Raspberry Pi, which, by having to synchronize the Blocks, can present problems as the number of blocks grows.

6.4.2 A Scenario using IoT and Smart Contract

SC IoT Apps that need a history of commands could use Blockchain as a point of contact and send change of state using Smart Contract, providing command history in a resilient, auditable, and immutable environment. Use cases where devices consult status and change them, such as opening doors and valves in sync, or traffic lights, for example, could record these actions in Blockchain through Smart Contracts. To interact IoT devices in a Blockchain, we propose a gateway that integrates with MQTT topics and Smart Contract deployed using Fog Computing architecture. This architectural proposal has potential because it is already found in the IoT industry with support for Publish-Subscribe (Pub-Sub). Pub/sub proposals make it possible to efficiently make available and collect data from multiple points simultaneously — using a message broker using the MQTT protocol.

This scenario follows the architecture Figure 6.2, composed of devices calling Smart contract using a gateway Ethereum MQTT that publishes and subscribes messages to topics and calls Smart Contracts. The IoT device, such as a Raspberry Pi and desktop Linux uses MQTT protocol to send and receive messages.

We do experiments using this proposal architecture and develop FogEthMQTT, a node.js daemon that publishes and subscribes to MQTT, invoking the Ethereum Smart Contract using the web3.js library, the code of the testbed is in [143].

Like cloud computing to go up an Ethereum Blockchain network with a bootstrap node and 2 miners using the Docker Compose script from the gethdev [144] on an I5 notebook and 8GB of RAM running Ubuntu 22 and Docker 20. The Smart Contract, conceptProof.sol write in Solidity [145] was deployed on this server. As MQTT was used, the open source Eclipse Mosquitto [146] was installed on the same server we used, FogEthMQTT.

We modified the genesis.json of [144] to start a network with PoW. Listing 6.1 is the genesis.

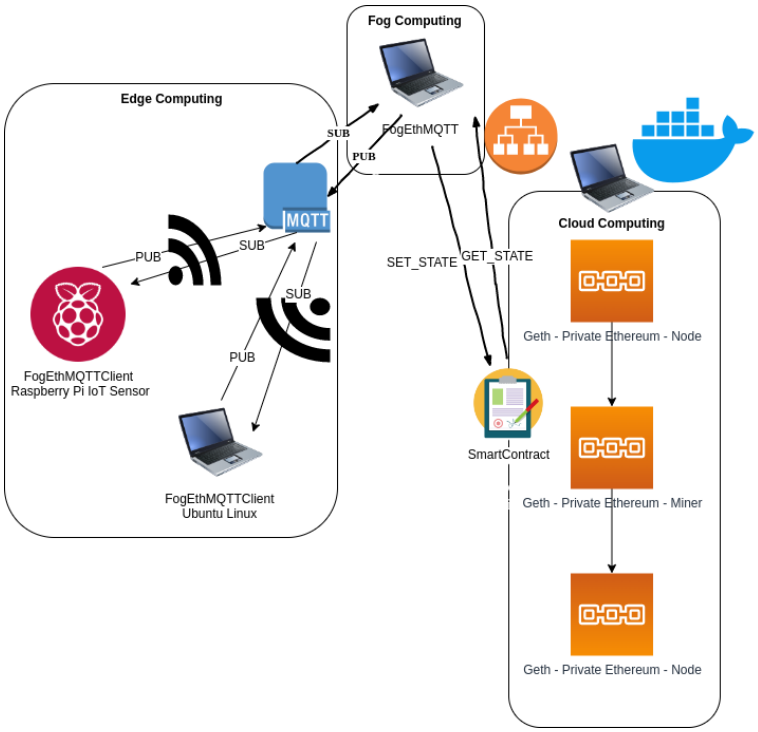


Figure 6.2: Fog Ethereum MQTT IoT network architecture

This scenario proposes a simple change of state of a device IoT, which can be useful for applications IoT that need to synchronize, such as opening doors of an event or opening and closing floodgates of waste. Devices send state change commands and receive and change their internal values using publish-subscribe using MQTT topics. The state change is done by FogEthMQTT, which receives commands from devices via topics, changes the variable's state using *set_state*, and gets the state by calling *get_state* using the Smart Contract deployed in Ethereum. The Smart Contract was written in Solidity, and its code is the conceptProof.sol listed above.

```
pragma solidity ^0.5.1;

contract ConceptProof {
    uint8 private state;

    constructor() public {
        state =0;
    }

    event changeState(uint timeChanged);

    function set_mystate(uint8 _state) public payable {
        require (_state >= 0 && _state <=2);
        state = _state;
        emit changeState(now);
```

```
    }

    function get_mystate() public view returns(uint8){
        return state;
    }

    function get_time() public view returns(uint){
        uint time = now;
        return time;
    }

}
```

This scenario is an example of device agnostic use IoT using a decentralized infrastructure to store values and log history. This immutable schedule can be generated by *emit* calls in the Smart Contract that store them in the Ethereum log.

The limits of this application are linked to the capacity of simultaneous transactions sent by an Ethereum account from a single point, the FogEthMQTT. We use a single Ethereum management account to generate transactions that change states; for this, exposing your private key in the FogEthMQTT configuration was necessary. Depending on the application, this approach can be relevant in terms of security.

We experimented with up to 5 simultaneous client nodes running we node.js script *nodeClient/nodeMqtt.js* to pub and sub, submitting transactions in a private Ethereum with one and two miners. During the experiments, the nodes every 25s exchange their internal values pseudorandomly between the numbers 0 to 2. If it is a new value, it sends a command publishing in a topic MQTT. The averages of times found after 10 instances can be found in [147].

Figure 6.3 shows the times in seconds that a successful transaction takes to occur, considering the number of simultaneous nodes and the number of miners mining and writing transactions in the blocks.

We can observe that the time variations are very close to the values obtained in the experiment of the Session 6.4.1 seen in the table 6.1 since they use the same consensus algorithm and have the same difficulty setting as the genesis block.

Figure 6.4 and 6.5 show the number of transactions per minute written transaction changing the value on the Ethereum blocks in one and two miner scenarios, using simultaneous nodes.

Despite the pseudo-randomness of the value change requests, we can observe that there is a limit of 2 successful transactions per minute, even in scenarios with two miners.

A error occurs when a transaction is sent with a repeated nonce or out of order before the completion of the last one. Ethereum, to avoid replay attacks, needs

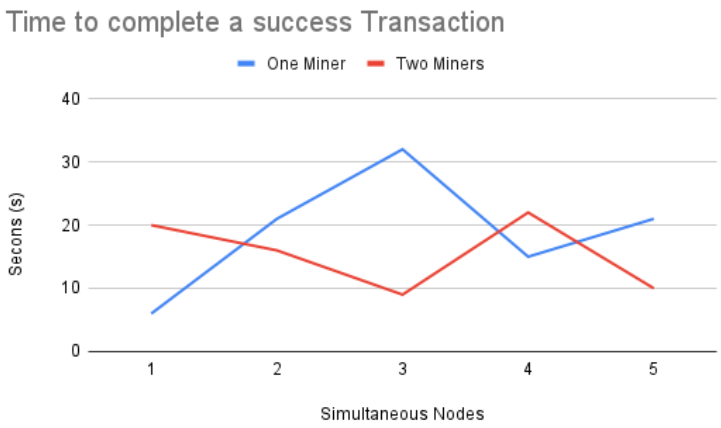


Figure 6.3: Time to Complete a Success Transaction

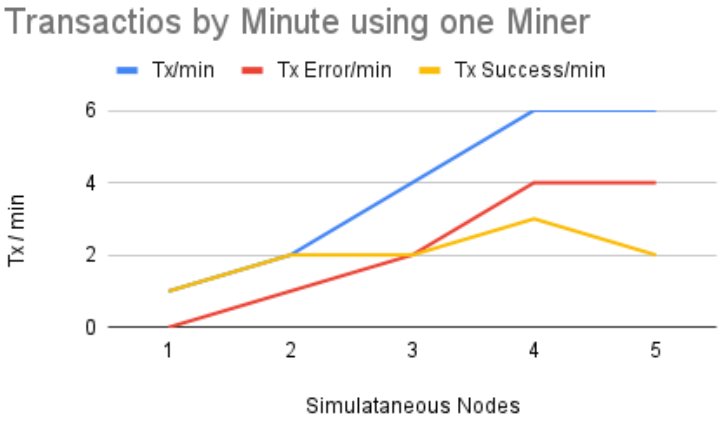


Figure 6.4: Transaction by Minute using on Miner

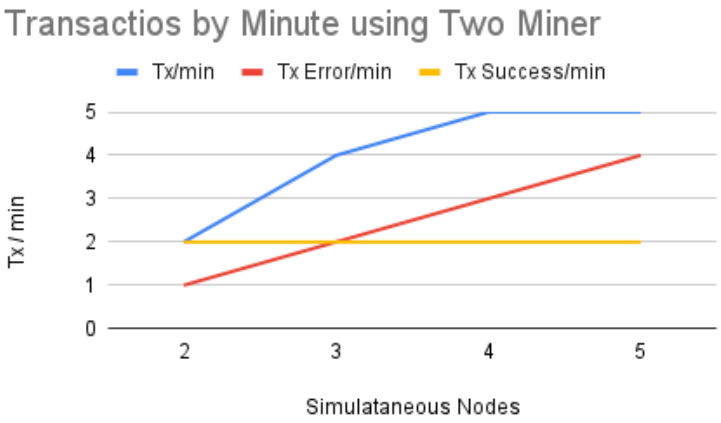


Figure 6.5: Transaction by Minute using Two Miner

respect to the transactions order control by the *nonce*. This limit on simultaneous transactions observed in the experiment is because we use a single Ethereum account to make transactions. Using more of an account to submit the transaction of a

central FogEthMQTT should mitigate this problem, but it should be investigated in future work.

This approach is an effective way of interacting with the Blockchain and using it as a source of registration and data, especially considering that it is not necessary to have any encryption, signature, or even processing of specific daemons on the Edge nodes. Compared with existing iussing a database or centralized , we have the advantage of Blockchain, which already has native support for resilience, immutability, and transaction history, avoiding, for example, fraud in altering historical data or even losing them due to central infrastructure failures.

6.5 Conclusion

This chapter discusses the scenarios and challenges of using Blockchain in SC and its possibilities when using Smart Contracts. We approached the possible Front Ends of an IoT solution with Blockchain and tested a scenario with Ethereum. It is possible to evaluate some limitations of using Blockchain when looking at the transaction times and information replication times between nodes. These observations follow the characteristics discussed and the differences between the consensus algorithms.

Chapter 7

Blockchain IoT Security In Smart Cities Apps

In this chapter, we address our investigation of ways to ensure and guarantee that messages from an Internet of Things (IoT) device come from reliable sources; this discussion is relevant because, in Smart Cities (SC), do not always possible know the details of what is used at the edges. This chapter has its content already published in the work [32].

7.1 Trusting in the data sources

We currently live in an urgent need to develop new and disruptive security solutions capable of supporting the long-awaited mass adoption of IoT devices promoted by the arrival of 5G. This IoT device is already widely available for purchase, using Application Programming Interface (API), protocols, and server infrastructure is often proprietary. An absence of standard and consensus among manufacturers leads to data exchange in an unstandardized network architecture.

IoT devices being set in urban areas as pipes, energy sources, sewage, waste bins, temperature, and the high mountain makes physical access to devices challenging. A simple battery change or firmware update in some places requires difficult operational effort, sometimes being impossible or costly to complete quickly. Based on this awaited scenario, it is impossible to use the security features provided by default; new layers and security proposals to exchange messages from outdated IoT devices and Apps are necessary. When extracting data from an urban area, the most common scenario is to use IoT devices to receive data from the most diverse sources and devices [6].

The SC is a fertile field for these applications. An urbanized space is used to obtain information and prevent and manage urban problems, with IoT being an

efficient way of data extraction [7]. An SC App had a significant percentage of data being characterized as sensitive; this makes security a prerequisite. The SC App is a strong candidate to be a pioneering use of mass IoT 5G connected, thus improving an unsafety design's costs and risks. This new IoT hardware is often unstandardized and unreliable. Its low cost and simplicity of configuration make scale management a challenge.

The security characteristics of the Blockchain allow for some of these challenges to be minimized. Blockchain allows robust security to receive data from participants that can often be unreliable. The possibility of writing a routine using Blockchain entities by Smart Contracts allows for developing new applications, such as the IoT devices, spread in an urban environment, like cities.

However, when evaluating IoT use cases, it is necessary to consider some of the limits and possibilities, proposing new connectivity layouts, frameworks, and consensus protocols. Blockchain has the properties and robust security characteristics to help in this challenge. However, it has achieved hype in recent years and has only been tested in applications restricted to financial markets, with cryptocurrencies being its biggest highlight.

The Blockchain has distributed infrastructure, and this makes it scalable and resilient to Distributed Denial of Service (DDOS) attack[148]. SC Apps' usual security is centralized security architecture, which is inefficient for unpredictable growth applications, focusing on attacks that are concentrated on a weak point, have low scalability and are ineffective in receiving an increasing number of simultaneous transactions.

The IoT security problems justify our research, and the possibilities for developing Blockchain-based applications SC are very promising. Countries like the United Arab Emirates, the US, and the UK, already use these in government and the public sector. For example, Dubai plans that all public services be Blockchain-based by 2020 [149].

A typical SC IoT App problem is registry devices for verifying the source of sensible urban data. The Blockchain characteristics make it possible to receive signed and identified payloads from unknown and untrustable IoT devices, often with outdated firmware. We hypothesize that using Fog computing strategies and Blockchain is possible to provide a reliable, robust, and decentralized security environment. It makes it possible to verify the origin and content of data from an IoT device.

In this work, we propose a model for identifying and registering an IoT device inserted in an SC App. We develop an API Gateway to verify the identity and authenticate sign messages received by IoT devices, using Blockchain and Smart Contracts.

This proposition of security use network paradigms Edge and Fog Computing.

These paradigms have potential in IoT implementations because they are efficient and economical, even with low data transfer rates. They are suitable for demands where the application's intelligence is close to the information-producing devices. The main idea is to send consolidated, signed, verified and identified messages to the endpoint API located in cloud or management network layers.

This is discussed, as a SC IoT App can benefit from Blockchain facing the conventional technologies. We have seen that by addressing a decentralized structure, some applications already reduce operating costs, reduce risk, and increase trust.

We develop API gateways **IoT Edge API Gateway** and **Blockchain API Gateway**. These API gateways run in Edge and Fog Computing, and they sign and verify the IoT message's authenticity, using Smart Contracts deployed in an Ethereum Blockchain. We present a testbed using real devices running the **IoT Edge API Gateway** to validate messages before sending them to a server running the project IoT Framework Engine [31]. The objective of the testbed is to represent a typical SC IoT App scenario of the message and device authentications. For our API Gateways, we use client test libraries, Ethereum Solidity Smart Contracts, and the IoT registration DApp from the IoT Device Management project [68],[30] as a base.

This chapter summarizes the following contributions:

- A local daemon running on IoT device which receives sensor messages, prepares and sends payloads that are used for future verification of authenticity;
- a API GAteway protects the application network, receiving payloads and verifying the sender's authenticity.
- we present scenarios of Blockchain and Smart Contract in SC Apps; and,
- experiments using testbed with real devices to send validated IoT payloads to a data management project as proof of concept of our API gateways.

7.2 Decentralized Management Security

Centralized security systems have a latent weakness in which we find user records, passwords, user access keys, and other artifacts. Even with auditing and governance rules, these centralized systems are not guaranteed to change data without the user's exclusive authorization.

Centralization exposes a single point of failure; the chance of an attack as DDOS succeeds increases, making these systems vulnerable to familiar and routine cyber attacks [126]. Centralized systems have unavoidable and unstable behavior when receiving them, making these Apps undesirable and intolerable.

Users' accesses and data are centralized and controlled by managers, the traditional security and infamous systems. Despite its systems having audit logs, they are not free from undesirable and unsolicited modifications or, many times, unauthorized by users. An A centralized security management system is often made by the managers and modified by them at their leisure. There are risks to privacy, as these centralized entities have no strict or guaranteed control over the use of private data, which is often confidential. Health information, shopping preferences, and behaviors stored on central servers are not guaranteed to have power over how user data are used or even by whom [150].

There is no explicit guarantee of privacy in a traditional centralized infrastructure. It is unclear who has access to or who is responsible for the data, and it is sometimes impossible to have reliable means to track the use of changes in the data. It remains for users of centralized services to trust management entities and their storage capacity and suitability. In some cases, third parties are responsible for processing and storing data and may be deleted or tampered with without explicit authorization [151].

Our proposal for a decentralized authentication and identification of payloads coming from the Edge points of large-scale networks, such as SC Networks, would allow added security during the receipt of payloads while using the decentralized and cryptography resources of Blockchain.

Decentralized security management of IoT sensors with Blockchain ensures greater credibility, more substantial transparency, and resilience to SC Apps' security, given the guarantee of a trust data source [152]. Thus, the Blockchain needs to be fed with characteristics that identify the edge IoT device as metadata. A strategy for verifying this previously registered metadata is to undertake a Merkle tree with them and store it in a Blockchain. The proof of metadata presence in a Merkle Tree is used for future verification of data validation.

We propose HTTP API Gateways in IoT Device, Edge Computing, Network Gateway, Fog Computing. A local HTTP API running in an IoT device receives the messages from an IoT device. The messages received in this local HTTP API are signed, and they send payloads to an HTTP endpoint that works as an API gateway. For verification and validation, the payload sent contains the sign message, the IoT firmware, the SC API web service address endpoint, and other metadata of interest that identify the data source's origin reliability, and truthfulness of data. On the delivery of the payload, the **Blockchain API Gateway** identifies the device, validates the payload, its signature, metadata by Merkle Tree proof, and firmware using Smart Contract on the Ethereum Blockchain. If the payload is validated and the device identified, the message is sent to the SC API web service address endpoint sent in the payload as metadata information.

7.3 SC APP Scenarios

A centralized SC network security architecture to IoT has several security limitations. An exposition to Denial of Service (DDOS) attacks is a typical example, where a single point of attack on centralized servers results in operation system failure. Regarding privacy, data stored on centralized servers have private information from users regarding health, purchasing preferences, and behaviors, as there is no guarantee of control over how user data is used or even by whom. Data stored in centralized infrastructure are often not explicitly responsible or traceable [123].

Third parties are often responsible for processing the data, deleting, modifying, or tampering without explicit authorization from the user. The volume of data watered by SC Apps doubts whether centralized security servers will be efficient enough to handle the volume of end-to-end transactions, produced by example by IoT devices [153].

Without a distributed and decentralized platform that guarantees security and transparency and mainly traceability of transactions, such as Blockchain, an SC App that uses data collected to generate revenue to cities through fees, can be victims of fraud or manipulation. The exchange of goods and services needs to trust and make transactions involving costs or using third-party financial resources, requiring trust. These resources are mandatory for low risk and transparency to transactions [154, 155].

A decentralized infrastructure does not rely on other nodes, nor does it need a central authority or trusted intermediary to exchange messages. A blockchain is a potential tool for secure and scalable communications in SC, and it can make them quickly become a reality [36, 60]. It was created to use cases of cryptocurrencies and has already been widely used in today's Fintech structures. [129]. Resilient, the Blockchain has an immutable and durable record; the transaction is only complete after a node consensus, with an immense computational effacement to change or delete it is necessary. Additionally, a P2P network is highly scalable. Together Blockchain and Smart Contract have the requirements to create techniques that minimize the risks inserted when producing data in an SC App.

In use cases where citizens' IoT devices can collect data, a Blockchain solution becomes essential [121]. It is an ideal solution for human rights issues, such as personal and data privacy, transparency of the Public Power, citizenship, and security. SC IoT Apps can raise important data for sustainable urban development standards when considering the current development and digital transformation occurring during the COVID19 pandemic.

The IoT and 5G technology can accelerate the popularization of high transfer rate GPS car sensors, allowing for city traffic data to predict future traffic and congestions

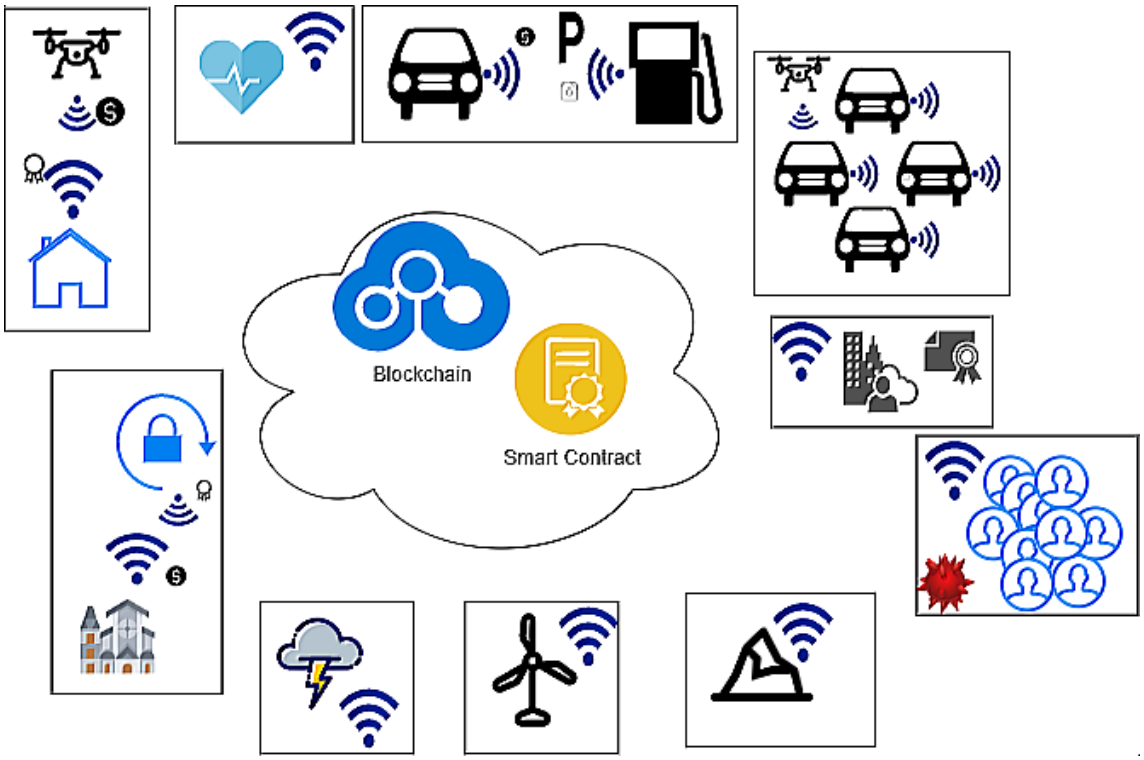


Figure 7.1: SC , Blockchain and IoT use cases.

[156]. Environmental data, including air quality, temperature, and rainfall, can help citizens or tourists avoid visiting a city point [157]. SC IoT devices, especially those placed in hard access locations, are its long-term use. Our API Gateway can support SC Apps that provide continuous and dynamic views of urban activities. We work to provide a more reliable architecture to receive IoT data while using Blockchain and Smart Contract for verification, even in scenarios where the IoT API is in obsolescence or that have firmware vulnerabilities [128]. Figure 7.1 presents the main IoT SC cases to use Blockchain and Smart Contracts.

The part of the actual SC App uses legacy HTTP API and centralized security management and can help to improve its security with this research.

The set of API gateways that we propose in this work can help in the problems that are described in SC Apps presented in Table 7.1.

Problems related to hardware and sensitive data are strong candidates to need extra layers of security. The proposed Blockchain scheme in this paper may be one of the alternatives. Its set of security features and decentralization features make it a key technology for the solution.

Outdated firmware on a device can cause an IoT device to be subject to several operational and security problems. Security weaknesses in these devices can cause the leakage of sensitive data or even operational unavailability. Falsifying data from these devices can cause credibility and financial damage to city administrations. Our proposition using API gateways with Blockchain for SC Apps is relevant for

Table 7.1: List of Smart Cities (SC) Apps.

SC App	Problem	Solution
City Trafficy	Firmware update cycle depending on vehicle maintenance or depreciation date	Security layers that work independently of firmware security features
Air Quality	Physical access in high altitude places cause delays in updated firmware and battery changes	LoW-power devices and a security layers that work independently of firmware security features
Temperature	Physical access in high altitude places cause delays in updated firmware and battery changes	LoW-power devices and a security layers that work independently of firmware security features
Analysis of Sewers	Physical access in hard access places cause delays in updated firmware and battery changes	Security layers that work independently of firmware security features
Rain Fall	Physical access in high altitude places cause delays in updated firmware and battery changes	LoW-power devices and a security layers that work independently of firmware security features
Tourism	Dissemination of tourist information and the price of false public attractions affecting the city's reputation, use of tourist data	Security layers that sign and verify the origin of messages
Public Health	False dissemination of citizen health data	Sign and verify the origin of messages
Public Services	False dissemination of citizen and service data	Sign and verify the origin of messages

avoiding message spoofing.

We use the characteristics of IoT devices to validate the payload. Extract the firmware and root hash from the Merkle tree created while using metadata for identification. The messages that are transmitted in the device payload are signed and validated in Ethereum by Smart Contract.

This strategy means that the previously registered characteristics are checked even if a device has outdated firmware. This routine is independent of the API and security features from the device. The falsification of messages is made more difficult by the need for these validations before receiving the message.

7.4 Materials and Methods

In this section, we describe the software and projects used to prove the concept of our work. We approach the details of the main functionalities of each project, technologies, and routines that are used to validate and identify the origin of messages sent from the devices.

7.4.1 API Gateways

A gateway that communicates and isolates the production API is one of the architectures currently considered best practices in designing a secure IoT network. These products can be found as API Gateways, and they are responsible for isolated environments and organized in separate internal and private business logic functions currently known as Microservice.

A popular solution component used to manage Microservices is API gateways management software, which completes tasks that allow developers to monitor, transform, and create security layers when exposed to a unique endpoint in their

internal production Microservices API.

We propose API gateways that identity, authentication, and the reputation of messages coming from the IoT devices of an edge network. These gateways would provide an additional security layer using Blockchain and Smart Contract. This set of technologies has implicit features for developing security in untrustable environments, such as unknown IoT devices that are spread in a SC APP network.

Our proposition differs from the already popular authentication tokens as Jason Web Token (JWT), adding Blockchain and Smart Contract in the background to verify the authenticity of the API Gateway's payloads before SC API receives them. API management is a recommended resource, mainly when various devices produce and consume data from different sources, due to SC characteristics and the diversity of software and API involved. In the next Section, we will present our API Gateways and the main interdependent components.

7.4.2 Components of Proposition

Our proposal of API Gateway to validate and identify the payloads that are received from IoT devices is composed of a web frontend for device registration, called **IoT Device Management**, an HTTP API on the IoT device, called **IoT Edge API Gateway**, and an HTTP API located at the border of the Fog or Cloud Network of an SC App, called **Blockchain API Gateway**.

IoT Device Management

IoT Device Management is the component responsible for registering devices in Blockchain using Smart Contracts. This component is presented in work [30], and the code is available online in [68]. The IoT Device Management was developed in NodeJs, React frontend ad, and the Smart Contracts are written in Solidity. Contracts are responsible for registering IoT devices and associating them with their owners called Entity. An Entity is an Ethereum account that is represented by its public address.

The list of Contracts that make up the basis of IoT Device Management are:

- Entitybase provides base functionalities for entities and is responsible for all entities and their public attributes.
- DeviceBase provides base functionalities for devices, responsible for associating with an Entity and creating the devices and their properties. Owner, identifier, metadataHash, and firmwareHash.
- DeviceHelper, provides extra functionalities for devices. The function isValidMetadataMember checks whether a provided item is a member of metadata

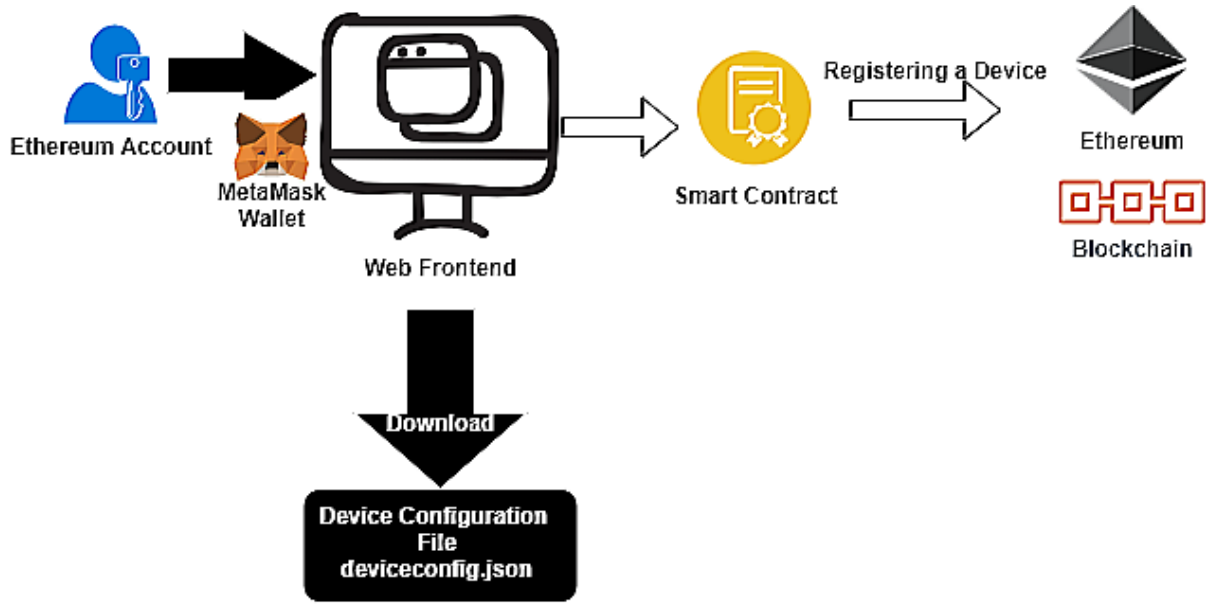


Figure 7.2: User interacfion using IoT Device Manager.

contained in the Merkle tree. The function isValidFirmwareHash, which checks whether a provided firmware hash is equal to the firmware hash device property. isValidEthMessage validates a previously signed message by **IoT Edge API Gateway** using an Ethereum private key.

- signatureBase is the base of functionalities for device signatures, creating and revoking a signature for a device.

Figure 7.2 represents a user's interaction with the IoT Device Manager to register a device in Ethereum. In the diagram, we can observe the Metamask [158] Crypto Wallet. It is an Ethereum account manager, and it is responsible for securing interaction with applications using web3.js [159] to call Smart Contracts.

During the device registration, the user must enter the following fields, the identifier, a set of metadata, and its firmware. The Firmware Hash and root hash of a Merkle tree containing the device's metadata are calculated and included in the Ethereum registration transaction, as in Figure 2.2. Finally, the Smart Contract that registers the new device is called. After its transaction is submitted and mined on the Ethereum network, the new registered IoT device's configuration file is downloaded, Figure 7.3. Figures 7.4–7.7 presents the sequence of a device registration in the IoT Device Management Frontend.

IoT Edge API Gateway

Developed in NodeJs language, an HTTP Deamon runs locally on the IoT device to receive messages extracted from local sensor data. The **IoT Edge API Gateway** in Figure 7.8 is a daemon running on the IoT device, responsible for signing the

{"identifier":"0x0f4c224a685da3e1d432f89acc548bc2777684ee",

"metadataHash":"507ea01c037cd1b6729395ab321169b334958fe1bfe558db1b2a085a692310dc",

"firmwareHash":"aaab56abb195b1bec63d541c2b63fa6fa7d6a00ed6a3f98e56354ff75745157a",

"metadata":[

"http://scapi/streams/ed4R34sd2312sde34edsflintind/data"

"Dow Town", "Sewer","Brasilia"],

"firmware":"....",

"address":"0x0f4c224a685da3e1d432f89acc548bc2777684ee",

"publicKey":".....",

"privateKey":"....",

"curve":"secp256k1",

"deviceId":1}

Figure 7.3: Device Configuration File.

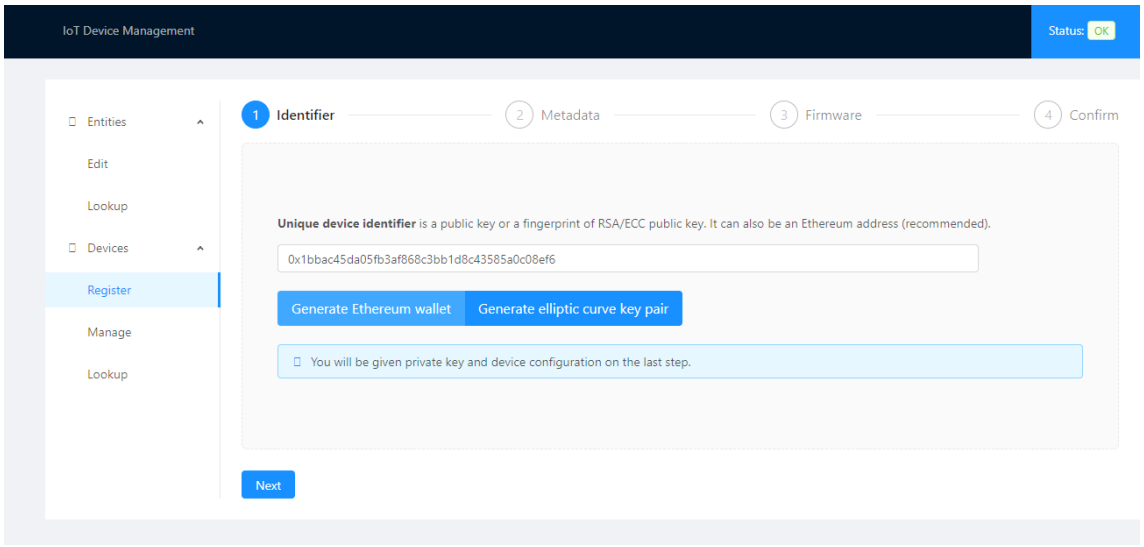


Figure 7.4: Identifier.

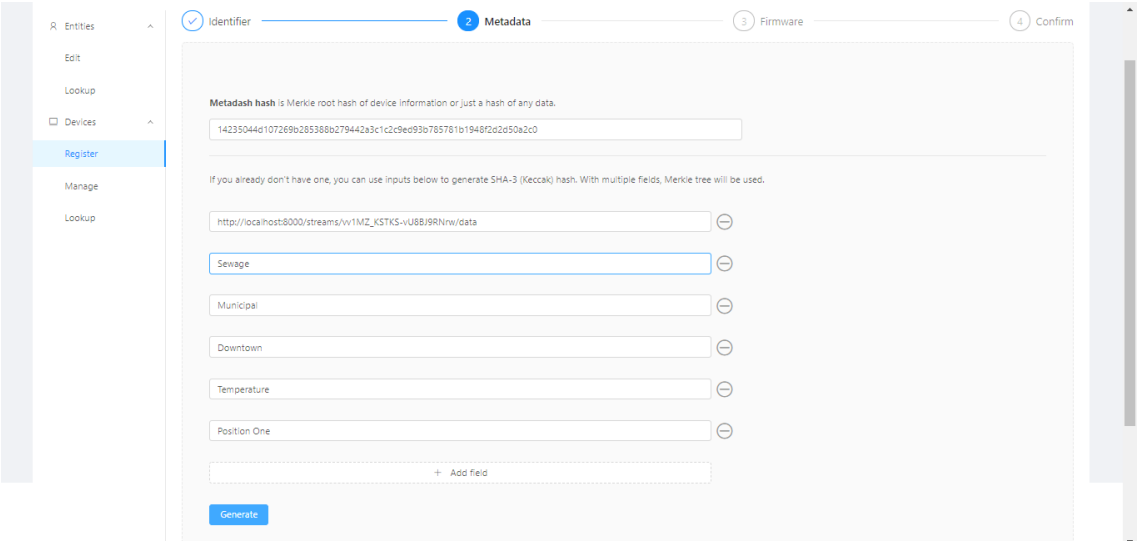


Figure 7.5: Metadata.

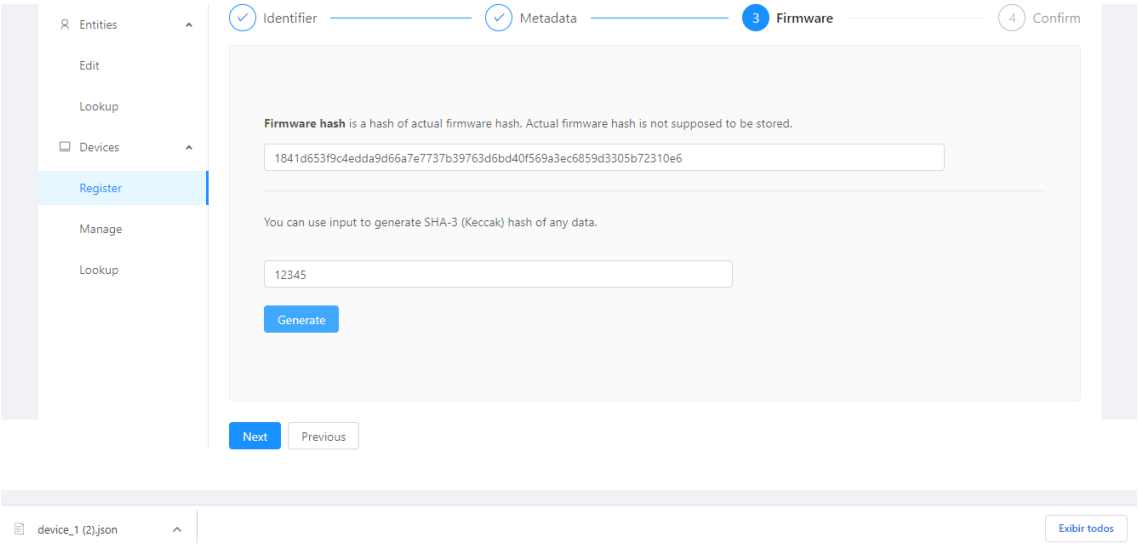


Figure 7.6: Firmware.

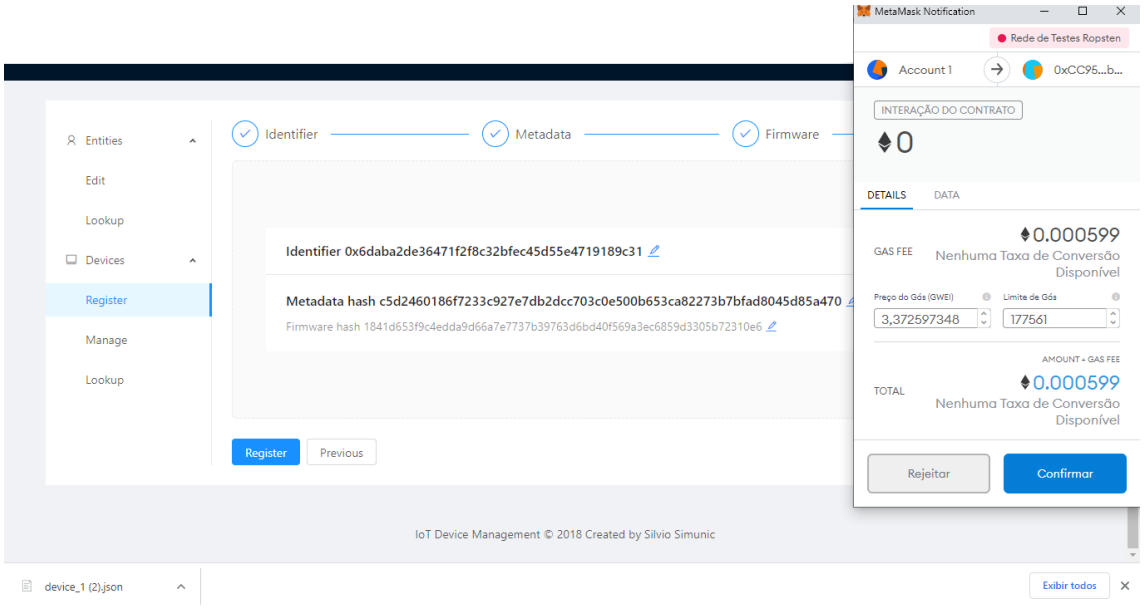


Figure 7.7: Blockchain Transaction.

message and composing the Payload with other attributes that prove the origin of the message and destines it to the **Blockchain API Gateway**. The keys and attributes of IoT Device Management device configuration are used to generate a payload. This Payload contains the device identification, the message, the signed message, the SC API HTTP address metadata, the Merkle proof, and the device's Firmware.

Blockchain API Gateway

The **Blockchain API Gateway**, as in Figure 7.8, is the component that is responsible for validating and identifying the payloads that arrive from IoT Edge Network devices. This typical component's localization is the border of the SC App Network. It is a NodeJs HTTP daemon that listens for payloads that arrive from the IoT Edge network. If the message of the Payload is validating, then it sends the message to the SC API. This API internal network address is the first metadata information created when it is registered in Ethereum.

The same Smart Contracts used by IoT Device Management are used to validate the **IoT Edge API Gateway** payloads. The **Blockchain API Gateway** uses the message, its signature, the metadata (HTTP address of SC API), its Merkle proof, and the Firmware received from Payload to validate. If validated, the message is forwarded to SC API, and the address is used as metadata. The **Blockchain API Gateway** and **IoT Edge API Gateway** are developed at node.js. The main libraries used are:

- *web3.js*, the base library for developing applications that make calls to Ethereum;

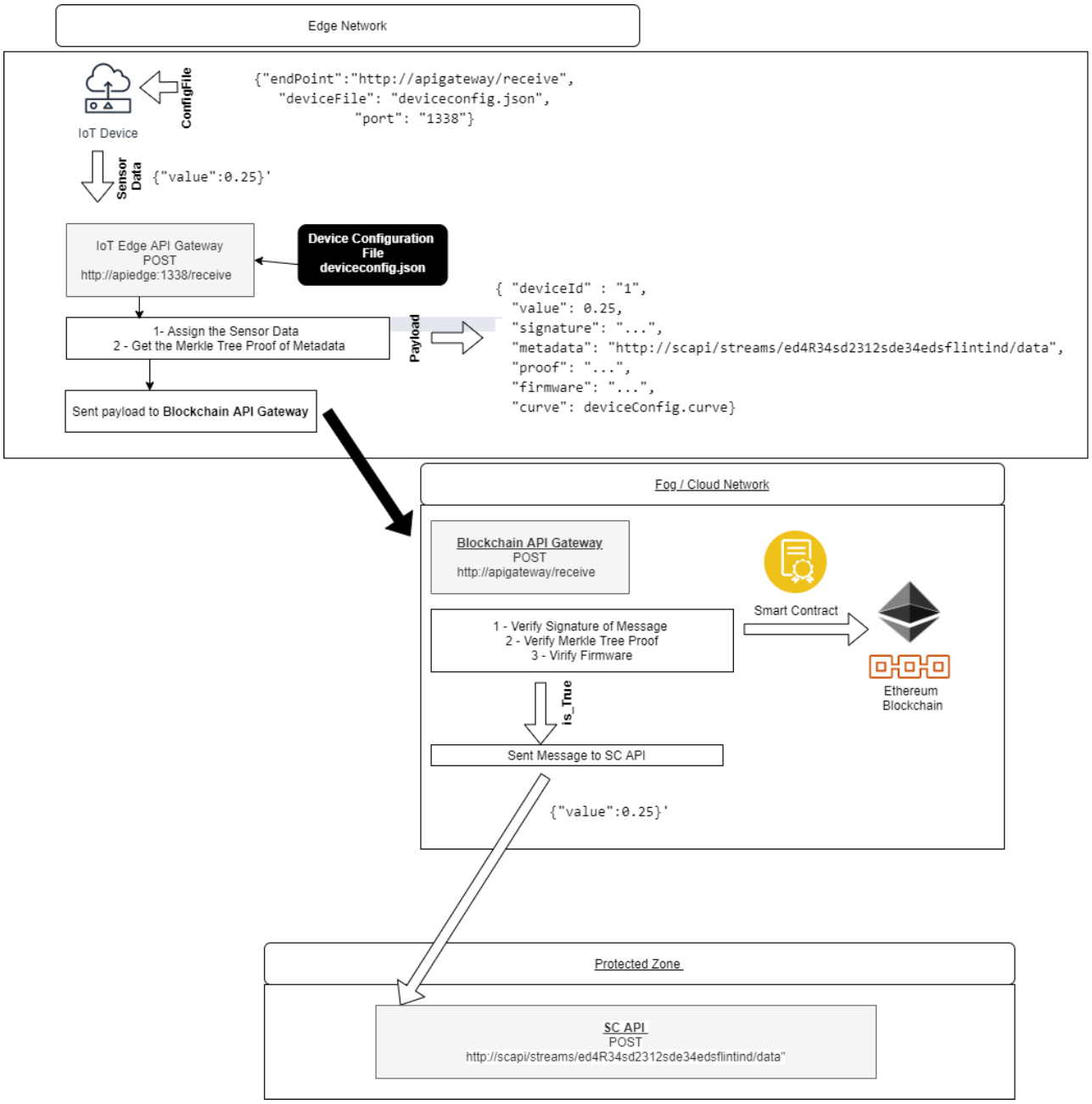


Figure 7.8: Blockchain API Gateway Diagram.

Table 7.2: List of atributes and parameters used in Testbed.

API Gateway	Type	SO	Hardware	Parameters	Network
Blockchain API Gateway	Docker Container, Pc	Linux	Intel	simultaneous transmiting nodes, average Time To Transaction, transactions Per Minute. average time to validation, validations per minute, CPU Average, Mem Average, time To New payload	FogNet
IoT Edge API Gateway	Docker Container, Raspberry, Ubuntu Linux	Raspbian, Linux	Intel, ARM	average time to transaction, transactions per minute, average time to assign, signatures per minute	EdgeNet
none	Docker Container, Raspberry, Ubuntu Linux	Raspbian, Linux	Intel, ARM	average time to transaction, transactions per minute	EdgeNet

and *ethereumjs-util*, acollection of utility functions for use with Ethereum

7.5 Experimental Testbed and Results

In order to see the feasibility and dynamics of the solution, we have implemented it as an experiment. This testbed simulates a Fog/Edge Computing network architecture using auxiliary components and a public Blockchain network.

7.5.1 Experimental Testbed

Our experiment uses a Docker Desktop on an Intel Pentium Silver N5000 1.10 GHz with 8 GB of RAM for network and Docker containers. Additionally, for experiments with a real IoT device, a Raspberry Pi 3 B +, with Broadcom BCM2837B0 Processor, Cortex-A53 ARMv8 64-bit SoC 1.4 GHz, 1 GB LPDDR2 SDRAM. Table 7.2 lists the attributes and parameters used in Testbed.

We use Docker and its resources for container management and virtual network to simulate an IoT network environment in contact with its SC API, an Edge/Fog Computing architecture. One of the tools available for the orchestration of containers and networks is Docker Compose. The *DockerCompose* file of the testbed is responsible for deploying three network subnets. The subnets FogNet, EdgeNet, and AppNet, have logical and network isolation.

The only container that has contact with IoT devices and SC API is the container running on **Blockchain API Gateway** in FogNet. The containers were deployed in the EdgeNet, with resources limited to 200 Mhz of CPU and 200 MB of RAM, running **IoT Edge API Gateway**, representing the IoT devices. In AppNet, we deploy SC API. As an SC API, we use the IoT-Framework Engine [160] application from work [31] and its IoT-Framework-Gui [161] frontend.

We chose this project to represent an SC API because its components were developed to scale. The core of its API is developed in Erlang, a platform that has

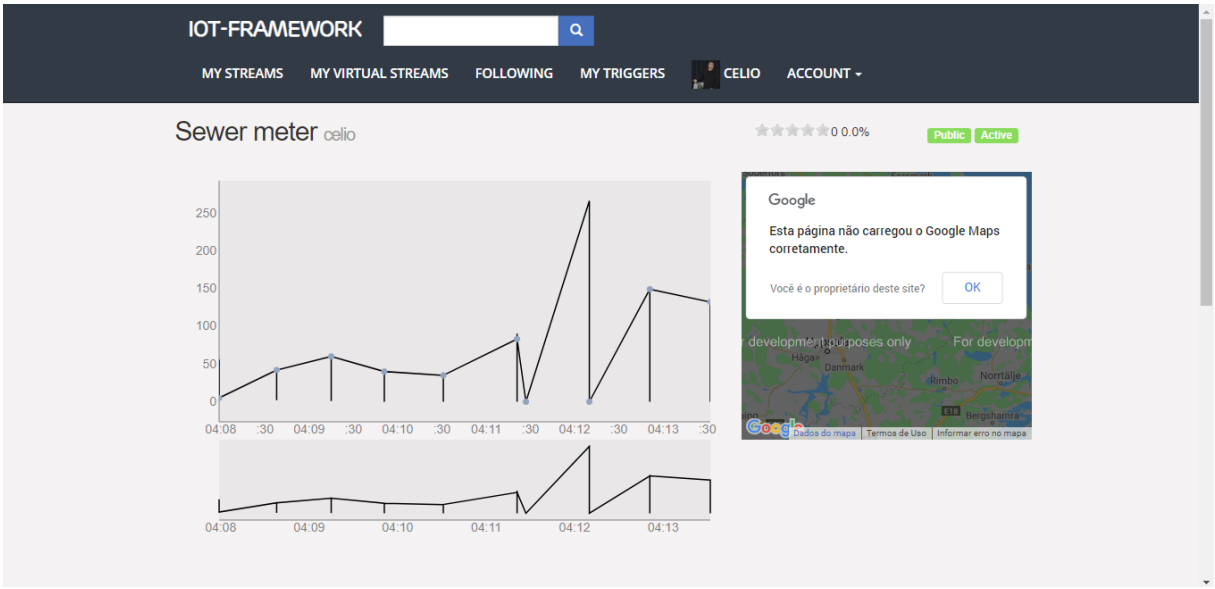


Figure 7.9: IoT-Framework-Gui.

shown promise in products that need to meet a large number of requirements.

Figure 7.9 shows its web interface, with a graph resulting from data collected from our testbed's IoT devices. In detail, a graph of density information in public sewage.

AppNet is an isolated network, and to contact it, EdgeNet always needs access to FogNet, which has contact with the internet, EdgeNet, and AppNet.

As Blockchain, we use one of Ethereum's public test networks, Ropsten. Ropsten's Blockchain is publicly accessible via the internet and has the same resources contained in Ethereum's main network. These Ethereum test networks help us to debug and test DApp and their Smart Contracts.

The DApp accesses Ropsten using the infura.io project [40]. Infura provides instant, scalable API to Ethereum networks. The Smart Contracts of IoT Device Management was deployed in Ropsten while using Truffle [162] to call Infura API.

Regarding the Web interface and registry of IoT devices, a version of the Web frontend of DApp IoT Device Management developed in React was deployed in Heroku. To registry and validate devices is called the Smart Contracts in Ropsten using an endpoint in Infura API. The codes of the testbed is in [163], and the frontend deployed of DApp IoT Device Management used in the testbed is in [164]. Figure 7.10 details the network components and Docker containers used for the testbed.

We use real Raspberry Pi 3 IoT devices, Docker containers, and Desktops for the testbed and install on these devices the **Iot Edge API Gateway**. On Raspberry, we use the Raspbian operating system. On Desktop and Containers, we use Ubuntu.

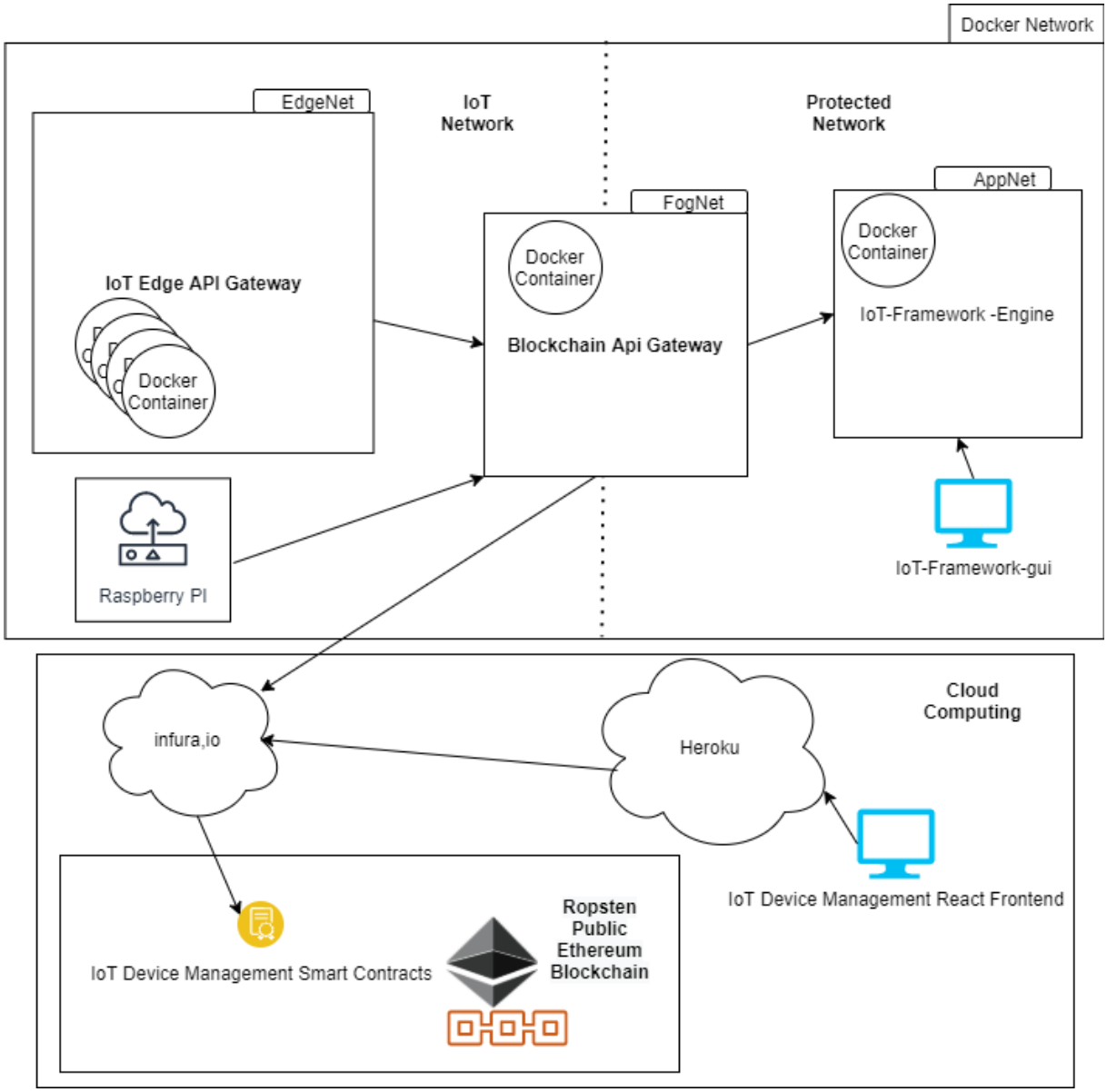
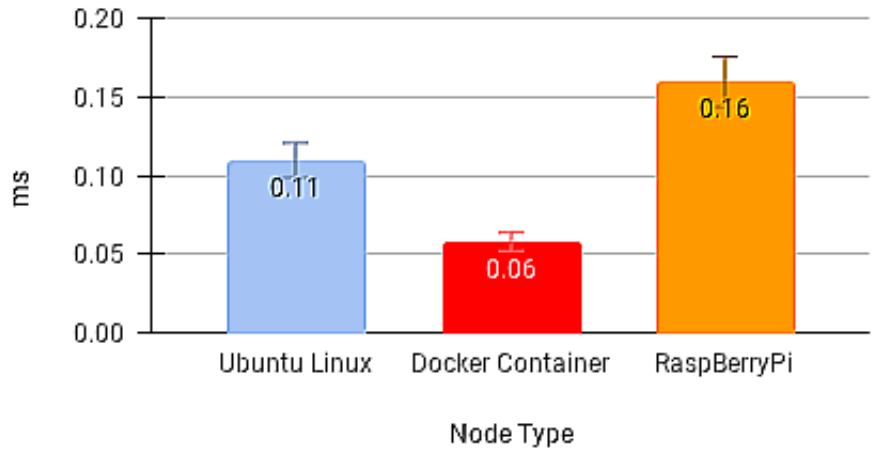
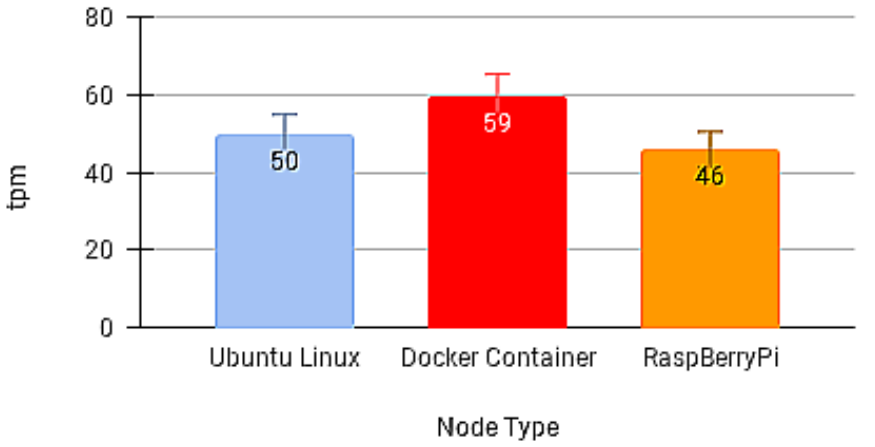


Figure 7.10: The Testbed network diagram.



(a) Transaction Time



(b) Transaction per Minute

Figure 7.11: Results without Blockchain API Gateway.

7.5.2 Results

We collected some experiments in scenarios using these devices for sending messages to understand the impact of times on the components of our API Gateways, on its layers, and the receipt of messages by the SC Apps in scenarios of multiple devices generating payloads.

Figure 7.11a,b present the transaction times for sending the message and the number of transactions per minute with no API Gateway.

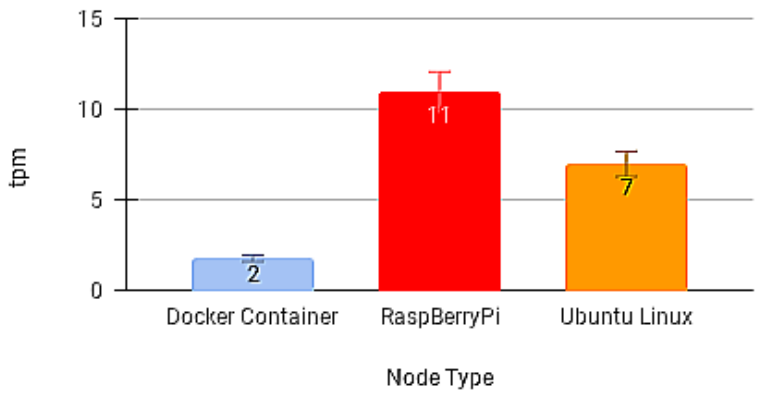
Figure 7.12b shows the average time for a transaction on the **IoT Edge API Gateway** when considering the shipping competition with other devices on the network. Figure 7.12d shows the average transaction time on the **IoT Edge API Gateway** by device type. Figure 7.12c shows the average time to **IoT Edge API Gateway** sign messages by technology. Figure 7.12a shows the number of transactions per minute achieved on devices using **IoT Edge API Gateway** to send

payloads to the **Blockchain API Gateway** and receive a response from the SC API.

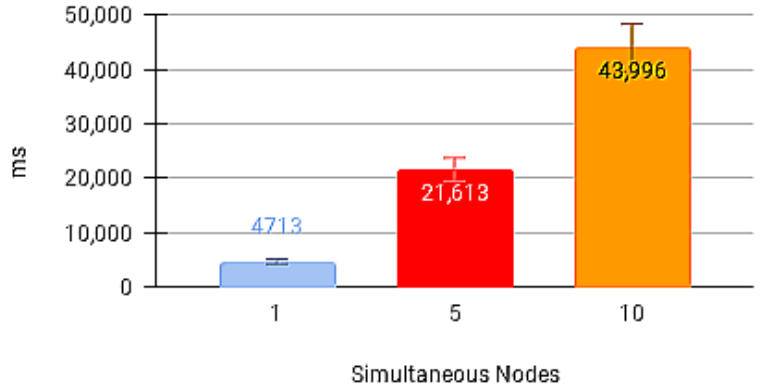
Figure 7.13a–c show the average payload transfer and validation times on the **Blockchain API Gateway** when considering the competition for sending with other devices on the network with one, five, and ten devices running the **IoT Edge API Gateway**. Figure 7.14a–c show the average of CPU use and Figure 7.15a–c memory use, during transactions in the **Blockchain API Gateway**.

The comparison of the times using the **Blockchain API Gateway** shown in Figure 7.12b, compared to the times without using the gateway shown in Figure 7.11a, illustrates the time to send the payload to SC API in the cloud. We observed an increase in time of 5 seconds for each node doing the validation routines, reaching almost a minute when we have 10 simultaneous nodes. These values are tolerable for applications SC IoT Apps that send a few packages during the day, such as those already discussed water pipes, energy sources, sewage, waste bins, temperature, and the high mountain monitoring avalanche. We also see that the message signature times are less relevant than the total time. The validation time as can be seen in Figures 7.13a 7.13b and 7.13c remains constant even in scenarios of sending frequencies every second. When many devices simultaneously use the **Blockchain API Gateway**, it is necessary to scale horizontally, putting more distributed gateways in load balancing and auto-scaling, thus increasing the capacity to leak these payloads to the target SC App API. CPU and memory usage in **Blockchain API Gateway** was not a limiting factor for the solution even for peak concurrent usage in our experiments which was 15 concurrent nodes, keeping at 30% CPU usage of 18 % of the total memory. The limiting values of **Blockchain API Gateway** are the number of concurrent transactions. The call times to validate payload using Ethereum Smart Contracts are similar to common Web service transactions and were not responsible for response delays. An important detail to remember is why transactions that require mining and writing of traces in the blocks have high response times, different from the call time that does not require mining and is not subject to synchronization session times. The devices we used for the **IoT Edge API Gateway** experiments presented similar transaction times and limits. The greater relevance of observing the behavior of nodes with computational power and Linux is one of the limitations of our approach, the need to have an operating system on the IoT device that supports node.js and web3.js.

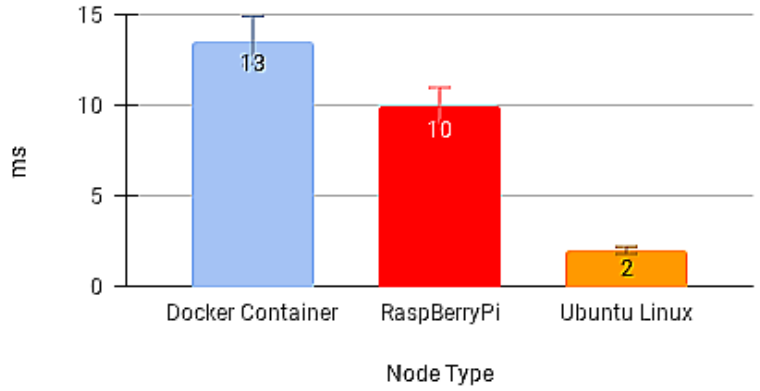
Based on our observations of this work, it is possible to perceive the potential of using Blockchains such as Ethereum as a decentralized security background, mainly considering its native characteristics and the possibility of executing routines during its transactions, the Smart Contracts. For example, it was possible to observe that Ethreurm's transaction times can be optimized when using options from other con-



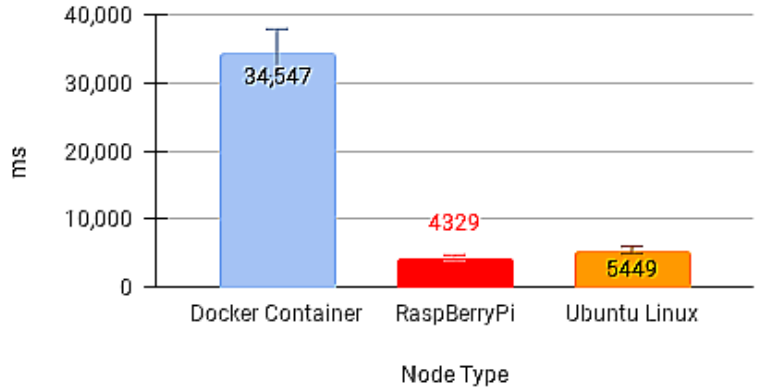
(a) Transaction per Minute



(b) Simultaneous Transaction

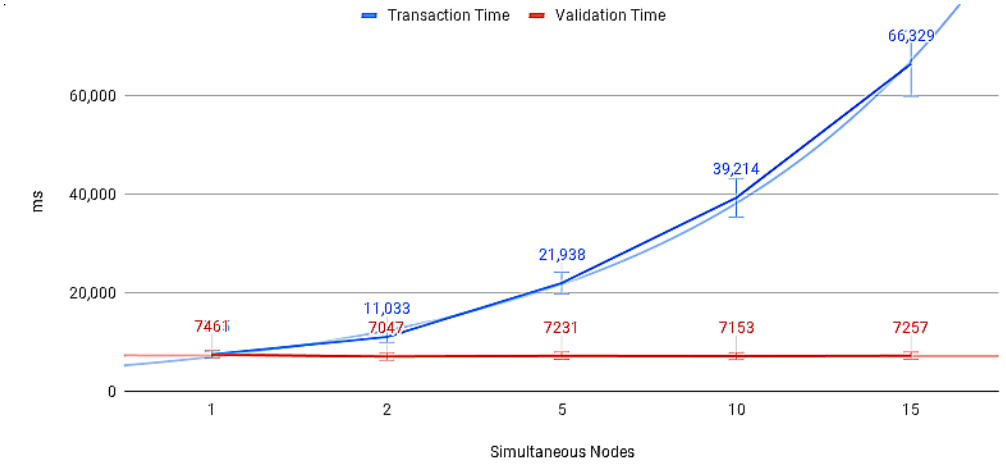


(c) Signature of a message

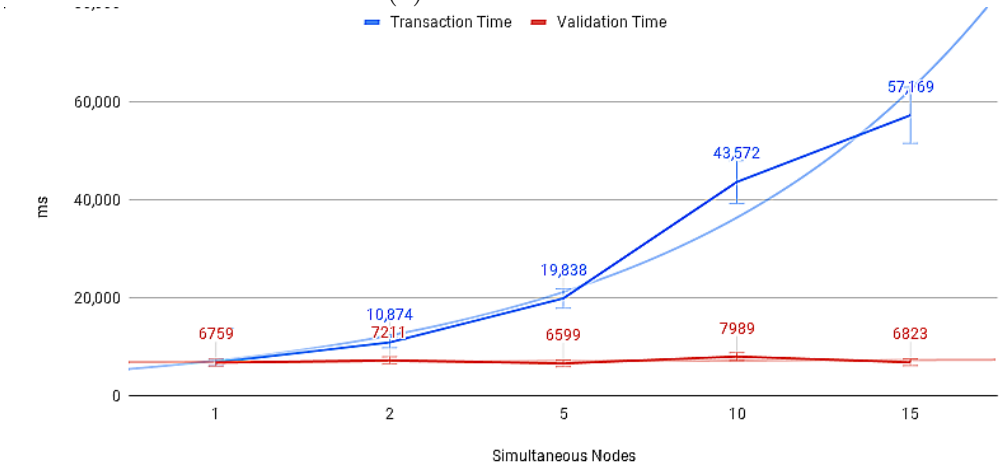


(d) Transaction by Node Type

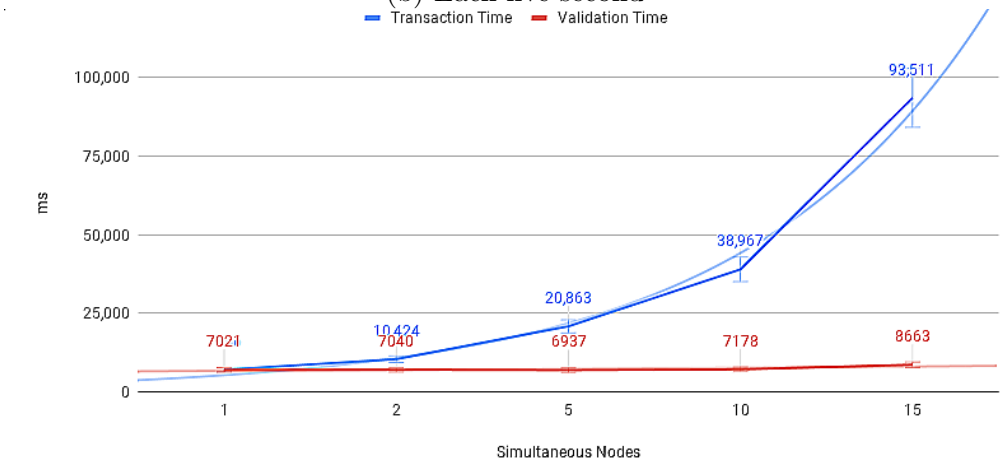
Figure 7.12: Average Time using the IoT Edge API Gateway.



(a) Each one second

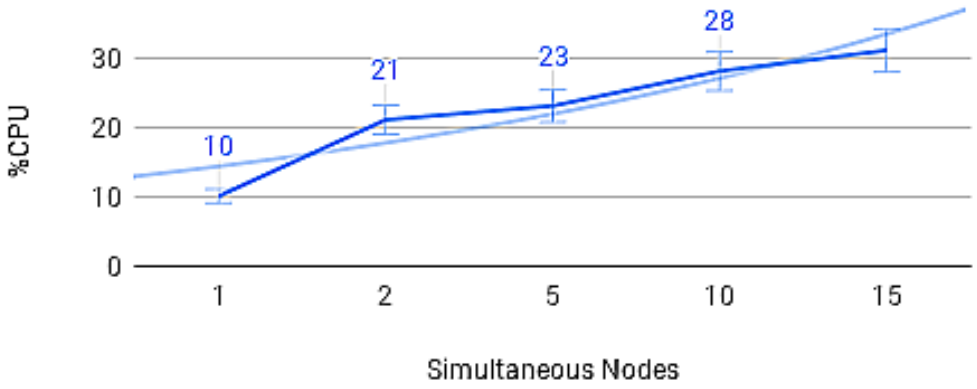


(b) Each five second

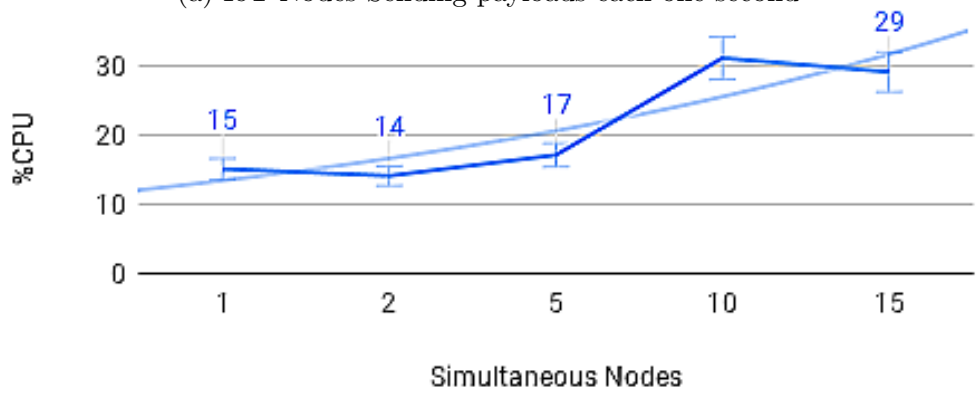


(c) Each ten second

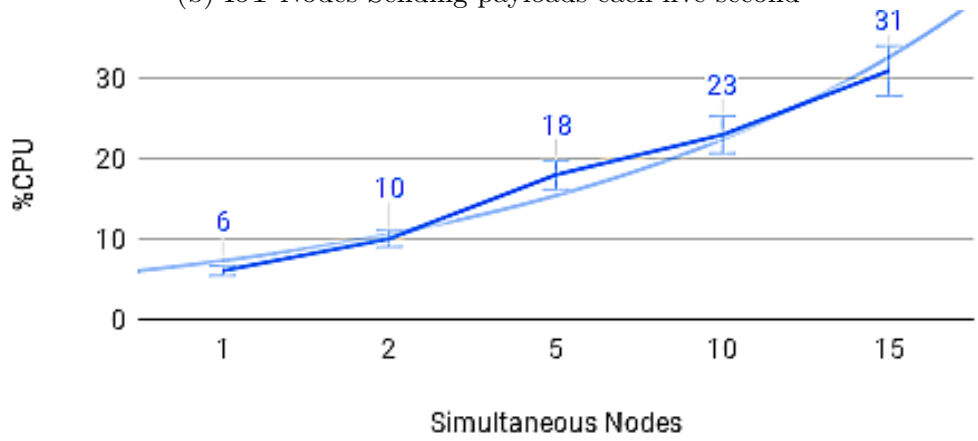
Figure 7.13: IoT nodes sending payloads.



(a) IoT Nodes Sending payloads each one second

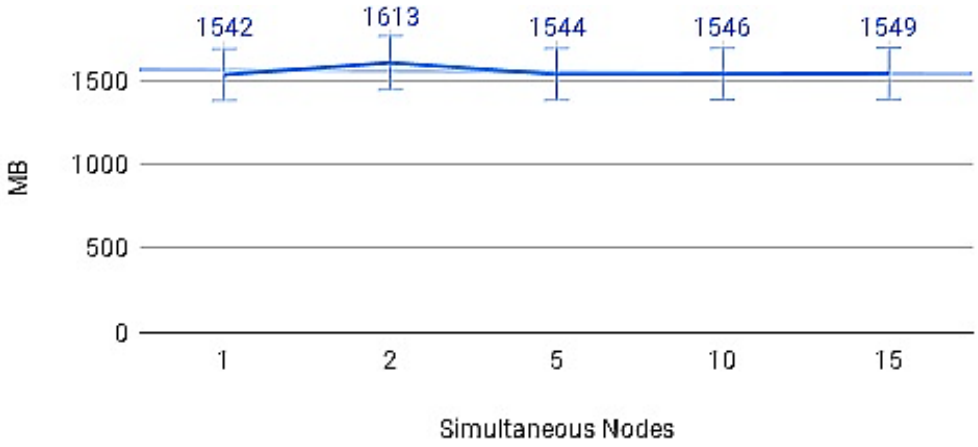


(b) IoT Nodes Sending payloads each five second

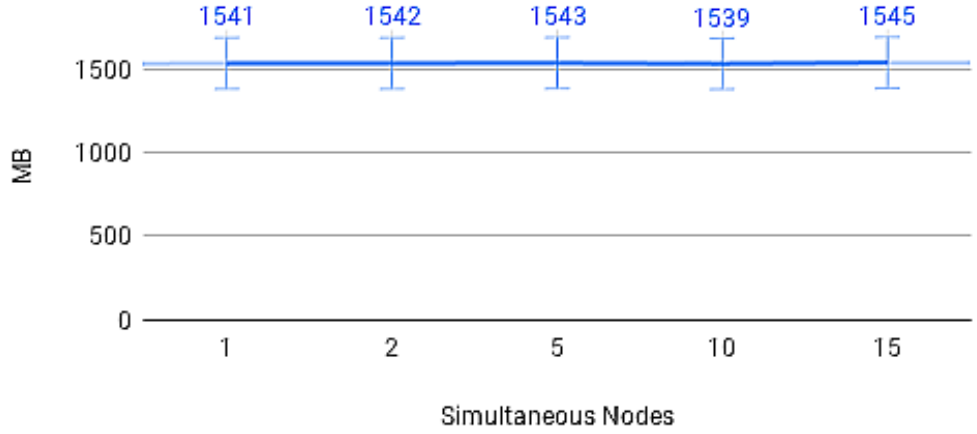


(c) IoT Nodes Sending payloads each ten second

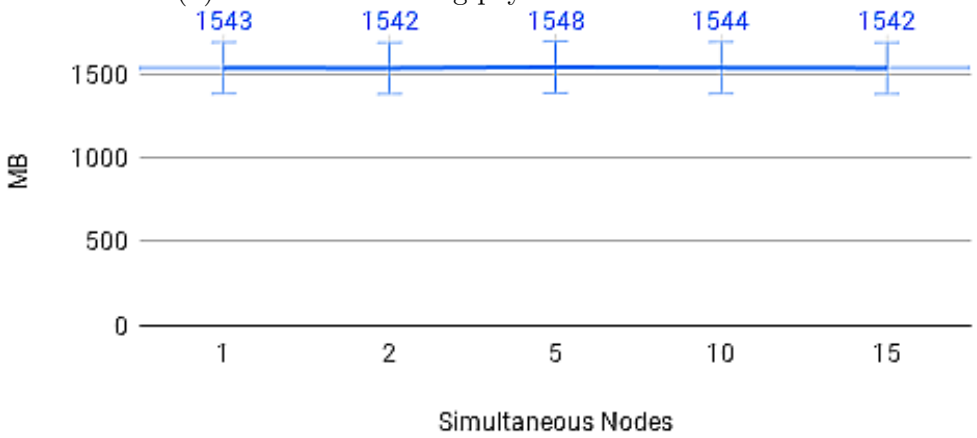
Figure 7.14: CPU usage.



(a) IoT Nodes Sending payloads each one second



(b) IoT Nodes Sending payloads each five second



(c) IoT Nodes Sending payloads each ten second

Figure 7.15: Memory usage.

sensus algorithms. This exchange is already being done even in the main Ethereum network that migrated to Prove of Stake (PoS) to be more energy efficient, increase the speed of transactions and reduce the cost; in the process, they called "The Merge." In our case study, for example, the routines of registering devices, and writing data to Ethereum, for example, would have benefits. Now, to be scalable in its decentralized phase represented by API Gateways, a scenario with multiple gateways is necessary, dividing the load and allowing a greater number of devices to be validated by the system.

7.6 Conclusion

Despite having SC Apps as a motivator, we can also apply our work to the vast majority of IoT use cases in Industry 4.0 that require a data origin guarantee.

We do not stress the possibilities of attacks in our scenario and thoroughly investigate the possible security holes in the implementation and architecture. Our API Gateways are an initial motivator for discussion to provide security and authenticity at the data source from the network's edge.

The routines for consulting the Blockchain have no significant interference in the transactions that depend on it to verify metadata's authenticity. Concluded that applications could use Smart Contracts that do not generate writing to the Blockchain without prejudice in performance.

The open-source Blockchain Ethereum is growing in popularity as a cryptocurrency platform for developing Smart Contracts. It has a good part of the attributes needed to develop DApp and projects like Interplanetary File System (IPFS). These projects aimed at decentralized development promise to change the paradigm of the next generation of applications in Web3. The tools developed by the community for Ethereum are continually changing. Furthermore, its integration with projects is natural due to the immense adherence of open source developer communities to the project. During our research, it was possible to use Ethereum's development networks, like Ropsten, enabling the experience of using a public Blockchain network for our implementation in the same molds of cryptocurrency applications.

Blockchain has not yet been extensively tested in non-crypto currency or financial scenarios. Use cases, like SC Apps, create a strong need for intense development of new and disruptive IoT Apps. This phenomenon creates an urgent need to investigate and extend many of the points covered in superficiality by this research. These investigations' importance becomes more relevant when there is a hypothesis that IoT Apps will be globally implemented in the coming years because the COVID-19 crisis causes technological acceleration. SC tools implemented with security models discussed in our research may help in the next health crises, providing reliable

information from institutions, validating data, and identifying their origins.

From our testbed's transaction times, we can conclude that, although we cannot carry out transactions in real-time, the solution has strong adherence in applications where the times for sending payloads via IoT have a frequency of hours or days.

In some SC Apps, the government agencies may not donate the device the citizens have acquired. A previous registration of this unknown device and the extraction of the hash of its firmware for validation can help to minimize these risks. It improves the risks for the use of devices with technical characteristics unknown and out of standard. It demonstrates the relevance of our deep in our research.

This security strategy can help to identify and validate the origin outside the IoT domain. For instance, the need to verify a news origin and identify the author is a common problem in the fight against fake news in current society.

We will investigate strategies for load balancing and automatic scheduling in future work, which would validate our proposal in a production environment.

One of our ideas for future works will be to investigate the possibility of integrating our solution with the API Gateway of the cloud industry, like Apigee, providing modules and plugins utilizing Blockchain as background.

Chapter 8

Extract Blockchain data using Semantic Web

This chapter discusses the consumption, linking, and use of data from Blockchain networks by external sources in a standardized way using Semantic Web and ontologies; the data is modeled with a graph allowing integration and linking with other pre-existing Web datasets. This chapter has a significant part of its content already published in the work [33].

8.1 Semantic Web and Ethereum Blockchain

The justifications for using Blockchain in the latest technological innovations are already far beyond Hype for adoption [165]. Have a wholly disruptive capacity due to being a platform known for strong data security, with the corruption of the data stored and threaded next to the impossibility.

The characteristics of the Blockchain allow the development of decentralized proposals for the domain of financial problems, the Internet of Money (IoM) Apps [166]. These decentralized platforms use encryption and digital signature in their transactions and use the security features provided by the network.

Initially proposed in the Bitcoin publication, the Blockchain has been used as a distributed database in a Peer-to-Peer (P2P) architecture, structured as a chain of blocks containing the transaction records and chained together using the previous block's hash as a reference. These transactions are created and signed by accounts of a cryptographic pair of public and private keys [167].

An advanced feature of transaction automation in Blockchain called Smart Contract allows the possibility to develop Decentralized Application (DApp) [36]. For the Smart Contract platform, we have the Blockchain Ethereum, also called World Computer, as it has recently become the preferred platform for DApp. The possibil-

ity of executing programs in Smart Contract format makes it potential for managing new cryptocurrencies or Tokens and Non-Fungible Tokens (NFT). Much is speculated even on the future overvaluation of Ether (ETH), its base currency, against Bitcoin, due to Ethereum's ability to be the platform for DApp that use Smart Contracts to automate their transactions [37].

This mention of Ethereum as a platform for a new application paradigm assumes that all components involved in the background are decentralized. But in reality, when integrating Blockchain with current Web applications, we have a centralized, decentralized hybrid architecture that justifies proposing techniques that allow the integration of the Ethereum dataset with other datasets, allowing a better integration and joint understanding with data from external entities.

One proposal to standardize models and integrate data is the Semantic Web. It proposes query data from the Web as a database, linking them with other ontologies represented by ontologies. Semantic Web technics enable standard access, integration, and query, expanding the possibilities of insights between different application domains.

Ontologies representing Ethereum give access to its Blockchain entities, such as accounts, Blocks, transactions, receipts, contracts, Tokens, NFT. For example, EthOn [168] is an ontology found in the literature to represent the main Ethereum entities using Resource Description Framework (RDF) and Ontology Web Language (OWL).

EthOn formalizes the concepts and terms of the Ethereum, Blockchain in OWL, describing the Ethereum objects as classes in ontology. It covers the major Blockchain and State Transition concepts as Blocks, Accounts, Transactions, Contract Messages, States, and State Transitions; and the network concepts, Blockchain, Node, Protocol Variant, forking, and Network.

Therefore, the Ethereum data represented by the RDF graph can be accessed using as a model the EthOn ontology. Moreover, it allows queries using tools like SPARQL and links to other external RDF graph databases found on the Web, allowing smart real-time insights.

This chapter presents our efforts in applying ontologies to the Ethereum network. Semantic Web and Ethereum Blockchain techniques provide data access standardization for new applications that use Ethereum and Smart Contracts as a tool. A recent example is the popularization of the new Blockchain Oracle services, which connect the Blockchain and external worlds. They are responsible for feeding Smart Contracts with external information coming from other domains to generate predefined actions in the Blockchain coordinated by Smart Contract. This alternative can solve the integration needs between the pre-existing datasets on the Web and the data in the Blockchain in production, allowing that new application that only

needs to consult the basic information of Ethereum or even the Oracles that feed Smart Contracts. These applications can use the data model in the RDF graph to make integrated query writing, such as SPARQL sentences.

The main contributions are:

- The EthExtras ontology adds components to EthOn, proposing some auxiliary classes that facilitate the understanding and relationship between Ethereum entities.
- A Web application that uses EthOn and EthExtras ontologies extract data from Ethereum in production and converts it into RDF, making these entities available as Universal Resource Identifier (URI) for visualization and query.

8.2 Ethereum Ontology

Ontologies represent objects and classes the organisms that represent a problem domain. In this research, we go deeper into the Ethereum platform, as it globally represents a Blockchain with Smart Contracts support, essential for the DApp responsible for the current disruptive DeFi, Tokens, and NFT projects.

In this session, we will cover the main points of our approach to using and extending the EthOn ontology with the EthExtras ontology. Figure 8.2 shows a global vision of our proposition.

8.2.1 EthOn Ontology

EthOn [168] is an ontology to be a semantical model representing Ethereum Blockchain and ecosystem. It has classes, objects, datatype properties, and annotation. Supports multiple inheritances; for instance, Block is a child class of State Transition and Block.

The complete specification covering the concepts, hierarchy classes, and object properties of EthOn can be found in [169]. EthOn taxonomy is constantly evolving, and its potential has not been explored yet. New restrictions, classes, and subclasses can contribute to new properties and links. To this task, we design a new ontology called EthExtras, extending or simplifying the EthOn ontology with extras Class and objects.

8.2.2 EthExtras Ontology

EthExtras is the name of our new ontology proposition to extend the EthOn ontology. Its format is in OWL and published in [170].

```
@prefix ethextras: <https://ethon.herokuapp.com/ethextras.owl#> .
@prefix ethon: <https://ethon.consensys.net/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://ethon.herokuapp.com/ropsten/transactions/0a20713e8289f56e4c94a47e3308610b16d1cd59fd4
3a7a9271a1ccdf33e7c8f> a ethon:Tx ;
    ethon:from
<https://ethon.herokuapp.com/ropsten/accounts/05377158a54dC8cC378c8ea4b7fB5fbF78D06d83> ;
    ethon:hasReceipt
<https://ethon.herokuapp.com/ropsten/receipts/0a20713e8289f56e4c94a47e3308610b16d1cd59fd43a7a
9271a1ccdf33e7c8f> ;
    ethon:msgPayload ""^^xsd:hexBinary ;
    ethon:to
<https://ethon.herokuapp.com/ropsten/accounts/2aaCF811aC1A60081EA39F7783c0D26c500871a8> ;
    ethon:txGasPrice 2000000000 ;
    ethon:txGasUsed 100000 ;
    ethon:txHash
"0a20713e8289f56e4c94a47e3308610b16d1cd59fd43a7a9271a1ccdf33e7c8f"^^xsd:hexBinary ;
    ethon:txIndex 13 ;
    ethon:txNonce 743941 ;
    ethon:txR
"9cc4486709569ed7ea38bf8e70652e437b624b35d1538d849fc95b2e56b827e0"^^xsd:hexBinary ;
    ethon:txS
"6cb982356dbce72dfd32711c5011613fe1cbe6f6dff9a76f650cc4c94af2c08c"^^xsd:hexBinary ;
    ethon:value 1 ;
    ethextras:blockNumber 4900105 ;
    ethextras:inBlock <https://ethon.herokuapp.com/ropsten/blocks/4900105> .
```

Figure 8.1: An endpoint in RDF of an Ethereum Receipt

EthExtras is an extension of EthOn, and the main idea is to cover the environment of Blockchain in Ethereum technologies, already proposing some links with external classes and datasets, an example of an external link, and a relationship with these. The Table 8.1 list the classes used in EthExtras.

For the design of the OWL format of EthExtras, we used Protégé [171], a project to develop ontologies from Stanford University. The Table 8.2 give details about the EthExtras class properties.

The representation of Blockchain Ethereum through these two ontologies makes it possible to generate a helpful model for consuming and providing data and usage in conjunction with other external data models exposing web endpoints. Figure 8.2 illustrates using the EthExtras and EthOn classes and properties. Figure 8.1 shows an endpoint of an Ethereum receipt modeled as RDF and ready to query.

8.3 Consuming Ethereum Data Using Semantic Web

Using the EthOn ontology and EthExtras, we built a middleware capable of generating soft real-time RDF graphs of Ethereum data. We base on the semantic adapter code from the IoT-Framework Engine project [31]. To Development, was used python with Flask [172], and the main libraries were RDFLib [173] for generating the graphs and Web3.py [79] for communicating with Ethereum. Our middleware

Classes used in EthExtras		
Class	**Description**	**Properties**
PlatformNet	A network of a Platform, Ropsten, for instance, is an Ethereum devel Network	hasGenesisBlock. hasName, netName
Platform	A Blockchain Platform, in this research is EThereum	hasCurrency, hasExternalReference, hasName, hasNet
Abi	The ABI of a Smart Contract	
Blockchain	The Blockchain	hasExternalReference. hasPlatform
Currency	The cryptocurrency of Platform in Ethereum is the Ether	hasCurrencyName, hasCurrencyPrefix
dbpedia:Blockchain	The dbpedia reference to Blockchain	
dbpedia:Etehereum	The dbpedia reference to Ethereum	
ethon:Account	An Ethereum Account	balance, hasOwner
ethon:Block	A Ehtereum Block	miner, mixHash, receiptsRoot, stateRoot, transactionsRoot
ethon:ContractAccount	The External Ethereum Account used as a reference to a Smart Contract calls	balance, hasABI
ethon:Tx	A Ethereum transaction	inBlock, belongstoBlock
ethon:TxReceipt	The receipt of a ethereum transaction	blockHash, contract, cumulativeGasUsed, from, gasUsed, inBlock, logsBloom, type, to, transactionHash, transactionIndex
ethon:Uncle	The uncle block it happens when more with one child block is created from a parent block	miner
foaf:Person	A abstract representation of a user	
*** dbpedia: http://dbpedia.org/resource/ , ethon: https://ethon.consensys.net , foaf: http://xmlns.com/foaf/0.1/**		

Table 8.1: Classes

Classes Properties of EthExtras	
Property	**Description**
balance	The property of classes *ethon: Account* and *ethon: Contract* , responsabile to represemnt a quantity of Ether in a Ethereum external Account or Contract
belongsToBlock	The number of block with belong the transaction *ethon:Tx*
blockHash	The Block Hash of the Block containing the *ethon:Tx* that originate a Receipt *ethon:TxReceipt*
contract	The property with returns a ethon:Contract of a *ethon:TxReceipt*
cumulativeGasUsed	The cumulative Gas used in transactions of a *ethon:txReceipt*
from	The ethon:Account responsabile to invocate the transaction with originate a *ethon:txReceipt*
gasUsed	The Gas used in transaction of a *ethon:txReceipt*
hasABI	The *Abi* of a contract *ethon:ContractAccount*
hasCurrency	The *Currency* used in the *Platform*
hasCurrencyName	The name of a *Currency*
hasCurrencyPrefix	The prefix of a *Currency*
hasExternalReference	The link with external dataset
hasName	The name of a Platform or *PlatformNet*
hasGenesisBlock	The Genesis block of a *PlatformNet*
hasNet	The *PlatformNet* of *Platform*
hasOwner	The name user of a *ethon:Account*
hasPlatform	The *Platform* of *Blockchain*
inBlock	The *ethon:Block* thatg belong a *ethon:Tx* or *ethon:TxReceipt*
logsBloom	The logsbloom of a *ethon:TxReceipt*, allows to filter the hash of each element that is in the *ethon:Block*
miner	The *ethon:Account* miner of a ethon:Block or ethon:Uncle
mixHash	The mixhash of a *ethon:Block* is a hash which, when combined with the nonce, proves computation effort
netName	The name of the platform network
receiptsRoot	hash of the root of the state trie, is the hash of the array of transaction receipts of a *ethon:Block*
stateRoot	hash of the root of the state trie, is the hash of the array of transaction of a *ethon:Block*
to	The ethon:Account destination of a *ethon:txReceipt*
transactionHash	The hash of the transaction ethon:Tx that originated the receipt ethon:TxReceipt
transactionIndex	The index of the transaction ethon:Tx that originated the receipt ethon:TxReceipt
transactionRoot	The root of f the transaction ethon:Tx that originated the receipt ethon:TxReceipt
type	Type of a ethon:TxReceipt
*** ethon: https://ethon.consensys.net**	

Table 8.2: Classes Properties

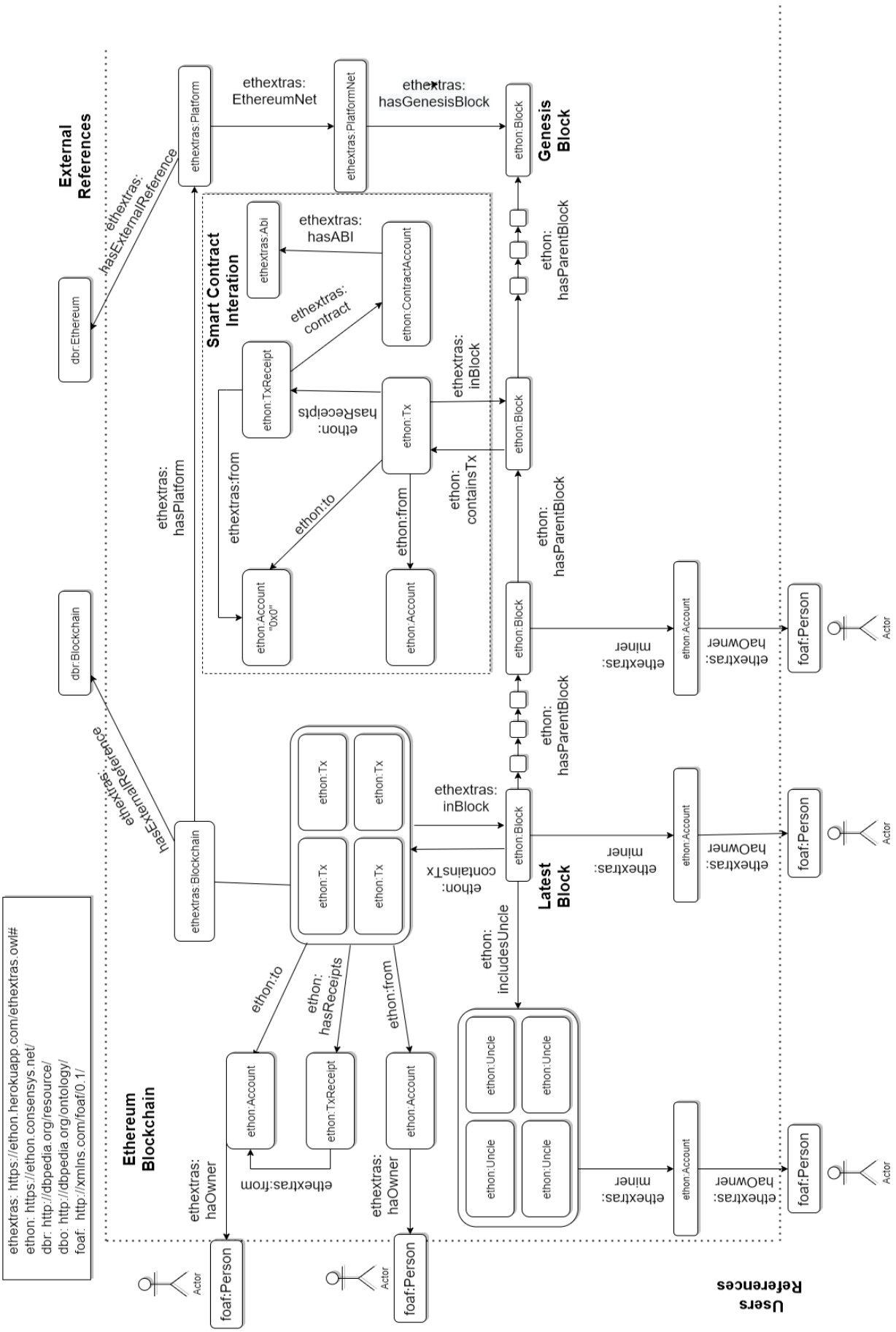


Figure 8.2: Ethon and EthonExtras Ontologies diagram of classes and using external references

Middleware Routes		
Route	**attributes**	**Description**
GET /		Its returns the index with examples
GET /ethNetwork	**ethNetwork** is a name of Ethreum network examples ropsten, mainnet	
GET /ethNetwork/accounts/id	**id** is a 42 character hexadecimal of an Ethereum address	This route returns an RDF containing the properties of an Ethereum Account
GET /ethNetwork/contracts/id	**id** is a 42 character hexadecimal of an Ethereum contract address	This route returns an RDF containing the properties of an Ethereum Contract Account
GET /ethNetwork/blocks/id	**id** is the number of a Ethereum Block	This route returns an RDF containing the properties of an Ethereum Block
GET /ethNetwork/blocks/id/uncles/uncleid	**id** is the number of a Ethereum Block and **uncleid** is the index of Uncle Block	This route returns an RDF containing the properties of an Ethereum Uncle Block
GET /ethNetwork/transactions/id	**id** is the Hash of a Ethereum Transaction	This route returns an RDF containing the properties of an Ethereum Transaction
GET /ethNetwork/receipts/transactionHash	**transactionHash** is the Hash of a Ethereum Transaction	This route returns an RDF containing the properties of receipt of a Ethereum Transaction

Table 8.3: Routes

code is found in [174]. A version to proof of concept using Infura [40] to communicate with Ethereum Networks was deployed in the Heroku platform in address [175]. The Figure 8.3 resumes the componets interactions of we middleware.

The Middleware connects to Ethereum and extracts its data through Web3.py. The extracted data is modeled in RDF graphs by RDFLib and using REST API in Flask to create endpoints SPARQL in EXtensible Markup Language (XML), JavaScript Object Notation for Linking Data (JSON-LD), N3, and Turtle formats.

The REST routes available in Middleware are in Table 8.3. The route returns an RDF capable of being used for SPARQL queries linked to other available datasets.

Using for example URI `https://ethon.herokuapp.com/mainnet/blocks/1?format=xml` and submitting the SPARQL query 8.1 , as a result have the the miner of the Ethreum block number one URI a Ethreum account represented by URI `https://ethon.herokuapp.com/mainnet/accounts/05a56E2D52c817161883f50c441c3228CFe54d9f`.

Listing 8.1: genesis.json

```
PREFIX ethextras: <https://ethon.herokuapp.com/ethextras.owl#>
PREFIX ethon: <https://ethon.consensys.net/>

SELECT ?miner
WHERE {
    ?block ethextras:miner ?miner .
}
```

8.4 Blockchain and WebSemantic Scenaries

The most common scenarios for using Blockchain with Web Semantic are cryptocurrencies and Decentralized Finance (DeFi) due to the origin of Blockchain and its most famous project, Bitcoin. Blockchain data queries, links, and external sources that generate currency market quotes exemplify this scenario.

DeFi are DApp frequently deployed in Ethereum, based on smart contracts and

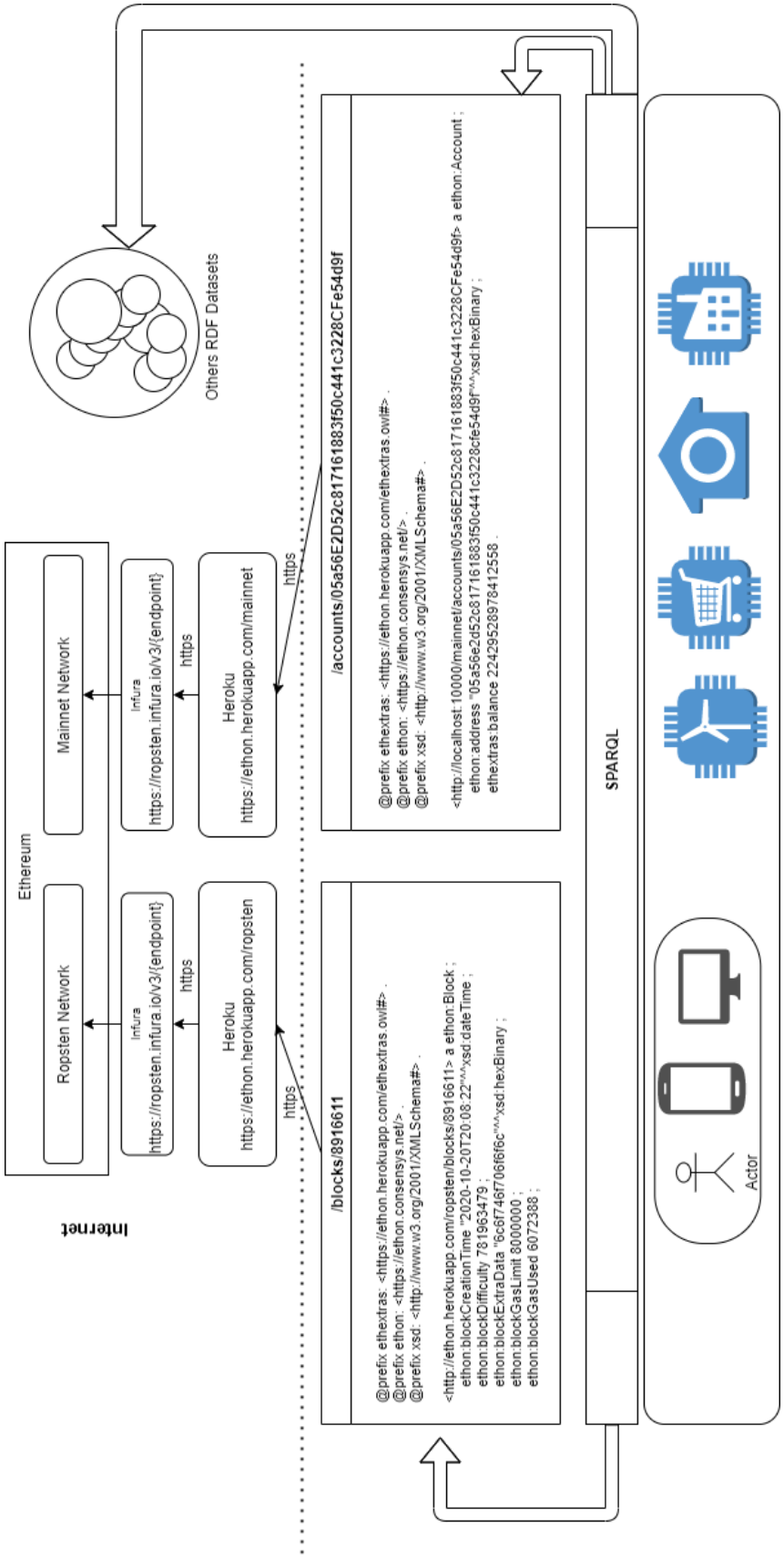


Figure 8.3: Diagram of middleware interactions

providing services and products associated with the traditional financial system, such as mutual funds, loans, exchanges, insurance, and securities trading. Links based on Semantic Web between Ethereum data and traditional web services would enable rich data for DeFi ecosystem applications that need to do consolidated searches in these databases, returning, for example, intelligent contracts and the DApp that own them.

In the domain of problems and new Blockchain applications, we have NFT. NFT are digital stamps on the Blockchain that prove ownership and reference tangible or digital assets. Real assets have a location, size, weight, and physical characteristics, and digital assets have a location in digital object storage. Some applications can use Semantic Web tools like our middleware to consume information from Blockchain, point to accounts, Smart Contracts, transaction history, and link with other information such as the asset's physical location and characteristics. This dependence on information will not always be consolidated, and reference them using links between datasets modeled in RDF graphs, has a potential use.

Still evaluating the Blockchain ecosystem, we have services called Oracles. They are potential beneficiaries of a Semantic Web-based dataset integration. Their function is to provide guaranteed and secure Smart Contracts that communicate with the outside world, querying, verifying, and authenticating external data sources. These services' external data could be temperature, product prices, credit information, payment verification, energy consumption, and traffic. In other words, it serves as a bridge between the Blockchain world and the External world, as Smart Contracts cannot access data outside the network (off-chain), and this access in some routines and necessary agreements in a DApp is highly relevant.

Ethereum Name Service (ENS) is a distributed name service coordinated by smart contracts, created opposite Domain Name System (DNS) and Interplanetary File System (IPFS), a distributed storage object proposition. Both make a DApp independent of centralized services and domains of private institutions and governments. Using Semantic Web techniques to generate datasets that organize, point out, and related data from these databases can bring unprecedented information and insights linked to information from the centralized and decentralized world advocated by the Ethereum community.

Scenarios of Blockchain applications different from the Fintech domain has recent spotlights. Blockchain's security features and the power and simplicity of data binding make the duo called Web3, Blockchain, and Semantic Web a potential solution. Due to the current expected demand for IoT, we can place the management of the expected high volume of data coming from these devices in Smart Cities (SC) and Industry 4.0 (I4.0) Apps as possible challenges in the coming years.

8.4.1 What can the new IoT Apps benefit from Semantic Web ?

The emergence of millions of devices that will produce data and interact with urban spaces in an increasingly intelligent and independent way is expected in the coming years. However, integrating these new devices in a standardized way and can benefit different scenarios simultaneously requires challenges regarding what we call the Internet of Things (IoT). Despite the various challenges and standardization problems, the investment in solutions that depend on IoT devices is constantly increasing every day due to its rapid adaptability, cost, and development [176]. A recent study suggests that, by 2025, global investments in industrial IoT should reach around trillions [177].

IoT devices that need to be used in hard-to-reach places or assets with cycles and long life expectancy, such as cars, need an infrastructure and solutions that support this feature while avoiding exposing them to vulnerabilities. New applications that use these devices to exchange goods and services and deal with financial resources or critical routines also need these devices to be reliable and reduce the risk, making several possible cases of use. It is impossible to guarantee the veracity of the data produced because the edge device is unknown. It is essential to address the possibility of using Blockchain, as it has the resources that allow transactions without necessarily knowing or trusting the originator. Taking this concept to IoT scenarios, Blockchain becomes a potential tool to protect communication from the devices. By allowing decentralized and reliable Peer-to-Peer (P2P) networks, communication between network nodes no longer need a reliable intermediary in exchanging messages or even the presence of a central authority [178].

All network nodes need a reliable central intermediate node to exchange messages. Some works using classic network security measures, firewalls, encryption techniques, and intrusion detection systems have already been proposed for IoT scenarios [179]. Such classic measures depend on trust in the central manager as they are based on a network and centralized infrastructure, such as cloud computing. In this context, we approach Blockchain in this work, proposing scalable and decentralized solutions for scenarios that need this approach and can benefit from it, like SC.

DApp make the duo of Blockchain and Smart Contract solutions stand out in developing applications not only in financial sectors, such as Fintechs. We already see in other sectors; Blockchain is being used as a data repository background to automate and allow the exchange of service messages between machines. Nevertheless, the current demand expected in the coming years for the use of IoT devices may make the management of their communication a challenge in the coming years, in

addition to the high volume of data expected for these devices in SC [180]. According to [181], potential sectors for IoT such as industrial automation already consider the volume and complex data management and already have big data problems and challenges. All industry assets, robots, Cyber-Physical Systems (CPS), sensors, actuators, industrial computers, and industrial networks are the relevant data sources for data analysis in Industry 4.0.

The intensive use of CPS and IoT devices leads to new frontiers and challenges in industrial sectors. Industry 4.0 mainly comprises recent technologies such as Artificial Intelligence and Smart Systems. These Smart Technologies tend to generate a flood of data, which we can no longer manipulate using classic tools and algorithms. The world data volume in 2011 was 1.8 ZB, and there are predictions that in the next few years, it should double every two years [182]. It is imminent that part of this data will be produced and stored in Blockchain by Smart Contract and distributed storages such as IPFS. In this challenging context, the motivation to propose models that extract data from Blockchain networks such as the Semantic Web gains relevance.

Using some Semantic Web concepts can solve issues related to the standardization of access to IoT devices and data produced by them that we do not have today due to the heterogeneity of interfaces and API. This data is consumed, identified, recorded, filtered, and discovered by a standard, and this makes IoT information integrate more collaboratively with other datasets, creating a Web of Things (WoT)[26]. WoT allows standardized services and solutions elaborated and combined with other RDF datasets as the endpoints RDF of this work, the Ethereum Web Semantic middleware, including filters and data aggregators.

A data model in RDF using Semantic Sensor Network (SSN) ontology [43] produced from IoT devices can be linked to endpoints that provide data modeled in EthOn and EthExtras, enabling SPARQL queries to generate new soft-realtime insights. SSN ontology describes actuators and sensors, covering their observations, procedures, characteristics, and observed properties. Some propositions of data extract and model in SSN are found in the literature. One is OpenIoT using XGSN [44], which implements a virtual sensor layer to give data visibility using SSN. The virtual sensor annotates the data and its storage in a graph database, the Open Virtuoso, allowing SPARQL queries.

8.4.2 Blockchain and Semantic Web in an Smart City IoT App

The recent implementation of 5G networks will provoke revolutions in our daily routines; one of the scenarios that should be significantly impacted is SC. The de-

mand for IoT devices on the market is already notable, with traffic, temperature, and climate sensors observed in urban environments in various traffic management, maintenance, and disaster prevention applications. These current network applications in SC have centralized systems with security flaws, mainly due to the difficulty in standardizing and trusting edge IoT devices. Urban environments should receive solutions based on IoT devices on a massive and large scale. In this approach, it is necessary to use tools that support transparency, scalability, and robust security, implicit requirements of Blockchain networks that support Smart Contracts.

Blockchain is a distributed, widely used ledger for managing tokens and NFT; applying it in SC Apps can bring positive inventory, reducing and eliminating the need for citizens to have certificates and physical documents. This procedure enables transparency, the agility of processes, and verification of authenticity in the use of public resources, making it possible to mitigate fraud and tampering with data and devices used by the services [88].

Security, especially reliability and immutability, are essential requirements for SC applications. A reliable base of historical and immutable records inhibits corruption, fraud, misuse, and waste practices. There are requirements to ensure public transparency and enable auditing of use and collection fees for urban services, usually controlled by IoT devices. Accurate data coming from reliable devices can prevent defects and accidents caused by negligence. These data provide analytical tools for analyzing asset obsolescence, scale adjustment, and adequacy to the volume of demand requested by the neighborhood. It provides insights into an adjustment of value by a socio-economic profile of users to adequation of the values to be charged by public services.

When implementing services based on the Smart Contract and Blockchain, it is possible to imagine hypothetical scenarios in which the public service user uses city tokens to access urban services. These users can use DApp to acquire and control the number of tokens that give access to services such as transportation, hospitals, entertainment spots, tourist spots, and events.

These services have mobility features, and IoT devices are the natural, technological choice. The Semantic Web is a potential solution to interact with the other databases and standardize the access and extraction of these data. The data in this pattern can be linked by Oracles and DApp in order to make decisions that involve predictive insights from various datasets to make decisions, such as natural disasters and virus pandemics.

When dealing with SC problems, a critical area is sustainable computing, a potential area to apply the Blockchain, DApp, IoT approaches mentioned in this article. These applications would provide users with data to raise awareness of natural resources, giving managers more accurate data and controls. Such reliable

management of the device base would allow better management of critical resources such as water and energy in urban areas. This information can help the aggregate city, state, or country sustainability data. Moreover, it exemplifies how reliable data management from IoT devices causes amplified social and ecological impacts.

Water distribution problems in Brazil exemplify a real scenario of a potential problem to be addressed with sustainable computing. Despite its abundant water resources, the population of Brazil is concentrated in areas with little water availability [183]. Recent climatic impacts have intensified the country's water crisis, and water managers are already seeking alternative solutions. An example is water extraction by desalination, still received in Brazil. Its expansion and effectiveness are dependent on technical, economic, and political issues still in progress.

An IoT hydrometer can be used using Blockchain and Smart Contract for scenarios where water management and intended use are needed. The decentralized basis of the Blockchain, its immutability combined with the usage policy built into Smart Contacts, would allow the analysis and even control of the daily consumption of water in residence. AI can handle excessive water usage patterns and identify leaks in the house. Valve opening and closing control triggers can be implemented by Smart Contracts and sent as commands to the IoT hydrometer avoiding waste.

This intelligent management of IoT Hydrometers in an SC can generate relevant savings for the final consumer in the water supply companies' bills and generate transparency and information on their consumption practice providing consumption awareness. In the long term, using a Blockchain solution expects minor environmental impacts by avoiding water crises, adding analytical intelligence with data from decentralized, immutable, traceable, auditable systems, and still being transparent and objective with the solution participants.

Therefore, the Blockchain Smart Contract approach presented is an instrument that can assist in governance processes as managers gain information on user behavior and the use of public resources, reducing administrative costs. Blockchain infrastructure setup uses P2P in cloud computing infrastructures, so some applications require low response latency, [184–186], requiring approaches that seek to reduce the high latency of the network by implementing Blockchain directly on IoT devices boarded.

8.5 Conclusion

This research contributes to some problems still open in the literature, mainly when considering the few works combining Blockchain and Web semantics. We propose the EthExtras ontology extension, complete the EthOn and create a model that can be used as a base to consume data of Ethereum simply. It is motivated to propose

integrating data from a Blockchain like Ethereum with external world applications in a standardized way. Our middleware using this combination allows Ethereum data in soft-realtime. This middleware and its endpoints RDF representing the objects of Ethereum can be used to use and link with external datasets, giving power and flexibility to developers to create o queries in an option of traditional API as Etherscan [187]. In future works, we are adding the NFT, Tokens, ENS, IPFS and some famous Oracle Services to EthExtras and implement this in the middleware as proof of concept.

Part IV

Final considerations

Chapter 9

Evolutions of our work

This chapter approaches some evolutions of our work, showing the results that reference us and the use of our contribution.

9.1 Low Power Smart Cities IoT network

Our conference paper [29] "Low-Energy Smart Cities Network with LoRa and Bluetooth" discussed in Chapter 5, is used as a reference in Low Power, Smart Cities (SC), and Internet of Things (IoT) networking works.

The work [188] proposes a flexible Fog Computing architecture that allows selecting between two different communication links (WiFi and Long Range (LoRa)) in real-time. According to the authors, the proposed Fog Computing architecture is formed by sensor nodes and an IoT gateway with LoRaWAN services, avoiding sending data workloads to the cloud by processing them in the perimeters of the Fog network. LoRa communications are used when the distance between the gateway and the IoT nodes is kilometers. This approach seeks to offer a solution with a low energy consumption profile. The work also applies a methodology that measures the energy consumption of the sensor node to compare and choose between the two different technology links (LoRa and WiFi), considering duty cycle, Payload size, and Scatter Factor (SF). In addition, the text cites our contribution as generic architectures that integrate Fog Computing in IoT-based applications, addressing the fact that we present results in a testbed and simulation to evaluate the feasibility of the proposal.

The paper [189] presents a network architecture that combines Long Range Wide Area Network (LoRaWAN) and Narrow Band Internet of Things (NB-IoT) for communication between sensor nodes, multi-protocol gateways, and cloud computing instances. Sensor nodes can be either LoRaWAN or NB-IoT and communicate with multi-protocol gateways that receive LoRaWAN packets and upload them to the cloud using Message Queuing Telemetry Transport (MQTT) over NB-IoT. This ap-

proach contributes to developing research on flexible infrastructures for complex IoT networks. The work references us because of Bluetooth Low Energy (BLE), which was used to configure local data transmission to the nodes that were integrated with LoRa clusters, with BLE connectivity being seen as an extension of the local area of a LoRa network.

The research [190] explores through a literature review and online consultation the feasibility of using Low Power Wide Area Network (LPWAN) in transport systems. This work cites our work because we approach LPWAN. The work carried out experiments with LoRa in a university campus area and a rural area. It concludes that the device mounting heights, the distance between the gateway and the sensor nodes, and the device brands affect the performance of an LPWAN infrastructure. As a result, it was possible to see that the application LPWAN in the USA is still at an early stage. Many agencies were not familiar with LPWAN technology due to the lack of ready-to-use LPWAN products available for transportation systems.

The chapter [191] cites our research effort to obtain low consumption and energy efficiency proposals for SC. In his work, he addresses intelligent methodologies that minimize transmission overhead by dynamically selecting optimal nodes for data transmission in Smart Parks. Smart Park in a SC is an integral part. These spaces allow people of different age groups, sedentary or active, to walk and moderate jogging/running. These activities are often monitored through individual smart devices connected to a smart health network. This data generated by multiple individuals is essential and can be further processed for additional clinical insights.

Works that discuss LPWAN in mesh topology as [192] to demonstrate mobility and the dynamics in the implementation of this nature of the network. They use our work as a reference for our work due to our discussion around using these networks in SC proposals and our experiments and simulation using LoRa networks.

9.2 Decentralized Applications

The book chapter "Blockchain for Machine to Machine Interaction in Industry 4.0" [34] e deeply discussed in Chapter 6, served as the basis for our first investigations in the field of decentralized applications. We discussed some possible and concrete possibilities of applications and projects applying Smart Contract and Blockchain in SC and Industry 4.0 (I4.0).

The systematic review [193] explores the relationship between Artificial Intelligence (AI) and the workplace, referring to research on human-machine interaction and I4.0.

The survey [194] discusses the potential of Blockchain in I4.0, listing drivers, facilitators, and resources associated with technology for insights, as well as relevant

applications for the topic. The survey emphasizes the importance of understanding the new opportunities created by using Blockchain and its ability to contribute to the processes inherent to I4.0. This work cites our research as a reference in the literature on data search by sensors.

The article [155] discusses consumer perceptions of cryptocurrencies. and cites our work for our discussion of disrupting traditional ways with Machine-to-Machine (M2M) payments, the IoT, and sharing economy, and [195] reference us as I4.0 and Blockchain research.

The research [196] on Internet of Healthcare Things (IoHT), or Internet of Things (IoT) lists IoT research in the field of healthcare that seeks to increase the quality of life with work that creates intelligent environments and increases the efficiency and intelligence of the services provided. The article addresses open challenges in IoHT device authentication mechanisms and Blockchain-based techniques. This article cites our chapter about detailing Ethereum Virtual Machine (EVM) Ethereum, which is used to run Smart Contracts.

9.3 IoT authenticating and authorization

The discussion around secure proposals for IoT and SC Apps have been motivated mainly by security weaknesses that can cause confidential data leakage and operational downtime in these applications. In these scenarios, any data falsification that IoT devices may present is critical and sensitive and can cause immediate discredit.

Our article "IoT Registration and Authentication in Smart City Applications with Blockchain" [32] discussed in Chapter 7, proposes to use API gateways that, in the background, validate messages from these devices by Blockchain, and we use Smart Contracts to identify and validate IoT devices on Ethereum, the most frequent platform for deploying this infrastructure as it mitigates message forgery. Some research already cites our approaches and scenarios proposed in this article and shows that our study contributes to works that present new authentication techniques and authorization of messages.

The article [197] presents a model of remote computing, where three layers of computing nodes are implemented to optimize computing tasks and traffic routing. This work references our approach in massively IoT deployments as the SC scenarios awaited the introduction of 5G cellular communications and new opportunities for IoT Apps.

In [198], Blockchain is used to register nodes of an Internet of Sensor Things (IoST) network and store data packet transactions in a secure routing model using Blockchain in the Proof of Authority (PoA) consensus model. During packet routing, Genetic Algorithm-based Support Vector Machine (GA-SVM) and Genetic

Algorithm-based Decision Tree (GA-DT) models are used in detecting malicious nodes, with the Dijkstra algorithm being used to find the ideal route. The option to use PoA in a change of the traditional Proof of Work (PoW) comes from the need for mechanisms with less time impact and processing overhead. The article references us for proposing a decentralized authentication model based on Blockchain, which provides the best fault tolerance for the network, emphasizing our ability in SC to extract data from outdated devices in unattended environments.

The research [199] is proposing a new model for managing IoT communication logs using Blockchain storage to ensure data privacy and research efficiency. The work designs a secure research scheme on Blockchain, using Asymmetric Scalar Product Preservation (ASPE) cryptography approach. This work cites our adoption of blockchain authentication to improve IoT security in SC Apps.

Networks of IoT devices, such as power grids or water supply systems, have emerged with priority, and several studies address IoT security. This. Existing authorization and authentication schemes are insufficient to deal with security due to the anticipated scale of IoT networks and the limited resources nature of devices. In this context, the article [200] addresses the interest in using machine learning techniques to assist in the authentication and authorization process of IoT devices. In this work, new advances and proposals for authentication and authorization for IoT networks are reviewed, including our work referenced by our research on IoT node registration.

Cyber-Physical Systems (CPS) combine physical objects with computing resources and storage during a data exchange over a network. Blockchain is a promising solution for CPS Apps such as Industrial IoT (IIoT) as it is fault-tolerant, reliable, and secure. Blockchain in CPS/IoT ensure secure message exchange in industrial applications. The [201] work references our work and presents applications, opportunities, and challenges combining CPS, IoT, and Blockchain.

Although IoT is one of the superior technologies of I4.0, data storage, computing, and communication-based on centralized clouds have several gaps, such as transmission delay, Single Point of Failure (SPOF), and privacy. Centralized access control in IoT Apps also restricts their availability and scalability. In this decentralized, tamper-proof, reliable, transparent, and immutable nature Blockchain scenario brings opportunities for new robust distributed and decentralized applications, Smart Health, Smart Finance, Smart Supply Chain, Smart Cities, Smart Manufacturing, Smart Government, Smart Agriculture, Smart Transportation, Smart Education, Smart e-commerce, and the Smart Grid. The article [202] references us and addresses all these opportunities consolidated with the advent of 5G and the popularization of the use of Artificial Intelligence.

The chapter [203] covers the medical resource allocation in SC, using big data

for projections and management decisions. These systems use IoT to collect massive data from medical resources and generate insights. The quality of the data collected allows for the optimal classification and allocation of medical resources in cities. Our article for addressing the topic of SC IoT Apps is referenced in this work.

In [204], the security management of devices in the IoT network, their maintenance, and accessibility are addressed. The article cites our work on IoT security in SC using Blockchain. It addresses issues such as leakage, data alteration/modification, and loss of privacy. This work reviews the various ways to improve the IoT system, such as algorithms and consensus techniques, covering the security and confidentiality of data in Smart Homes and Smart Cities using Blockchain.

The article [205] covers the main challenges and possible security issues in IoT Apps. It points out that IoT is already widely used in domestic, healthcare, telecommunications, environment, industry, construction, water, and energy management applications. Unlike computers, laptops, and mobile devices, personal data is generated by sensors, making it possible to combine real and virtual worlds. In this scenario, the article shows the crucial importance of investigating new security techniques, such as the need for light encryption, due to the limitation of computing resources of the devices. The work refers to us as IoT and Security research, whose objective is to provide a path in search of a secure IoT service.

The article [206] reviews studies that address Blockchain Apps and Smart Contracts related to chemical industries. The work highlights us for developing an API Gateway for IoT devices to sign, identify and authorize messages, using keys and essential characteristics of devices registered in Blockchain. Literature is classified, and our work is included in SC.

9.4 Conclusion

In this chapter, we address some works that cite our work and how they approach our results in the field of Fog Computing paradigm, SC, IoT, and Blockchain.

Chapter 10

Putting the case studies together

In this chapter, we will address scenarios that unite the propositions presented in our use cases; the idea is to give a general idea of bringing together the main technologies and contributions in an Smart Cities (SC) Internet of Things (IoT) scenario.

Our research addresses some possible scenarios to be used considering the challenge of proposing viable solutions to be implemented in SC applications with an IoT profile. The use cases covered during the chapters navigate to solutions that meet the need for infrastructure, availability, performance, security, and standardization attributes. Using the union of IoT technology attributes in a hypothetic SC scenario, we can illustrate all future potential of this meeting of technologies and propositions. We search for affordable solutions and a simplified setup with a security background and data management consistent with the requirements of such an extensive scenario of options and challenges in terms of volume of transactions and accesses.

The object of discussion in this chapter is the idea of seeking a single scenario that unites the technologies and architectures proposed by our case studies. Figure 10.1 and 10.2 represent these hypothetical scenarios considering Long Range (LoRa) as Edge communications, these architectures have not been tested, and it is just a provocation that presents the set of use of our propositions in this work.

When we consider IoT applications that need Long Range communication, we show the advantages of architecture such as Fog Computing as a point of intervention between these devices and their management infrastructure.

Architectures already established as an Low Power Wide Area Network (LPWAN) solutions, such as Long Range Wide Area Network (LoRaWAN) networks, already have the tools to receive the payload from distant devices with acceptable security standards discussed by the community. Data coming from devices can come from unknown or even fake hardware. A common point with Blockchain can therefore be found.

10.1 LoRaWan

The default security Activation by Personalization (ABP) and Over The Air Activation (OTAA) provides LoRaWAN security provides endpoint activations. ABP's setup uses fixed-end devices address, security sessions, and network parameters. The device address and the session keys change when a new session is established. The OTAA, the endpoint joined to the network using a dynamic device address, is the root key for deriving the session keys. However, these endpoint security methods do not guarantee the authenticity of the endpoint firmware or the message generated on the device. Therefore, a scenario using API gateways that authenticate and authorize the devices discussed in Chapter 7 could be complementary and bring additional security to the LoRaWAN activation protocol.

Bluetooth Low Energy (BLE) devices provide data on their characteristics extracted by LoRaEdge, which runs the IoT Edge API Gateway. After extracting the BLE feature, the data is sent to the IoT Edge API Gateway that signs the message, extracts the firmware from the LoRaEdge Device, and adds its metadata creating the payload to be sent. The LoraEdge payload is sent by LoRaWAN Servers using LoRaWAN Gateway. LoRaWAN Application Server sends data to the Blockchain API Gateway using Webhooks that validate the LoRaEdge device and its firmware and validate the message signature and the metadata containing the SC Application endpoint API address. For applications that want to interact with Smart Contracts on Ethereum, as discussed in Section 6.4.2, one way can be to use the pub/sub integration of the LoRaWAN Application Server. Ethereum data can be queried using Semantic Web tools. Our Middleware discussed in Chapter 8 exposes an Resource Description Framework (RDF) endpoint of Ethereum entity objects using EthOn and EthExtras vocabulary. SPARQL is used to generate queries to these endpoints and even link them to external databases.

10.1.1 Limitations

Some limits of this architecture can be observed. BLE devices are not authorized and authenticated; only the device that extracts the data, LoRaEdge, has its integrity checked by the security routine. The LoRaWAN network has data reception windows limited to a short space after receiving the data, so actuation routines cannot be intense in sending commands to the device. The integration with external API and MQTT is done by LoRaWAN servers, responsible for the security cycle of the network layers and application integration of data of the devices. It is possible to act with more freedom, tools, and Middleware before capturing the data by the device and after its integration into the LoRaWAN Application Server.

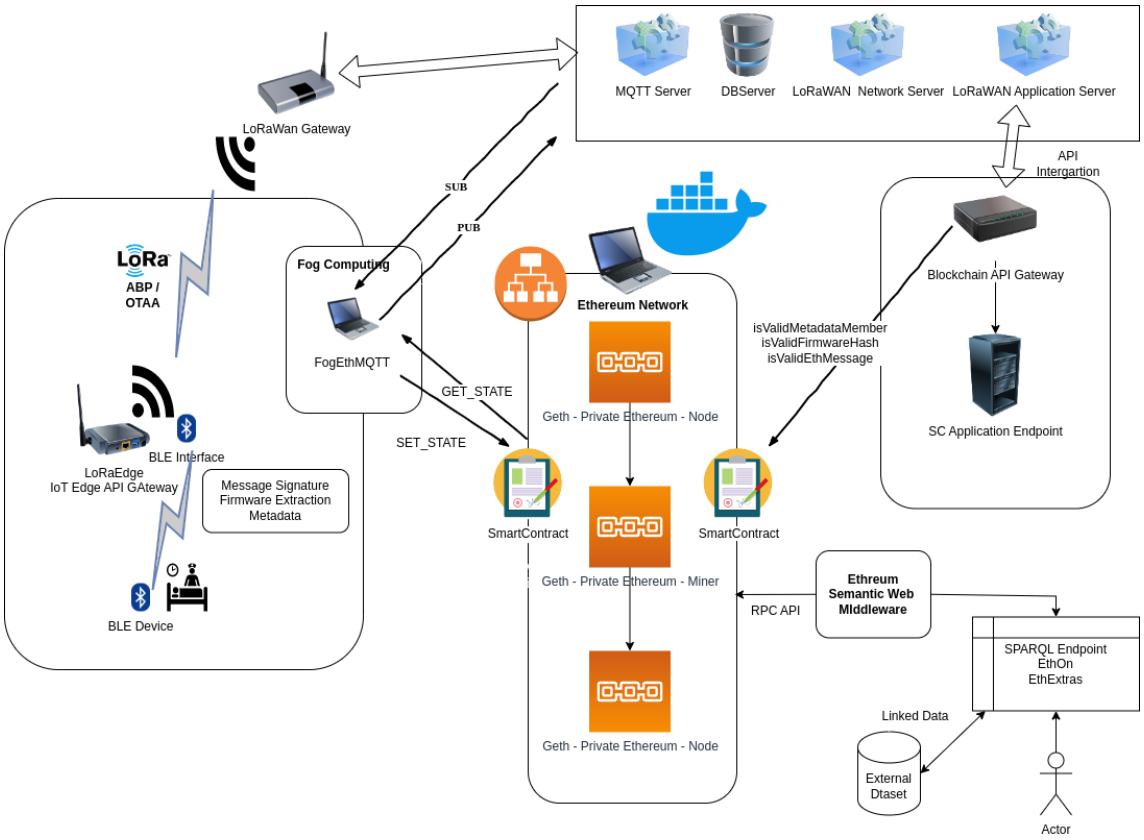


Figure 10.1: SC IoT with LoRaWan

10.2 Raw LoRa

Therefore, a scenario using Raw LoRa and API gateways that authenticate and authorize the devices discussed in Chapter 7 can also be complementary and bring added security. In a Raw LoRa link, there is no security and network control, and for these features to be used, they must be implemented from scratch, as discussed in Chapter 5.

Architectures using API gateways that authenticate and authorize the devices discussed in Chapter 7 can bring some security of origin and authenticity to the messages produced by the LoRaEdge device.

BLE devices provide data on their characteristics extracted by LoRaEdge, which runs the IoT Edge API Gateway. After extracting the BLE feature, the data is sent to the IoT Edge API Gateway that signs the message, extracts the firmware from the LoRaEdge Device, and adds its metadata creating the payload to be sent.

The LoraEdge payload is sent to LoRaFog. LoRaFog using Blockchain API Gateway validates the LoRaEdge device and its firmware and message signature and metadata containing the API address of the SC application endpoint that receive the payload if validated. For applications that want to interact with Smart Contracts on Ethereum, as discussed in Section 6.4.2, FogEthMQTT can be integrated into

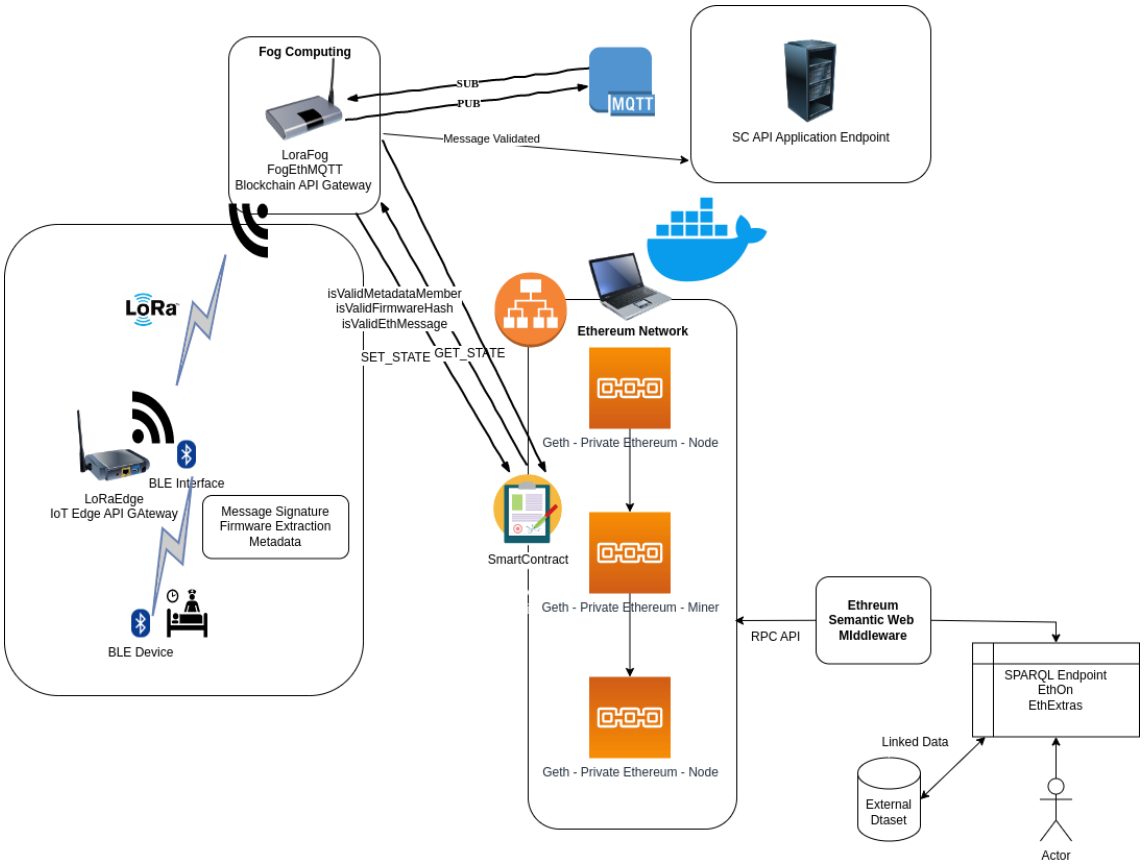


Figure 10.2: SC IoT with Raw LoRa

LoRAFog.

Ethereum data can be queried using Semantic Web tools. Our Middleware discussed in Chapter 8 exposes an RDF endpoint of Ethereum entity objects using EthOn and EthExtras vocabulary. SPARQL is used to generate queries to these endpoints and even link them to external databases.

10.2.1 Limitations

Some limits of this architecture can be observed. The downstream window in Raw LoRa and network and data security need to be implemented from scratch. As in the scenario with LoRaWAN, BLE devices are not authorized and authenticated; only the device that extracts the data, LoRaEdge, has its integrity checked by the security routine. LoRaFog is a centralized component and must have redundancy and load-balancing strategy.

10.3 General Aspects

The scenario discussed in 10.2 presents the Edge connection by LoRa, but it serves and can be thought of in use cases that need to use other wireless network technolo-

gies in Long Range, such as 5G and Wi-Fi, among others. The applications these architectures cover are presented in 5 the SC LPWAN applications. We highlight those with hard-reach locations where LoRa and its ability and resilience to communicate efficiently without sight and underground. However, suppose we remove the Long Range component from the requirements. In that case, it is possible to have TCP/IP links between the devices and Gateway Fog Computing, serving a vast mass of applications that need the means to validate their payloads and use Ethereum as a repository of historical logs. An IoTApp concept also introduced in 5, responsible for abstracting an application made to run on IoT devices in the style of the mobile App Store, can use validation and security routines as well as interaction with Smart Contract in a special way called IoTDApp (Internet of Things Decentralized Applications).

10.4 Conclusion

In this chapter, we present architectures that satisfy the central idea of each of our use cases. Although hypothetical, it illustrates some of the points to be work in future studies and the limits of our proposition.

Chapter 11

Discussion

This chapter discusses this research's significant advances and challenges, considering the Smart Cities (SC) scenarios. SC is ideal for new and disruptive Internet of Things (IoT) Apps, using urban space to propose solutions and benefits for a population.

11.1 Low Power SC Network Discussions

Environmental information, such as air quality, temperature, and precipitation, help managers to predict disasters or even inform residents or tourists about the conditions of the city's points [157]. The data extracted from an SC has a significant percentage of data characterized as sensitive, which makes security a prerequisite. These IoT devices are not standardized due to the substantial variability of available brands and the absence of a communication standard. An urban environment that is deployed is often unknown or is not sure of the suitability of the installer and the user and cannot be trusted. Its low cost of ownership and simplicity of configuration make managing scale a challenge.

Updating IoT devices installed in places of difficult physical access is a security challenge. Examples are underground pipes, power sources, sewage, dumps, constant low or high-temperature sites, and high mountains. In these cases, a simple battery change or firmware update leads to operational risk and is often economically unfeasible.

This scenario reflects on the security features provided by default in these IoT devices; New layers and security proposals are needed to exchange messages from IoT devices and applications that cannot rely on a proper firmware update cycle. A similar scenario would be that of IoT car devices, which have longer update cycles, as they depend on the dates and availability of their users to schedule revisions and recalls.

Problems of infrastructure available to SC IoT Apps are expected. The contributions of our research present scenarios of IoT Low Power networks using LPWAN technologies for communication in inhospitable places without, where frequent replacement of batteries is hard approached in Chapter 5. This proposition could minimize access to these locations and lessen the risk of having an out-of-date device sending data. The validation of a message sent by a legacy IoT device with outdated firmware has a proposed solution in Chapter 7.

11.1.1 IoT a SC solution

Extracting data from this IoT devices provides relevant insights such as disaster prevention, transportation management, and area occupancy. These data used in predictive models can even help public managers to make informed decisions about their actions [7].

Technology such as 5G can accelerate the popularization of high-throughput automobile IoT devices, allowing city traffic data to predict future traffic and congestion [156]. The SC is a candidate to be a pioneering use of mass connected 5G IoT, justifying our arguments of IoT mass adoption and improving the costs and risks of an insecure project. When extracting data from an urban area, the most common scenario is to use IoT devices to receive data from the most diverse sources and devices [6]. Therefore, some places have unique characteristics.

All these mass adoption issues of IoT devices justify our research, seeing the possibilities of new Blockchain-based applications deployed widely in an SC imminent. Countries like the United Arab Emirates, the USA, and the UK already use Blockchain in the public sector. Dubai plans that all public services be Blockchain-based by 2020 [149].

11.1.2 LowPower Network and Blockchain in a SC IOT Solution

Projects such as Helium use LoRaWAN services to send and receive LoRa packets from nodes, using low-cost, low-power wireless networks and blockchain. Its cryptocurrency HNT is used to incentivize the participants to create public gateways. Hotspots form miles of wireless network coverage with miner devices using innovative PoW or Proof of Capacity (PoC), and Low Power networks like LoRa [81].

An edge computing strategy that addresses cooperation and collaboration is addressed in [64], where an incentive-based mechanism is adopted to share resources and deliver services, offering a reward to Blockchain participants. An Artificial Intelligence PnP model in the Edge Computing paradigm in SC is approached in [66].

11.2 Security Discussions

Our investigation of propositions using Blockchain in SC has been highlighted when we list centralized and cloud models' limitations and risks characteristics. In a centralized network paradigm, auditing or controlling users' actions is based on the trust of the company or organization that manages the infrastructure.

The Centralized security systems, where we find user records, passwords, user access keys, and other artifacts, have a latent weakness. Even with auditing and governance rules, these centralized systems are not guaranteed to change data without the user's exclusive authorization.

Infrastructure centralized have scalability costly and restricted, and the risk of a single point of failure to be used in a cyberattack as Distributed Denial of Service (DDoS) increases [126].

One SC App has usual security centralized architecture, sometimes inefficient for unpredictable growth applications. In attacks concentrated on a weak point, low scalability and centralized network architecture could be ineffective in receiving an increasing number of simultaneous transactions.

These Blockchain characteristics allowed us to use it as a basis for our investigations and propositions. Still, as presented to us, the possibility of developing applications with Smart Contracts Chapter 4 allows us to use other potential scenarios for its use besides Fintechs.

11.2.1 Blockchain in focus

The justifications for using Blockchain in the latest technological innovations are already far beyond Hype for adoption. Have a wholly disruptive capacity due to being a platform known for strong data security, with the corruption of the stored and threaded next to the impossibility.

Blockchain is already advancing in journeys beyond cryptocurrency applications. Blockchain is widely used in new financial decentralized services called Decentralized Finance (DeFi) that uses cryptocurrencies as the primary tool, Figure 11.2. But the strong cryptography, scalability, and resilience of Blockchain potential and disruptive technology can be applied in various application domains and sectors of our society, covering almost all aspects of business, industry, finance, and governance. Surveys as [57] address Blockchain apps and their security aspects, and study [58] approach challenges and opportunities of using Blockchain as background for IoT Apps.

The most common use of Blockchain is at Fintechs in trading and managing digital assets, the Internet of Money (IoM) [166]. However, the decentralized and security characteristics of the Blockchain allow the development of proposals far from the financial applications domain, Figure 11.1.

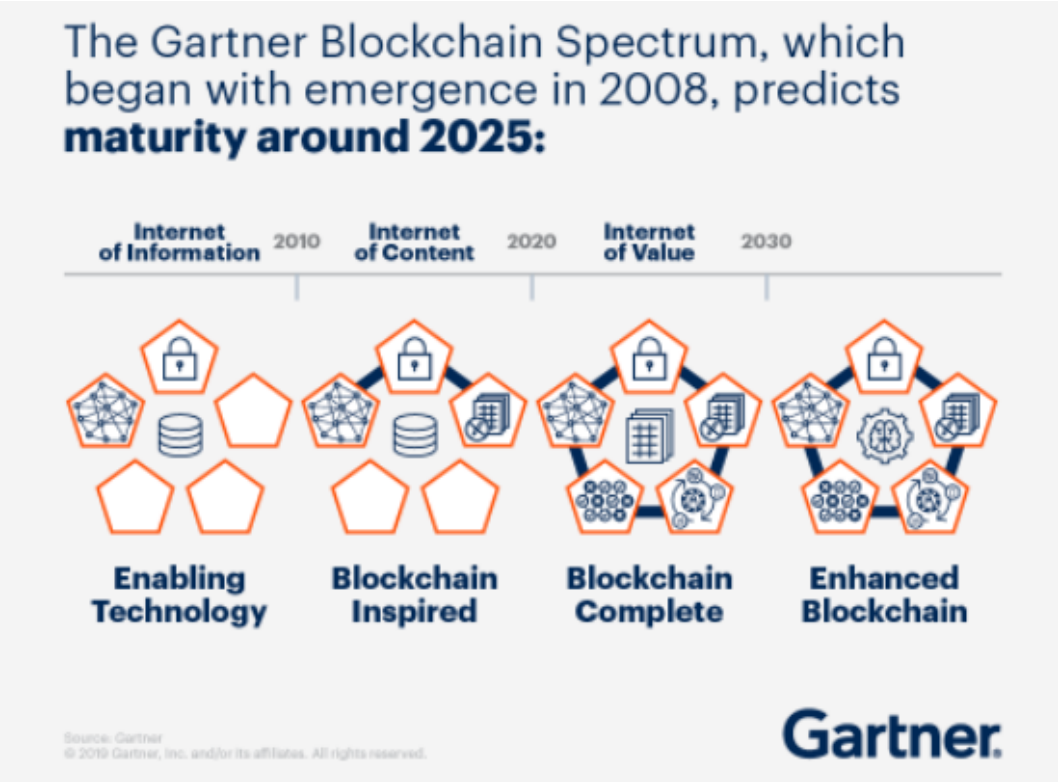
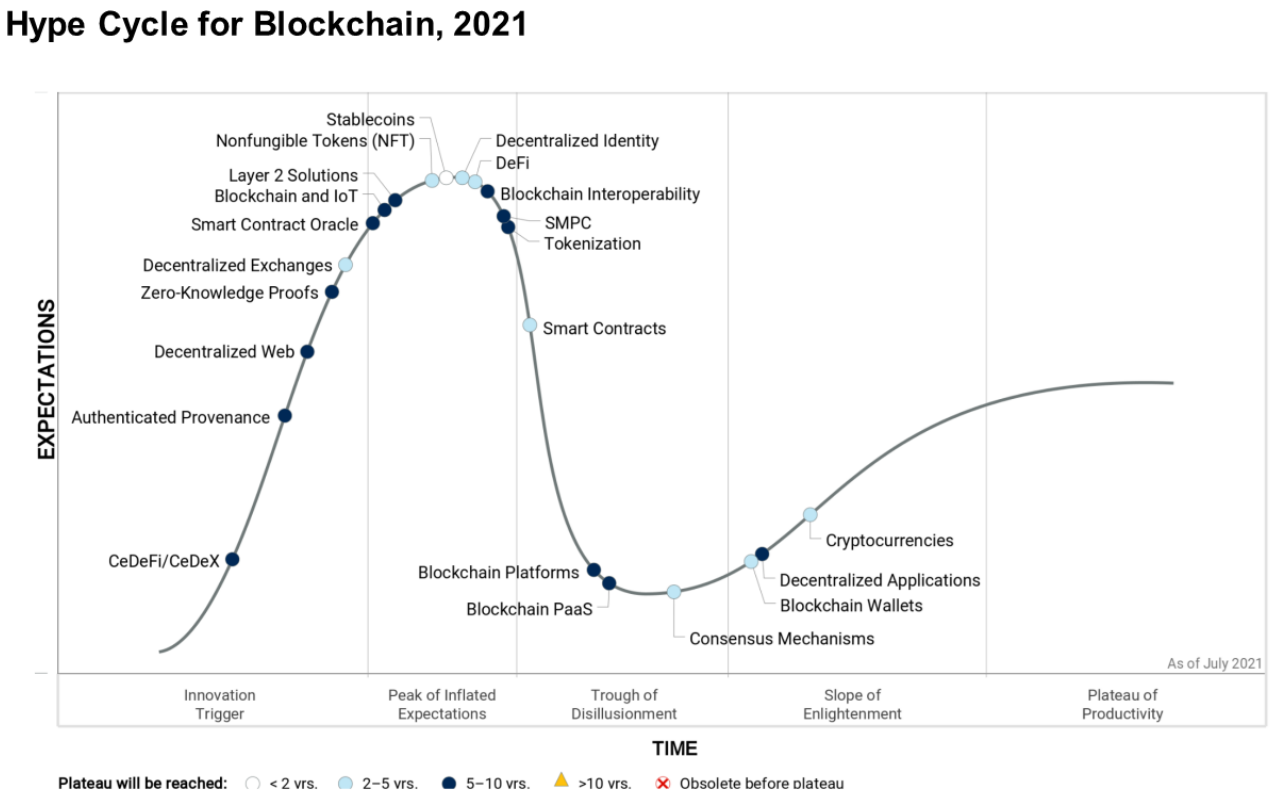


Figure 11.1: The Gartner Blockchain Spectrum [1]



Source: Gartner (July 2021)

747513

Figure 11.2: Hype Cycle for Blockchain 2021; More Action than Hype [2]

For example, The Blockchain network is scalability and resilient and is effective against Distributed Denial of Service (DDOS) attack [148] because of its decentralized architecture.

Blockchain is used in data security out of SC, the revocation in industrial environments is presented in [65]; this work focuses on data access and revocation in Smart Factories using IoT. The security management implemented blocks and revokes the access of malicious users responsible for identity authentication, public keys, registration of user attributes, and revocation lists. The work proposed an access control scheme and protocol using the characteristics of an intelligent factory.

An advanced feature of transaction automation in Blockchain is called Smart Contract, which allows developing the multi propose Decentralized Application (DApp) [36], using the robust security cryptography in the resilient P2P network.

It is already possible to see new economic models based on Blockchain to enable basic communication infrastructures, and cryptocurrency rewards for participants begin to be seen in several projects. There are scenarios where the leading computer network is expensive for users or controlled by central institutions, such as governments and private companies. One of the options is to form spontaneous communication networks that monetize the participants with cryptocurrencies, increasing the chance of encouraging area coverage.

Edge computing using cooperation and collaboration is proposed in [64] To share resources and deliver services. In this research, an incentive-based mechanism is adopted to offer a reward to the participant using Blockchain.

The DApp Paradigm

For a DApp to be completely decentralized is necessary a new paradigm of devel, which assumes that all its components are decentralized without dependencies on traditional internet standards or any centralized service. An example of this independence is Ethereum Name Service (ENS), which is responsible for a resolution name of a DApp using a Smart Contract without internet name resolution standards such as Domain Name System (DNS). The disruptive decentralized storage services, IPFS, show relevance in this scenario, are responsible for providing decentralized storage resources, and Whisper is used in a decentralized message system. This set of tools possible the DApp developer independence of cloud computing services or thirty part message Application Programming Interface (API) to deploy a secure, decentralized, and resilient application triggered by a Smart Contract and running in an Ethereum Virtual Machine (EVM).

These development tools for the decentralized paradigm are already widely available in code published by the community for Ethereum. Due to its disruptive characteristics, this software is constantly changing. Their effective integration with

existing Web projects such as Javascript has been widely observed due to the immense adhesion of the open-source developer communities to the project.

One notable tendency coming from a decentralized ecosystem as Ethereum DApp has high relevance to the big technology companies and its necessary great attention. To access real-world services and applications outside the decentralized Blockchain ecosystem, the DApp community proposes a new concept to access an external service by Smart Contracts called the Oracles. This concept and paradigm can be applied and integrated with the current Facebook development strategy, allowing the new Oracles, DApp, and DeFi to be integrated with current application services.

11.2.2 Why Ethereum

Mainly a Blockchain Smart Contract platform has obtained standout on stage, Ethereum. It is already also called World Computer, become in recent years the preferred platform for DApp. Ethereum is a preferred platform for managing new cryptocurrencies, Tokens, digital assets, or Non-Fungible Tokens (NFT). Much is speculated even on the future overvaluation of Ether (ETH), its base currency, against Bitcoin, due to Ethereum's ability to be the platform for DApp that use Smart Contracts to automate their transactions [37].

The open-source Ethereum Blockchain is growing in popularity as a cryptocurrency platform for developing smart contracts. It has most of the attributes needed to create DApp. Today's communities of developers of DApp already have an ecosystem of decentralized and usually open-source tools that allow them to expand features and resources not yet natively provided by Ethereum or other Blockchains support Smart Contract. One of these examples is Decentralized Storage. Projects with Swarm and Interplanetary File System (IPFS) fulfill this task, providing DApp developers with a decentralized platform to store their files with immutability availability.

These decentralized projects promise to change the next generation of applications by forming a new development paradigm for Web3.

Community-developed tools for Ethereum follow a continuous and accelerated pace of change, seeking to provide increasingly integrated and simplified means for integration and relying on spontaneous support from open-source developer communities.

Ethereum Networks

During our research, we use Ethereum development networks such as Ropsten, which enables the experience of using a public Blockchain network in our experiments in the same way found in existing applications that focus on cryptocurrencies and

digital asset trading NFTs.

A more practical way to access the API of public Ethereum networks, MainNet, its leading network, and test networks such as Ropsten can be accessed using projects such as Infura. It allows prototype routines on Ethereum, such as creating Smart Contracts and calling them without privately deploying the infrastructure of Ethereum nodes.

The Ropsten Blockchain has the same characteristics as the leading Ethereum network, making it possible to debug and test DApp and their Smart Contracts without MainNet's Ether consumption. There are several ways to get ETH on Ethereum test networks; one of the ways is the so-called Faucets, widely found on the internet.

During the development of our research, Ethereum became the second most traded currency and the leading platform for developing DApps. The long-awaited Ethereum 2.0 version, which should change transaction times and costs, should further accelerate the platform's adoption for other cases besides Fintechs, such as SC and I4.0, as presented in this work. We use Ethereum as a Blockchain base and present validation models for IoT devices 7 and a standardized consumption model that can be integrated with other bases in 8.

11.2.3 Identification and autorizarion IoT

We propose identifying, registering, and verifying an IoT device inserted in an SC App. One of the pieces of our architecture uses an API Gateway to verify identity and authenticate sign messages incoming of Edge IoT devices, using Ethereum Blockchain and Smart Contracts.

IoT devices installed in hard-to-reach areas typically have long-term use characteristics. Data from reliable IoT devices, such as our authorization and authentication proposal **Blockchain API Gateway**, can give transparency and certainty to SC Apps that provide visualizations and insights into urban activities.

A typical SC IoT App problem is registry devices to verify the source of sensible urban data. A reliable architecture to transmit and receive IoT data using Blockchain and Smart Contract for authentication and verification, even the obsolescence of firmware or device vulnerabilities are no longer obstacles.[128].

Validating and identifing a Device in a SC

IoT devices that have a long life and are subject to their firmware without updating and support make them vulnerable. Any falsification of data from these devices can cause discredit. Security weaknesses can cause confidential data leakage and operational downtime. The proposition of using API gateways and Blockchain to

validate messages from these devices becomes relevant as it would, for example, avoid message forgery. Scenarios like this are potential candidates for researching and proposing new techniques for authenticating and authorizing messages.

We propose Blockchain and Smart Contracts to identify and validate IoT devices in Ethereum, the most frequent platform for deploying this infrastructure. In later identification, the Merkle Tree data model is used to validate metadata identifying an IoT device. A Merkle root hash is generated based on this metadata and stored by Smart Contract on the Blockchain for future validations of the device [207].

By using the characteristics of IoT devices to validate the payloads of an IoT device, we achieve an additional level of confidence. This payload is accompanied by elements and attributes that identify the IoT device and verify if it matches the same device previously registered. These characteristics are, for example, the firmware hash and the root hash of a Merkle Tree originating from metadata that identifies, for example, the device's name, location, the owner, and others. These messages containing the payload and these attributes are signed at transmission to be later validated on Ethereum through a Smart Contract. We use an API gateway that bridges the gap between the IoT data management servers and the device world, making it possible for the message to be delivered to the server's zone only after effective validation and identification.

The previously registered characteristics are verified, regardless of the Device's default API and security features, and message forgery is made difficult because of an extra layer of validations. This routine, for example, allows that even if a device has outdated firmware. It is subject to exploiting security flaws; it can receive an extra layer of validation if its messages are delivered before effective delivery to IoT data management servers.

In our work, we compose an architecture for validation and authentication of messages coming from API devices in the Fog Computing paradigm, using SC Apps as a motivator. Despite having SC Apps as a motivator, we can also apply our work to the vast majority of IoT use cases in Industry 4.0 that require a data origin guarantee.

We develop daemons that act as API Gateway. We use the project IoT Device Management [30] as a basis; we use its Fron End to register a device and its metadata on the Blockchain, in addition to Smart Contracts that validate messages and the presence of metadata using the Merkle tree. Two Gateways are developed in the same paradigm, the Fog Computing **IoT Edge API Gateway** and **Blockchain API Gateway**.

The **IoT Edge API Gateway** is the daemon responsible for receiving the messages from the local sensors and preparing a payload containing the device identifier, the IoT message, and its signature, the address of the destination API, and the proof

of Merkle, used for further validation of this metadata in the Merkle tree.

The **Blockchain API Gateway** is the daemon located in the Fog network and has contact with the application network. The daemon receives the payloads from the IoT devices. It is responsible for authenticating the Device and its message by verifying the signature owner and its authenticity using Smart Contract calls to the Blockchain. The payload has metadata with the address of the API destination of the message; this metadata is verified by Merkle proof by Smart Contract in the Blockchain to be authorized for its sending.

When approaching the issue of authenticating an IoT Device and validating its transmitted data, we use the new decentralized paradigm of DApp to merge with the centralized world, the reality of most applications. In our prototype network architecture, we seek to isolate the AppNet application network from the IoT EdgeNet devices network. We use a container as the only point of contact between networks running we **Blockchain API Gateway** in a Fog Computing paradigm on FogNet. This container is responsible for validating and authenticating messages arriving from EdgeNet IoT devices using Blockchain and Smart Contract and sending the messages to some SC management application API on AppNet.

To test, we deployed containers with limited resources representing a specific IoT Device in the Edge Network. Edge Network participants represent IoT devices by signing and sending their payloads to **Blockchain API Gateway** on FogNet. The endpoint address is sent as metadata to know which endpoint delivers the payload message.

During the device registration on the Blockchain, the Firmware Hash and Merkle Root generated by its metadata are registered, one of these metadata being the endpoint API address of the message. For testing, we use a well-known SC API from the literature, the IoT-Framework Engine [160][31], and its web frontend, the IoT-Framework-Gui [161].

We chose this project to represent an SC API because of its architecture and components developed in Erlang, a platform that has shown promise in products that need to meet a large number of requirements and allow large scale.

The technologies and Web3 tools used in the API Gateway helps to understand and generate metrics for a possible SC IoT App. It is essential because most use cases of Blockchain and Smart Contract are used for NFT or Cryptocurrencies projects. Our experiments can show in detail the main features, strengths, and weaknesses of each Web3 technology and propose an IoT validation routine to identify the origin of messages sent from the devices.

11.2.4 Using Blockchain Ethereum as a tool

Our proofs of concept use widely available, free software tools. We use containers and Docker orchestration as a tool for testing at scale, commonly known as an application modernization strategy. Containers allow us to instantiate services in a standardized and simplified way. For testing a real IoT, we used Raspberry Pi and Pycom, also available and compatible with free operating systems and widely known languages such as Python.

To automate the deployment and orchestration of the network and container environment, we use Docker Compose. It was responsible for deploying the Fog, Edge, and SC App Servers subnets, enabling logical and network isolation.

For our tests, we deployed the Smart Contracts on Ropsten accessing via Infura. Smart Contracts called by the IoT device registration DApp and **Blockchain API Gateway** calls access Ropsten using the endpoints provided by the infura.io project.

Infura provides instant and scalable API for Ethereum networks. Smart Contracts of IoT Device Management was deployed on Ropsten using the Truffle tool using the API endpoints offered by Infura.

The IoT Device Management web interface and IoT device registry, a version of the DApp IoT Device Management web frontend developed in React, was deployed on a PaaS (Platform as a Service) cloud service Heroku.

11.3 Web Semantic Discussions

Our proofs of concept use widely available, free software tools. We use containers and Docker orchestration as a tool for testing at scale, commonly known as an application modernization strategy. Containers allow us to instantiate services in a standardized and simplified way. For testing a real IoT, we used Raspberry Pi and Pycom, also available and compatible with free operating systems and widely known languages such as Python.

To automate the deployment and orchestration of the network and container environment, we use Docker Compose. It was responsible for deploying the Fog, Edge, and SC App Servers subnets, enabling logical and network isolation.

For our tests, we deployed the Smart Contracts on Ropsten accessing via Infura. Smart Contracts called by the IoT device registration DApp and **Blockchain API Gateway** calls access Ropsten using the endpoints provided by the infura.io project. The current Web API, Mobile, and IoT Apps need to consume Ethereum data and receive the call of DApp. The Ethereum Blockchain has relevant information in entities, such as accounts, Blocks, transactions, receipts, contracts, Tokens, NFT.

Our research aims to address scenarios that can integrate and consume data from

this new ecosystem of DApp. Using data of the traditional datasets and applications of the current Internet, already foreseeing its use with the expressive increase of IoT devices expected with the popularization of 5G, using Semantic Web.

In literature, many propositions exist to standardize models and integrate data from different sources. However, one of these is the Semantic Web, which proposes linking Web distinct Resource Description Framework (RDF) dataset sources, allowing query data from the Web similar to a database, using as schema Ontology Web Language (OWL). Using Semantic Web technics enables standard access and integration, expanding the possibilities of obtaining insights by queries provided between different application domains.

Our research is the efforts to apply Semantic Web technics, extracting and linking data of an Ethereum network. It provides data access standardization to DApp, Oracles, IoT Devices, and new applications that use Ethereum and Smart Contracts as developments tool.

EthOn [168] is an ontology found in the literature to represent the main Ethereum entities using RDF and OWL. It describes the Ethereum objects as classes in ontology covering the major Blockchain and State Transition concepts as Blocks, Accounts, Transactions, Contract Messages, States, Nodes, Protocol variants, forkings, and network properties.

The EthExtras ontology adds components to EthOn, proposing some auxiliary classes that facilitate the understanding and relationship between Ethereum entities. A middleware web that uses EthOn and EthExtras ontologies extract data in soft real-time of Ethereum producing endpoints RDF, making Ethereum entities available as Universal Resource Identifier (URI) for visualization, link, and query with other datasets.

This research contributes to some problems still open in the literature, mainly considering the few works combining Blockchain and Semantic Web. Explore new App integrations with Ethereum in social platforms as Facebook would enable WhatsApp using DApp data or other Blockchain details as Token information. Instagram could monetize digital assets as a Photo using NFT registered in Ethereum and accessed by API.

We proposition the EthExtras ontology extension to improve the EthOn and create a simplified schema to consume the Ethereum Blockchain ecosystem data. We prove that the middleware Web allows the access of Ethereum data in soft-realtime, show, and a Semantic Web model of Ethereum possible to use by applications in a standardized way.

The middleware endpoints are presented in RDF and represent the objects of Ethereum to be used to link with external datasets, giving power and flexibility to developers to create queries in an option of existing Ethereum visualizations API as

Etherscan [187].

11.4 Case studies, limits and weights

11.4.1 Low Power SC IoT Network

LoRa and BLE networks have low transmission rates compared to 4G, 5G, and Wi-fi networks; these limits are improved using gateways in architectures such as Fog Computing, allowing processes or services to be managed closer to the network's edge. Fog computing saves up the IoT device's computing and bandwidth capacity. One example of a limit that restricts LoRa use is in streaming applications. In our case study, mixing Low Power Technologies such as LoRa and BLE, this low bandwidth capacity is supplied with Fog Computing, responsible for filtering and processing data before uploading to the cloud using an IoTApp.

LoRa has limited ranges promised in its 45 km specification, using rates between 0.3 and 50 kbps on unlicensed frequency. LPWAN LoRa, have a limit of energy consumption, and the idea is that the device wakes up the minimum necessary to send or receive data. Its type of network technology is unsuitable for synchronized mesh networks, be the ALOHA the architecture preferred to non-cellular star network., using the transmission medium only when needed to send a frame over the network.

In a network infrastructure, using Wi-Fi has high power consumption; some IoT scenarios are unfeasible; stay restricted to situations SC with wall power with gateways connected to a fiber optic backhaul or high-throughput wireless connections. However, when we do not have any available backhaul or the cost of infrastructure and energy is prohibitive, these problems and limitations do Low Power networks become a relevant option.

We don't have a detailed analysis of the testbed power consumption and the crshortlora network simulations of the 5 case study. An example of a phase that can be an energy bottleneck is the BLE device scan of the LoRaEdge Algorithm, responsible for extracting information from edge devices. In this case, a network with an excessive number of IoT devices BLE at the edge and an excessive frequency of extraction of characteristics using GATT can consume a lot of energy.

11.5 SC IoT Security

The centralized network security architecture discussed in depth in Chapter 7 has several security limitations, DDoS attacks being a typical example, where a concentration attack on centralized servers could result in failures. Privacy is another risk point to be considered in centralized servers that can have sensitive data from the

citizens of the cities, such as health information, purchase preferences, and behavior of visiting places and spaces. In this case, there is no guarantee of control over how the user's data is used in its management environment or even by whom, as data stored in a centralized infrastructure generally does not necessarily have an obligation to identify those responsible, audit access, and modify data.

However, when we seek a decentralized architecture using background tools such as Blockchain networks. We are faced with the need to initially consider the limitations of transaction times and cost and storage limits of the solution. Consensus algorithms such as Prove of Stake (PoS) are new proposals that aim to solve some of the transaction times and energy costs of mining a block. However, we have not yet tested them, being a potential scenario for future work. In a Blockchain network, we have the consensus algorithms as a limiter of the transaction times, as we analyzed in Section 6.4.1, where PoW did not manage to provide us with satisfactory times, but for a payment gateway or low-volume writing applications during the day, could these times be tolerated.

Our API Gateways proposition uses a hybrid model that merges the conventional infrastructure with the security background of Ethereum Blockchain, Web DApps, and Validation Smart Contract. The limits of these propositions are found in the technologies involved and centralized and decentralized architectures. The Blockchain API Gateway, being the central point of validation of the payloads, is exposed, for example, to DDoS as well as the API SC; this happens because it represents the hybrid contact part between Cloud Computing and Fog Computing. Conventional security rules and strategies such as firewalls, DMZ, and API call limits become necessary in a possible production deployment. We did not use HTTPS in any experiments, but this protocol is highly recommended in production environments exposed mainly to the internet. Load balancing and auto-scaling should be welcomed in the solution to increasing the resiliency of the Fog Computing Gateway.

The devices running the IoT Edge API Gateway have some limitations related to the accessible nature of these devices. To run the IoT Edge API Gateway daemon, this must have a Linux operating system and be able to run node.js and web3.js. For it to be effectively secure, it is necessary to mitigate the exposure of private keys necessary for the signature of messages; in this limitation, tamper-proofing devices, for example, could be used.

Our solution does not deal with problems and hardware arising from reading, for example, a sensor, an example of this error is a collection of a wrong temperature and recording on the network. The idea of device validation with Blockchain API Gateway is to prevent data from being generated by hardware with unknown or altered firmware; if this data is generated wrong by reading it, the data will be sent wrong. The idea of using Blockchain to aid in verifying IoT messages is to

have confidence that even if it is wrong, it is registered in its destination API after authentication and authorization. Suppose the data is written in a Smart Contract in future works. In that case, it still has the immutability of a decentralized base as a guarantee. Data written in Blockchain is immutable, auditable, and reliable, making its modification or fraud by managers or other stakeholders impossible.

11.6 Semantic Web Ethreum Middleware

Our Middleware consumes data using the EthOn ontologies, and our proposed EthExtras extension was built for the Ethereum Blockchain. Although the concepts and attributes involved are common to all Blockchains, it is specific.

Middleware was developed using the web3.py library and therefore limited to its resources. The idea of using SPARQL to generate queries to Ethereum and being able to link them to other databases is the main motivator of our work by presenting the case study in Chapter 8. But when using more elaborate queries that require calls to several URI endpoints of entities with a sequence, such as Ethereum blocks, in search of aggregating information, the response time of this query can be prohibitive.

The Middleware was developed as a web service and has centralized features, requiring network firewall protection, load balancing, and auto-scaling to have resilience and security.

11.7 Conclusion

This chapter discusses the main topics and results of our research, and we show some significant advances and challenges of this research in addition to delving into some relevant themes that we addressed during the presentation of the case studies. We tried to address some of the main questions we received while reviewing articles and presentations.

Chapter 12

Conclusion and Future Works

12.1 Conclusions

In this Chapter, we show the conclusion regarding our research problems and hypotheses, and we discuss some future works that could follow and deepen our investigation presented in this work.

12.1.1 Low Power networks and Fog Computing

In Chapter 5, we approach networks with Low Power architecture to communicate Smart Cities applications that send few data in low daily frequencies using unlicensed frequencies and independence of external stakeholders such as mobile network operators. In our hypothesis, when using Fog Computing, some limits imposed by the low bandwidth capacity of LPWAN technologies such as LoRa and processing limits of IoT devices can be circumvented.

When mixing technologies with BLE and LoRa, we could see that they have the requirements of low power networks and simplicity of setup and deployment in urban environments, already found in several market solutions.

In our experiments, we did not use the write attributes, only the read attributes. In implementing applications that need to act on BLE devices, it will be necessary to investigate the behavior, prerequisites, and difference of pairing routines.

The freedom of implementation provided by Raw LoRa has its dangerous side. Implementing its gateway for production, like LoRaFog, is challenging beyond simply receiving a payload. Although we do not need to follow the protocol rules imposed by protocols such as LoRaWan, it is necessary to implement the entire security layer and communication routines such as data reception windows. Due to bandwidth limitations and being an intermediary architecture organism before contact with large capacity networks and the cloud, Fog computing is a paradigm to always be considered when processing and bandwidth limitations are at the edges. In our

experiments, we can verify the Long Range capacity of long-range LoRa networks setup links up to 3 km. In the simulation, we could perceive the technology's potential to meet a volume of devices deployed in an urban area, often without a direct target with the gateway. This technical feature of LoRa makes its signal resilient in places that need penetration, vegetation, and underground and that move the IoT device at speed without impacting the doppler effect [208]. These attributes are potential in the context of Smart Cities applications, especially those that transmit short payloads and with some frequency during the day. In LoRa network simulations, we saw limits on the number of simulated nodes, such as the $4,000$ for applications sending payloads every 1 hour to a single gateway and numbers above $20,000$ for multi-gateway scenarios. We conclude that LPWAN technologies such as LoRA have requirements for applications in urban environments. Adding BLE at the edges expands the range of devices available. They also have the potential to compose a network that meets the gateway to the edge with Low Power feature technologies.

12.1.2 Using Blockchain for IoT Security

The COVID-19 pandemic created a need for humanity's survival, in which the scientific csubommunity needed to quickly and effectively seek solutions that would avoid contamination by the virus. This health crisis leads us to a technological acceleration being Blockchain is a disruptive technology that guarantees privacy and security to this new reality.

With the recent conflicts and humanitarian crises experienced, there was a more significant discussion around the dependency on the internet, clean energy sources, use of cryptocurrency to secure wealth. When considering severe pa communication problems in environments such as a city without energy and internet due to the actions of a conflict, these themes lead to some discussions of information and currency control by governments. Even the use of sanctions on some governments to exchange bodies between countries needs to consider that there are already DeFi projects that deal with this problem by smart contract and cryptocurrency. Some refugees from conflict areas are often saved from total loss of heritage by owning crypto asset portfolios.

The IoT and Edge computing are expanded with the emergence and popularization of 5G networks and the perception that centralized models are fragile and often ineffective in extreme environments, such as moments of conflict or war.

In this environment without the internet, the formation of ad-hoc PtoP spontaneous networks can supply the lack of existing internet infrastructure. More use is expected for this independent network and consumption and payment via cryp-

tocurrencies.

SC has created challenges that lead to propositions and a strong need for rapid and intense development of new disruptive application propositions using IoT and Blockchain.

We research approach the SC and IoT scenarios in deep. The core of the discussion is to provide new ways of communicating using networks independent of mobile companies and using city resources with privacy and reliability provided by Blockchain. IoT validation tools in an SC such as the one discussed in our research can be helpful in the subsequent health crises, providing reliable information to the edges for institutions, which can use reliable data with origin identification.

In some SC Apps, government agencies do not provide the device, leaving it to the citizens to acquire. Its behavior has risks, mainly when using devices with unknown technical characteristics and often out of the minimum quality standard for data collection. It motivates to have a previous record of these unknown devices by extracting the hash of their firmware. The hash of firmware during validation can help minimize risks and fraud and show the relevance of delving further into further research using this technique.

Using Blockchain as a security background can help identify and validate data sources outside the SC IoT Apps domain; an example of this would be a certification of the origin of a news item by identifying the author. It is a typical use case that could somewhat mitigate the presence of fake news, which is harmful to our society and generates misinformation.

Blockchain has not yet been widely tested in scenarios outside of cryptocurrencies and financial startups, use cases that involve security and reliability of the origin and integrity of data, make it stand out and be evaluated as a potential solution.

As already discussed in Chapter 7, an investigation and background proposition composing load balancers and clusters for our API gateway can give robustness to the Fog computing perimeter. A gateway in the Fog perimeter using Blockchain for validation is responsible for communication with a decentralized and centralized world. One of the future works in the evolution of our research may be to propose new load balancing and redundancy strategies that would validate our proposal in a large-scale production environment. One of these possibilities would be to integrate our **Blockchain API Gateway** in API Gateways of the cloud industry, such as Apigee, to provide modules and plugins using Smart Contract calls as a background to identify and validate IoT payloads.

An out-of-date firmware on a device opens the door for an IoT device to be subject to confidential data leakage or operational unavailability, a problem in critical environments. False information from these devices can cause credibility crises and financial losses. Our proposal to use API gateways with Blockchain for SC Apps

seeks to mitigate message forgery.

Cyber attacks using legacy or outdated firmware security vulnerabilities likely occupy a good part of the work plan related to IoT security. These vulnerabilities of IoT devices cannot be remedied promptly, partly due to the longer expected life cycle or even because, in some use cases, the environment is so inhospitable that it is impossible to replace or update them quickly.

As proposed by our work, the proposition of device verification using decentralized Blockchain background can be one of the alternatives to be adopted to mitigate this risk. Our set of API that apply an additional layer of security makes our proposition a critical piece to investigate as a solution.

These challenging scenarios foreseen for the coming years justify the relevance of our research and continue to investigate the points open and covered by us with superficiality.

Blockchains like Ethereum have transaction times that still do not allow IoT applications that need streaming data due to the consensus process, especially when evaluating PoW. The decentralized paradigm, as applied by Blockchain, still has a long way to go for popularization beyond DeFi. However, it can currently be used to find applications that want independence from organizations and resilience of infrastructures, such as SC IoT. Blockchain tends to be the disruptive technology with the greatest impact on change in the coming technological cycles due to its security and resilience characteristics.

After our experiments, we conclude with our proposition using API Gateways to validate and authenticate IoT devices. Cryptocurrency technologies such as Ethereum Blockchain can be used for the security and identity of these IoT devices in an SC Using Fog Computing, even when we have scenarios of the use of Consensus in PoW. We came to this conclusion because of the times verified in the transactions when the Blockchain API Gateway meets the requirements required in an IoT SC application environment that sends few payloads during the day and requires accurate identification of the device, its firmware, and destination Web service API.

Although we do not stress the possibilities of attacks in our scenario and thoroughly investigate possible security holes in the implementation and architecture. Our API Gateways is an initial motivator for discussion to provide security and authenticity at the data source from the network's Edge.

12.1.3 Extracting Blockchain Data

Making Ethereum frameworks usable as an integrated web database. In the Chapter 8, we investigate and propose using Semantic Web tools to extract data from a

Blockchain in production. We can conclude that our vocabulary represented by the EthExtras ontology, an extension of a well-known ontology from the literature, EthOn, allows us to create extractors that consume and link data from Ethereum in a standardized way. We conclude that although it has not yet reached its full potential and popularity, the Semantic Web shows rare attributes that standardization of data exposure. Our middleware, not being design specific for data consumption of the SC Apps, can receive new classes added by EthExtras to expose, for example, data from Smart Contracts logs as URI that are to be queried by SPARQL to generate richer data. In the case of Ethereum datasets, it can go far beyond transaction status and current account balances.

12.2 Future Works

Our research addresses problems and identifies an eventual adoption of new strategies and applications within the circle of SC problems. The very definition of SC can still be considered disruptive, and many of its issues meet answers in technological advances related to innovation and the digital transformation of society. During the COVID-19 pandemic, in which it was necessary to evolve the use of digital tools and new ways of working and consuming, we can observe the advance in the use of some technologies addressed in our research even faster. Blockchain, for example, has become a strong background in the culture beyond cryptocurrencies and can already be seen as the background of a complex of solutions called decentralized finance. These financial projects and startups are already viewed using digital transactions and purchases between companies across the globe. We put some possible works that can be investigated from our approach included in this Thesis.

Despite coming as a solution that promises to connect things at high speed and ubiquitously, 5G still in large part of the solutions found and commercialized will be found under centralized coordination and influence of mobile companies, big world tech companies, and governments. This scenario can inhibit and delay new opportunities and applications that depend on the freedom and low cost to form IoT management networks.

Nations that control sensitive information from their citizens use this dependence on centralized media and communication companies for their purposes. This scenario has a growing demand for open and accessible communication technologies and network options. Our motivator to form networks using only open technologies and Low Power, a solution with characteristics and bands with higher throughput can be a search for a solution with characteristics and bands with higher throughput than LoRa networks can be investigated. Options that use a mix of WiFi and Bluetooth networks forming AD-Hoc networks and an optional connection to a current

centralized internet network via satellites can be options and solutions in scenarios of cities in humanitarian crises and conflicts. A testbed in future work can compose spontaneous SC networks using LoRa, Bluetooth, WiFi, and satellite networks as internet last mile. The choice of network link technology depends on the size of data transported and infrastructure availability. A Fog or the Edge device can be responsible for making the link type choice decision and forming an Edge network independent of the internet and its tools.

Spontaneous networks composed of Bluetooth devices at the edges and homes can be low-energy and internet-independent solutions. These spontaneous networks can be used to exchange information between machines and, in case of an emergency, connect to other exponential networks to exchange information external to the residences and belonging to an SC domain, for example.

The benefits of having large volumes of data extracted from SC structures are expected with the popularization of IoT devices. However, their real applicability is still far from reaching their full potential. Much of the communication technologies are still in the hands of large technology companies and mobile phone companies. Even after the long-awaited communication with 5G technology, much of the benefits of having ubiquitous communication and with all things connected, the commercial and financial exploitation of this technology will be in the hands of a few, and the high acquisition costs will delay use by the entire population.

In some home automation solutions, we already see the frequent use of voice assistants that control the devices and receive application installation to respond to games, news, and language teaching. Still, we can't see advanced Apps using the information from devices. We can imagine this evolution in a Fog SC IoT App scenario using our IoT APP proposition being installed in a management center that receives an application capable of handling intelligence and using this data in addition to controlling the devices. An example would be the sale of a kit of pressure and temperature sensors, but according to the installed application, they would present different feedback and insights. In a hypothetical example, we have the IoT App of fire that interprets the presence and temperature. An IoT App of people's flow in a place analyzes the temperature and value of people of another condition. Future work developing an IoT App and testing these possibilities of insights can be new contributions and results to the Fog Computing and AI research area.

When 5g and its costs arrive in this scenario, our research and future related projects can be imagined. One would be to evaluate high-speed 5g links, their energy consumption, and their merging with Low Power networks. Most applications that only transmit information from sensors or receive actuation commands do not need high throughput bands. In this scenario, it is possible to imagine applications that send videos and images using 5G links, and basic sensing information

with Low Power networks, forming a communication infrastructure with a hybrid characteristic.

An example of this type of hybrid SC IoT App is disaster detection. The temperature, humidity, heat, and ground vibration sensors send messages using LPWAN LoRa. If any relevant variation occurs, an image or video would be sent by 5G or satellite link.

This data type segregation by link type is not new in the literature, but a mix of Low Power networks with 5G could minimize the dependence of some applications on proprietary networks, optimize the infrastructure existing structure in addition to reducing acquisition and maintenance costs. By joining all of this to data intelligence centers using Edge and Fog gateways, it is possible to imagine scenarios in which relevant data analysis applying AI could would identify in real-time and in a predictive way pattern of problems and disasters.

Using Bluetooth with automatic sensors links sensors could be a solution in an eventual application that needs to use its device networks privately and without dependence on networks with higher energy consumption and not always available. Bluetooth is widely found in most sensors and electronic devices, with a mobility profile manufactured a few years ago. Bluetooth has in its specification a WPAN formation called Scatternet. Scatternets are not new, and until today there are no relevant applications that use this network formation, leaving WiFi the protagonist role in the wireless communication of the IoT Apps found. Previously collected, WiFi may not be a viable solution depending on the environment and distance between these points. Moreover, think, for example, of a scene outside the SC such as a runner athlete using equipment with wearable heart sensing devices, body temperature, sweat, and interacting with other members of his team and his support base to hydrate and feed following data. Fire and flood detection apps can have sensors interacting and making critical decisions at the edge, using WPAN networks and sending long-distance data via LoRa. Thus, we have the Low Power pattern again with Lora, and BLE investigated a simplified and lower-cost solution in this work.

In this work, we contribute to some topics still largely unexplored in Blockchain, such as the Semantic Web field. The Semantic Web is nothing new. It still faces resistance to its effective popularization, despite its flexibility in standardizing data access through RDF graphs and being a powerful tool in the integration and usability of web databases.

Our investigation and proposition of use of the Semantic Web are still in the field of theory, with our middleware being a use case of our ontology. A future work that integrates the SC and IoT theme would be to create links and ontologies that interact with the current Semantic Sensor Network (SSN) and SC ontologies, bringing a link

and making it possible to develop SC and IoT datasets with integrated Blockchain structures.

We propose the extension of the EthExtras ontology, complete the EthOn and create a model that can be used as a basis for consuming data from Ethereum in a simple way. In future explorations and contributions, we can model ontologies to access the ERC-721 Non-Fungible Token Standard or simply Ethereum NFT, proposing the links with the web datasets that represent the images or works in this ontology. Ontologies describe the ERC-20 Tokens and their effective external integration with their quotes, descriptions, and cryptocurrency trading platforms.

An experiment could be done with an evolution of our middleware that provides Ethereum information through RDF endpoints and uses it by making calls to a full node of the Ethereum MainNet.

Using our middleware to submit the queries to the MainNet's Ethereum full node, we have the queries with primary Ethereum data. Such as transaction status and checking account balances. We can imagine some relevant queries via SPARQL. What are the most popular DApps, listing which Smart Contacts are called in the highest number of transactions and the ERC-20 Tokens, which have more transactions than others? This experiment aims to check if extracting this data from Ethereum using the RDF endpoints is possible.

Our middleware allows us to consume Ethereum data in soft-realtime. We motivated is to propose integrating data from a Blockchain such as Ethereum with applications from the outside world in a standardized way. This future research aims to measure the times required for queries aggregated to Ethereum full node using SPARQL on RDF endpoints. Our middleware and its RDF endpoints represent Ethereum objects that can be used to use and link with external datasets, giving developers the power and flexibility to build queries in a traditional API option like Etherscan cite etherscan. Since this is a future work, we can still add other objects to our model like NFT, Tokens, ENS, IPFS .

Prototyping an Oracle service using our ontologies and middleware could be a proof of concept of using Web Semantica to provide data to real-world services and startups. Our middleware makes it possible to provide a public Ethereum graph dataset, even being able to participate in the Linked Open Data (LOD) [42]. Our research has centered on the SC theme but can be applied to different scenarios and use cases that will see an increase in the use of solutions using IoT, such as I4.0 As discussed in this work, these new applications that need to guarantee data origin can use techniques to identify and guarantee the origin of data from unknown devices. We do not delve into the possibilities of attacks. The integration with other modern cybersecurity techniques is future work to verify that our API Gateways are not in the naive field of security propositions for some use cases.

Investigating the possible security holes in our implementation and architecture can help find a more secure architecture. Our API Gateways are just a starting point for discussion and a motivator in the quest for security and authenticity at the data source from the network's edge.

Smart Contracts functions that only read data on the Blockchain have no significant interference in the time of transactions, and when it is called a read Function, there is no need to pay GAS. The read functions do not change any data on the Ethereum Blockchain. One example is the API Gateway routine that verifies metadata's authenticity by Merkle Tree. We concluded that applications could use smart contracts that do not generate writing to the Blockchain without prejudice. However, future work measuring these times in other scenarios and other use cases would bring accurate numbers of these times and limits for use in real-time scenario applications.

During the research time of this work, open-source Blockchain Ethereum grew and expanded in popularity as a cryptocurrency and today is home to the majority of DApps developed using Smart Contracts. Future work could explore the Fog Computing components of our application as Gateways and middleware in non-Permissioned Blockchains like Hyperledger [209], and Parachains like Polkadot [210].

Future work deploys a testbed of a DApp using new approaches to fully decentralized networking independent of internet standards as ENS, and a good part of the attributes and services needed to develop of its as the decentralized projects like Interplanetary File System (IPFS). This architecture promise to change the paradigm of the next generation of applications. Measuring the impacts and dependencies of this composition of Web3 tools would contribute to understanding the limits of this new proposition.

The tools that the community has developed for Ethereum are continually changing. The version Ethereum 2.0 is coming to the MainNet when the transition from PoW to PoS consense algorithm occurs. It leads us to a work future investigating the new times of replication and formation of new blocks and other impacts, such as the cost of transactions in GAS values.

Because it is open source and one of the pioneers of the Smart Contract, Ethereum has immense adherence to open source developer communities, and an ecosystem of tools has already been found around its cryptocurrency ETH. We only used test Ethereum's networks during our research, like Ropsten, and using the MainNet Ethereum 2.0 can lead to new and unpredictable results because PoS. It leads to the possibility of having experience in using a more recent version of the public Ethereum network in the same models of cryptocurrency Apps.

One of the reasons to explore new experiments in DApp Blockchain is that this has not yet been extensively tested outside of cryptocurrencies in financial

scenarios. These investigations become more relevant when there is a hypothesis that IoT Apps will be massively implemented in the coming years because of the transition of digital advances caused by the COVID19 crisis. This technological acceleration may take SC tools implemented with security models discussed in our research. Furthermore, exploring new approaches to trusting devices IoT may help in the subsequent health crises and natural disasters, providing reliable information from institutions, validating data, and identifying their origins.

In our experiments, we can conclude that Blockchain is permissionless as Ethereum does not allow real-time transactions. However, the robust security of this solution has adherence in applications where the times for sending payloads via IoT have a frequency of hours or days.

Identifying and trusting unknown devices acquired in the SC Apps by the citizens reduces the risks of using devices with unknown technical characteristics. A previous registration and hash extraction of the firmware for validation can help. Our research proposes firmware validation and prototype this, but does not develop in the testbed an extract and validate the firmware on real devices working and in operation; this work would be interesting in future research.

One of the critical points we can address regarding the privacy of configuration files in future works would be to propose techniques to protect the Device Configuration File 7.3 deployed in a IoT device. This file carries the keys, and its capture or exposure is a problem to be worked on in other research, which may use cryptography techniques and tamper-resistant security modules.

This authorization and validation routine by API Gateway can help identify and validate the origin outside the SC IoT apps. An example is the need to verify a news origin and identify the author's post, a common problem in the fight against fake news.

In future work, we will investigate strategies for load balancing and auto-scaling of gateways and middlewares. It is which would validate our proposal in a production environment and future works to investigate the possibility of integrating our solution with the API Gateway of the cloud industry, like Apigee, providing modules and plugins utilizing Blockchain as background.

12.3 Final Conclusions

Although we have SC Apps and scenarios as the motivator of our research, the result of our work would apply to other potential IoT and Blockchain usage scenarios such as Industry 4.0 (I4.0) and Agriculture 4.0. These scenarios and SC need reliability, traceability, and a guarantee of data origin.

Although our work does not emphasize possible cyberattack scenarios, our API

Gateways can be an initial motivator of discussion for architectures with greater sophistication in terms of security regarding the authenticity of the data source coming from the edge of the network.

Smart Contract Ethereum calls that only query data on the Blockchain do not cost GAS and do not generate significant delays. The mining and writing phases of transactions in Blocks are not necessary. Therefore, the message authentication and IoT metadata queries in the Merkle tree have acceptable performance and scalability in non-realtime SC application scenarios. It is possible to conclude that applications can use on a large-scale read call to Blockchain network using Smart Contracts, as they do not generate writing on the Blockchain, and therefore without significant damage to the performance of a city management application.

We can conclude from the transaction times of our testbed that, although we cannot carry out transactions in real-time, the solution has a strong adherence in applications where the times for sending loads via IoT have a frequency of hours or days.

Can use DApps and read calls via Smart Contracts without performance loss. In other words, we do not observe relevant delays in these transactions in applications that only query data on the Blockchain. However, the only thing is that because it is a decentralized network and PtoP feature, delays may occur during replication and the query of recent records data may take a while to be available.

Our experiments conclude that although we cannot carry out transactions in real-time, our solutions have adherence and performance in applications whose times of sending new payloads by IoT devices are in frequencies of hours or days.

Bibliography

[1] GARTNER. "Blockchain Technology: What's Ahead?" Last Visited in 16/05/2022. Disponível em: <https://www.gartner.com/en/information-technology/insights/blockchain>.

[2] GARTNER. "Hype Cycle for Blockchain 2021; More Action than Hype". Last Visited in 16/05/2022. Disponível em: <https://blogs.gartner.com/avivah-litan/2021/07/14/hype-cycle-for-blockchain-2021-more-action-than-hype/>.

[3] AL-FUQAHA, A., GUIZANI, M., MOHAMMADI, M., et al. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", *IEEE Communications Surveys Tutorials*, v. 17, n. 4, pp. 2347–2376, Fourthquarter 2015. ISSN: 1553-877X. doi: 10.1109/COMST.2015.2444095.

[4] CHETTRI, L., BERA, R. "A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems", *IEEE Internet of Things Journal*, v. 7, n. 1, pp. 16–32, 2020. doi: 10.1109/JIOT.2019.2948888.

[5] ALRASHDI, I., ALQAZZAZ, A., ALOUFI, E., et al. "AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning". In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0305–0310, 2019. doi: 10.1109/CCWC.2019.8666450.

[6] SUN, J., YAN, J., ZHANG, K. Z. K. "Blockchain-based sharing services: What blockchain technology can contribute to smart cities", *Financial Innovation*, v. 2, n. 1, pp. 26, Dec 2016. ISSN: 2199-4730. doi: 10.1186/s40854-016-0040-y. Disponível em: <https://doi.org/10.1186/s40854-016-0040-y>.

[7] PEREIRA, C., RODRIGUES, J., PINTO, A., et al. "Smartphones as M2M gateways in smart cities IoT applications". In: *2016 23rd International*

Conference on Telecommunications (ICT), pp. 1–7, May 2016. doi: 10.1109/ICT.2016.7500481.

[8] BARDYN, J. P., MELLY, T., SELLER, O., et al. "IoT: The era of LPWAN is starting now". In: *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, pp. 25–30, Sept 2016. doi: 10.1109/ESSCIRC.2016.7598235.

[9] NEUMANN, P., MONTAVONT, J., NOËL, T. "Indoor deployment of low-power wide area networks (LPWAN): A LoRaWAN case study". In: *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–8, Oct 2016. doi: 10.1109/WiMOB.2016.7763213.

[10] SINAEEPOURFARD, A., GARCIA, J., MASIP-BRUIN, X., et al. "A Novel Architecture for Efficient Fog to Cloud Data Management in Smart Cities". In: *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 2622–2623, 2017. doi: 10.1109/ICDCS.2017.202.

[11] RADANLIEV, P., DE ROURE, D., CANNADY, S., et al. "Economic impact of IoT cyber risk - Analysing past and present to predict the future developments in IoT risk analysis and IoT cyber insurance". In: *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, pp. 1–9, 2018. doi: 10.1049/cp.2018.0003.

[12] ALWIS, L. S. M., BUSTAMANTE, H., BREMER, K., et al. "A pilot study: Evaluation of sensor system design for optical fibre humidity sensors subjected to aggressive air sewer environment". In: *2016 IEEE SENSORS*, pp. 1–3, Oct 2016. doi: 10.1109/ICSENS.2016.7808482.

[13] PETÄJÄJÄRVI, J., MIKHAYLOV, K., HÄMÄLÄINEN, M., et al. "Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring". In: *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, pp. 1–5, March 2016. doi: 10.1109/ISMICT.2016.7498898.

[14] PHAM, C., RAHIM, A., COUSIN, P. "Low-cost, Long-range open IoT for smarter rural African villages". In: *2016 IEEE International Smart Cities Conference (ISC2)*, pp. 1–6, Sept 2016. doi: 10.1109/ISC2.2016.7580823.

[15] VILLARI, M., FAZIO, M., DUSTDAR, S., et al. "Osmotic Computing: A New Paradigm for Edge/Cloud Integration", *IEEE Cloud Computing*, v. 3, n. 6, pp. 76–83, Nov 2016. ISSN: 2325-6095. doi: 10.1109/MCC.2016.124.

[16] VAN ZOONEN, L. "Privacy concerns in smart cities", *Government Information Quarterly*, v. 33, n. 3, pp. 472–480, 2016. ISSN: 0740-624X. doi: https://doi.org/10.1016/j.giq.2016.06.004. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0740624X16300818>. Open and Smart Governments: Strategies, Tools, and Experiences.

[17] KHAN, M. A., SALAH, K. "IoT security: Review, blockchain solutions, and open challenges", *Future Generation Computer Systems*, v. 82, pp. 395–411, 2018. ISSN: 0167-739X. doi: https://doi.org/10.1016/j.future.2017.11.022. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167739X17315765>.

[18] BETTAYEB, M., NASIR, Q., TALIB, M. A. "Firmware Update Attacks and Security for IoT Devices: Survey". In: *Proceedings of the ArabWIC 6th Annual International Conference Research Track*, ArabWIC 2019, New York, NY, USA, 2019. Association for Computing Machinery. ISBN: 9781450360890. doi: 10.1145/3333165.3333169. Disponível em: <https://doi.org/10.1145/3333165.3333169>.

[19] BINTI MOHAMAD NOOR, M., HASSAN, W. H. "Current research on Internet of Things (IoT) security: A survey", *Computer Networks*, v. 148, pp. 283–294, 2019. ISSN: 1389-1286. doi: https://doi.org/10.1016/j.comnet.2018.11.025. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389128618307035>.

[20] DAI, H.-N., ZHENG, Z., ZHANG, Y. "Blockchain for Internet of Things: A Survey", *IEEE Internet of Things Journal*, v. 6, n. 5, pp. 8076–8094, 2019. doi: 10.1109/JIOT.2019.2920987.

[21] SHAMILI, P., MURUGANANTHAM, B. "Enhancing the Decentralized Application (Dapp) for E-commerce by Using the Ethereum Blockchain". In: Mahapatra, R. P., Peddoju, S. K., Roy, S., et al. (Eds.), *Proceedings of International Conference on Recent Trends in Computing*, pp. 679–694, Singapore, 2022. Springer Nature Singapore. ISBN: 978-981-16-7118-0.

[22] WOOD, G., OTHERS. "Ethereum: a secure decentralised generalised transaction ledger (2014)". 2017.

[23] BUTERIN, V. "What is ethereum?" *Ethereum Official webpage. Available: http://www. ethdocs. org/en/latest/introduction/what-is-ethereum. html*, 2016.

[24] GYRARD, A., SERRANO, M., ATEMEZING, G. A. "Semantic web methodologies, best practices and ontology engineering applied to Internet of Things". In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 412–417, 2015. doi: 10.1109/WF-IoT.2015.7389090.

[25] SZILAGYI, I., WIRA, P. "Ontologies and Semantic Web for the Internet of Things - a survey". In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pp. 6949–6954, 2016. doi: 10.1109/IECON.2016.7793744.

[26] GUINARD, D., TRIFA, V. "Towards the Web of Things: Web Mashups for Embedded Devices". In: *WWW 2009*, 2009.

[27] CENA, F., HALLER, A., LEFRANÇOIS, M., et al. "Weather Data Publication on the LOD Using SOSA/SSN Ontology", *Semant. Web*, v. 11, n. 4, pp. 581–591, jan 2020. ISSN: 1570-0844. doi: 10.3233/SW-200375. Disponível em: <https://doi.org/10.3233/SW-200375>.

[28] BENIICHE, A. "A Study of Blockchain Oracles". 2020. Disponível em: <https://arxiv.org/abs/2004.07140>.

[29] FERREIRA, C. M. S., OLIVEIRA, R. A. R., SILVA, J. S. "Low-Energy Smart Cities Network with LoRa and Bluetooth". In: *2019 7th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pp. 24–29, 2019. doi: 10.1109/MobileCloud.2019.00011.

[30] SIMUNIC, S. *Using blockchain for registration and control of IoT devices*, 2018 (accessed December 30, 2020). https://urn.nsk.hr/urn:nbn:hr:190:464395.

[31] VANDIKAS, K., TSIATSIS, V. "Performance Evaluation of an IoT Platform". In: *2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*, pp. 141–146, 2014. doi: 10.1109/NGMAST.2014.66.

[32] FERREIRA, C. M. S., GARROCHO, C. T. B., OLIVEIRA, R. A. R., et al. "IoT Registration and Authentication in Smart City Applications with Blockchain", *Sensors*, v. 21, n. 4, 2021. ISSN: 1424-8220. doi: 10.3390/s21041323. Disponível em: <https://www.mdpi.com/1424-8220/21/4/1323>.

[33] FERREIRA, C., GARROCHO, C., CAVALCANTI, C., et al. "A middleware for systems consumes Ethereum data in soft real-time: a Semantic Web approach". In: *Anais Estendidos do XI Simpósio Brasileiro de Engenharia de Sistemas Computacionais*, pp. 122–127, Porto Alegre, RS, Brasil, 2021. SBC. doi: 10.5753/sbesc_estendido.2021.18503. Disponível em: <https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/18503>.

[34] FERREIRA, C. M. S., OLIVEIRA, R. A. R., SILVA, J. S., et al. "Blockchain for Machine to Machine Interaction in Industry 4.0". In: Rosa Righi, R. d., Alberti, A. M., Singh, M. (Eds.), *Blockchain Technology for Industry 4.0: Secure, Decentralized, Distributed and Trusted Industry Environment*, pp. 99–116, Singapore, Springer Singapore, 2020. ISBN: 978-981-15-1137-0. doi: 10.1007/978-981-15-1137-0_5. Disponível em: <https://doi.org/10.1007/978-981-15-1137-0_5>.

[35] KÕLVART, M., POOLA, M., RULL, A. "Smart Contracts". In: Kerikmäe, T., Rull, A. (Eds.), *The Future of Law and eTechnologies*, pp. 133–147, Cham, Springer International Publishing, 2016. ISBN: 978-3-319-26896-5. doi: 10.1007/978-3-319-26896-5_7. Disponível em: <https://doi.org/10.1007/978-3-319-26896-5_7>.

[36] KUZMIN, A. "Blockchain-based structures for a secure and operate IoT". In: *2017 Internet of Things Business Models, Users, and Networks*, pp. 1–7, Nov 2017. doi: 10.1109/CTTE.2017.8260937.

[37] NOVO, O. "Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT", *IEEE Internet of Things Journal*, v. 5, n. 2, pp. 1184–1195, April 2018. ISSN: 2327-4662. doi: 10.1109/JIOT.2018.2812239.

[38] MILLER, A., CAI, Z., JHA, S. "Smart Contracts and Opportunities for Formal Methods". In: Margaria, T., Steffen, B. (Eds.), *Leveraging Applications of Formal Methods, Verification and Validation. Industrial Practice*, pp. 280–299, Cham, 2018. Springer International Publishing. ISBN: 978-3-030-03427-6.

[39] SWARM.FUND. "swarm.fund". Disponível em: <https://www.swarm.fund/>.

[40] INFURA.IO. "infura.io". Last Visited in 3/03/2022. Disponível em: <https://infura.io/>.

[41] HTTPS://WWW.W3.ORG/2009/TALKS/0615 QBE/. "SPARQL By Example". Last Visited in 30/07/2022. Disponível em: <https://www.w3.org/2009/Talks/0615-qbe/>.

[42] LOD. "The Linked Open Data Cloud". Disponível em: <https://lod-cloud.net/>.

[43] COMPTON, M., BARNAGHI, P., BERMUDEZ, L., et al. "The SSN ontology of the W3C semantic sensor network incubator group", *Journal of Web Semantics*, v. 17, pp. 25 – 32, 2012. ISSN: 1570-8268. doi: https://doi.org/10.1016/j.websem.2012.05.003. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1570826812000571>.

[44] SERRANO, M., QUOC, H. N. M., LE PHUOC, D., et al. "Defining the Stack for Service Delivery Models and Interoperability in the Internet of Things: A Practical Case With OpenIoT-VDK", *IEEE Journal on Selected Areas in Communications*, v. 33, n. 4, pp. 676–689, April 2015. ISSN: 0733-8716. doi: 10.1109/JSAC.2015.2393491.

[45] JANOWICZ, K., HALLER, A., COX, S. J., et al. "SOSA: A lightweight ontology for sensors, observations, samples, and actuators", *Journal of Web Semantics*, v. 56, pp. 1 – 10, 2019. ISSN: 1570-8268. doi: https://doi.org/10.1016/j.websem.2018.06.003. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1570826818300295>.

[46] KOMNINOS, N., BRATSAS, C., KAKDERI, C., et al. "Smart city ontologies: Improving the effectiveness of smart city applications", *Journal of Smart Cities*, v. 1, n. 1, pp. 31–46, 2019. ISSN: 23826401.

[47] THETHINGSNETWORK.ORG. "The Things Network". Last Visited in 07/07/2018. Disponível em: <https://www.thethingsnetwork.org>.

[48] SAARI, M., BIN BAHARUDIN, A. M., SILLBERG, P., et al. "LoRa - A survey of recent research trends". In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 0872–0877, May 2018. doi: 10.23919/MIPRO.2018.8400161.

[49] NOREEN, U., BOUNCEUR, A., CLAVIER, L. "A study of LoRa low power and wide area network technology". In: *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pp. 1–6, 2017. doi: 10.1109/ATSIP.2017.8075570.

[50] RATHORE, M. M., AHMAD, A., PAUL, A. "IoT-based smart city development using big data analytical approach". In: *2016 IEEE International Conference on Automatica (ICA-ACCA)*, pp. 1–8, Oct 2016. doi: 10.1109/ICA-ACCA.2016.7778510.

[51] PETRIĆ, T., GOESSENS, M., NUAYMI, L., et al. "Measurements, performance and analysis of LoRa FABIAN, a real-world implementation of LPWAN". In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–7, Sept 2016. doi: 10.1109/PIMRC.2016.7794569.

[52] GEORGIOU, O., RAZA, U. "Low Power Wide Area Network Analysis: Can LoRa Scale?" *IEEE Wireless Communications Letters*, v. PP, n. 99, pp. 1–1, 2017. ISSN: 2162-2337. doi: 10.1109/LWC.2016.2647247.

[53] ZHAO, W., LIN, S., HAN, J., et al. "Design and Implementation of Smart Irrigation System Based on LoRa". In: *2017 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, Dec 2017. doi: 10.1109/GLOCOMW.2017.8269115.

[54] BOR, M. C., ROEDIG, U., VOIGT, T., et al. "Do LoRa Low-Power Wide-Area Networks Scale?" In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '16, pp. 59–67, New York, NY, USA, 2016. ACM. ISBN: 978-1-4503-4502-6. doi: 10.1145/2988287.2989163. Disponível em: <http://doi.acm.org/10.1145/2988287.2989163>.

[55] TSAI, P. H., HONG, H. J., CHENG, A. C., et al. "Distributed analytics in fog computing platforms using tensorflow and kubernetes". In: *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 145–150, Sept 2017. doi: 10.1109/APNOMS.2017.8094194.

[56] VENANZI, R., KANTARCI, B., FOSCHINI, L., et al. "MQTT-Driven Node Discovery for Integrated IoT-Fog Settings Revisited: The Impact of Advertiser Dynamicity". In: *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pp. 31–39, March 2018. doi: 10.1109/SOSE.2018.00013.

[57] BERDIK, D., OTOUM, S., SCHMIDT, N., et al. "A Survey on Blockchain for Information Systems Management and Security", *Information Processing Management*, v. 58, n. 1, pp. 102397, 2021. ISSN: 0306-4573. doi: https://doi.org/10.1016/j.ipm.2020.102397. Disponível em: <http://www.sciencedirect.com/science/article/pii/S030645732030892X>.

[58] TSENG, L., YAO, X., OTOUM, S., et al. "Blockchain-based database in an IoT environment: challenges, opportunities, and analysis", *Cluster Computing*, v. 23, n. 3, pp. 2151–2165, Sep 2020. ISSN: 1573-7543. doi: 10.1007/s10586-020-03138-7. Disponível em: <https://doi.org/10.1007/s10586-020-03138-7>.

[59] CHRISTIDIS, K., DEVETSIKIOTIS, M. "Blockchains and Smart Contracts for the Internet of Things", *IEEE Access*, v. 4, pp. 2292–2303, 2016. ISSN: 2169-3536. doi: 10.1109/ACCESS.2016.2566339.

[60] BISWAS, K., MUTHUKKUMARASAMY, V. "Securing Smart Cities Using Blockchain Technology". In: *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 1392–1393, Dec 2016. doi: 10.1109/HPCC-SmartCity-DSS.2016.0198.

[61] POLYZOS, G. C., FOTIOU, N. "Blockchain-Assisted Information Distribution for the Internet of Things". In: *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 75–78, Aug 2017. doi: 10.1109/IRI.2017.83.

[62] SINGH, P., NAYYAR, A., KAUR, A., et al. "Blockchain and Fog Based Architecture for Internet of Everything in Smart Cities", *Future Internet*, v. 12, n. 4, pp. 61, Mar 2020. ISSN: 1999-5903. doi: 10.3390/fi12040061. Disponível em: <http://dx.doi.org/10.3390/fi12040061>.

[63] REHAN, M., REHMANI, M. *Blockchain-enabled Fog and Edge Computing: Concepts, Architectures and Applications: Concepts, Architectures and Applications*. CRC Press, 2020. ISBN: 9781000096026. Disponível em: <https://books.google.com.br/books?id=JprzDwAAQBAJ>.

[64] AL RIDHAWI, I., ALOQAILY, M., JARARWEH, Y. "An Incentive-Based Mechanism for Volunteer Computing Using Blockchain", *ACM Trans. Internet Technol.*, v. 21, n. 4, jul 2021. ISSN: 1533-5399. doi: 10.1145/3419104. Disponível em: <https://doi.org/10.1145/3419104>.

[65] YU, K.-P., TAN, L., ALOQAILY, M., et al. "Blockchain-Enhanced Data Sharing with Traceable and Direct Revocation in IIoT", *IEEE Transactions on Industrial Informatics*, v. PP, pp. 1–1, 01 2021. doi: 10.1109/TII.2021.3049141.

[66] RIDHAWI, I. A., OTOUM, S., ALOQAILY, M., et al. "Generalizing AI: Challenges and Opportunities for Plug and Play AI Solutions", *IEEE Network*, pp. 1–8, 2020. doi: 10.1109/MNET.011.2000371.

[67] GAI, K., WU, Y., ZHU, L., et al. "Differential Privacy-Based Blockchain for Industrial Internet-of-Things", *IEEE Transactions on Industrial Informatics*, v. 16, n. 6, pp. 4156–4165, 2020. doi: 10.1109/TII.2019.2948094.

[68] SIMUNIC, S. *IoT Device Management*, 2019 (accessed December 30, 2020). `https://github.com/ssimunic/iot-device-management`.

[69] CANO-BENITO, J., CIMMINO, A., GARCÍA-CASTRO, R. "Towards Blockchain and Semantic Web". In: Abramowicz, W., Corchuelo, R. (Eds.), *Business Information Systems Workshops*, pp. 220–231, Cham, 2019. Springer International Publishing. ISBN: 978-3-030-36691-9.

[70] OLIVÉ, A. "The Conceptual Schema of Ethereum". In: Dobbie, G., Frank, U., Kappel, G., et al. (Eds.), *Conceptual Modeling*, pp. 418–428, Cham, 2020. Springer International Publishing. ISBN: 978-3-030-62522-1.

[71] HECTOR, U.-R., BORIS, C.-L. "BLONDiE: Blockchain Ontology with Dynamic Extensibility". 2020.

[72] THIRD, A., DOMINGUE, J. "Linked Data Indexing of Distributed Ledgers". In: *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, p. 1431–1436, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. ISBN: 9781450349147. doi: 10.1145/3041021.3053895. Disponível em: <`https://doi.org/10.1145/3041021.3053895`>.

[73] GRAUX, D., SEJDIU, G., JABEEN, H., et al. "Profiting from Kitties on Ethereum: Leveraging Blockchain RDF with SANSA". In: *14th International Conference on Semantic Systems, Poster & Demos*, 2018. Disponível em: <`http://jens-lehmann.org/files/2018/semantics_ethereum_pd.pdf`>.

[74] ALETH.IO. "Aleth.io". Disponível em: <`https://docs.aleth.io/`>.

[75] RUTA, M., SCIOSCIA, F., IEVA, S., et al. "Supply Chain Object Discovery with Semantic-Enhanced Blockchain". In: *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, SenSys '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN: 9781450354592. doi: 10.1145/3131672.3136974. Disponível em: <`https://doi.org/10.1145/3131672.3136974`>.

[76] ANDROULAKI, E., BARGER, A., BORTNIKOV, V., et al. "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains". In: *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, pp. 30:1–30:15, New York, NY, USA, 2018. ACM. ISBN: 978-1-4503-5584-1. doi: 10.1145/3190508.3190538. Disponível em: <http://doi.acm.org/10.1145/3190508.3190538>.

[77] UGARTE, H. "A more pragmatic Web 3.0: Linked Blockchain Data". Disponível em: <https://www.researchgate.net/publication/315619465_A_more_pragmatic_Web_30_Linked_Blockchain_Data>.

[78] CIMMINO, A., GARCÍA-CASTRO, R., CANO-BENITO, J. "Benchmarking the efficiency of RDF-based access for blockchain environments". In: *SEKE*, 2020.

[79] WEB3.PY. "Web3.py". Disponível em: <https://web3py.readthedocs.io/en/stable/#>.

[80] SOPEK, M., GRADZKI, P., KOSOWSKI, W., et al. "GraphChain: A Distributed Database with Explicit Semantics and Chained RDF Graphs". In: *Companion Proceedings of the The Web Conference 2018*, WWW '18, p. 1171–1178, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN: 9781450356404. doi: 10.1145/3184558.3191554. Disponível em: <https://doi.org/10.1145/3184558.3191554>.

[81] HELIUM. "helium.com". Last Visited in 3/03/2022. Disponível em: <https://www.helium.com/>.

[82] REYNA, A., MARTÍN, C., CHEN, J., et al. "On blockchain and its integration with IoT. Challenges and opportunities", *Future Generation Computer Systems*, v. 88, pp. 173 – 190, 2018. ISSN: 0167-739X. doi: https://doi.org/10.1016/j.future.2018.05.046. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167739X17329205>.

[83] CHRONICLED. "Chronicled". Disponível em: <https://www.chronicled.com/>.

[84] AEROTOKEN.COM. "aerotoken.com". Last Visited in 07/04/2019. Disponível em: <https://aerotoken.com>.

[85] SHARMA, V., YOU, I., KUL, G. "Socializing Drones for Inter-Service Operability in Ultra-Dense Wireless Networks Using Blockchain". In: *Pro-*

ceedings of the 2017 International Workshop on Managing Insider Security Threats, MIST '17, pp. 81–84, New York, NY, USA, 2017. ACM. ISBN: 978-1-4503-5177-5. doi: 10.1145/3139923.3139932. Disponível em: <http://doi.acm.org/10.1145/3139923.3139932>.

[86] CHAINOFTHINGS.COM. "The Chain of Things". Last Visited in 07/04/2019. Disponível em: <https://www.chainofthings.com>.

[87] IBM. "ADEPT: An IoT Practitioner Perspective". Last Visited in 07/04/2019. Disponível em: <https://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/55f73e5ee4b09b2bff5b2eca/55f73e72e4b09b2bff5b3267/1442266738638/IBM-ADEPT-Practictioner-Perspective-Pre-Publication-Draft-7-Jan-2015.pdf?format=original>.

[88] SAMANIEGO, M., DETERS, R. "Blockchain as a Service for IoT", *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 433–436, 2016.

[89] SERGIO F. T. DE O. MENDONCA, J. F. D. S. J., DE ALENCAR, F. M. R. "The Blockchain-based Internet of Things Development: Initiatives and Challenges". Disponível em: <https://www.thinkmind.org/download.php?articleid=icsea_2017_2_20_10012>.

[90] ZHENG, Z., XIE, S., CHEN, X., et al. "Blockchain challenges and opportunities: a survey", *IJWGS*, v. 14, pp. 352–375, 2018.

[91] MYBIT. "mybit.io". Disponível em: <https://mybit.io/>.

[92] SLOCK.IT. "slock.it". Last Visited in 07/04/2019. Disponível em: <https://slock.it>.

[93] BRADBURY, D. "Blockchainapos;s big deal [financial IT]", *Engineering Technology*, v. 11, pp. 44–44(0), November 2016. ISSN: 1750-9637. Disponível em: <https://digital-library.theiet.org/content/journals/10.1049/et.2016.1003>.

[94] HUCKLE, S., BHATTACHARYA, R., WHITE, M., et al. "Internet of Things, Blockchain and Shared Economy Applications", *Procedia Computer Science*, v. 98, pp. 461 – 466, 2016. ISSN: 1877-0509. doi: https://doi.org/10.1016/j.procs.2016.09.074. Disponível

em: <http://www.sciencedirect.com/science/article/pii/S1877050916322190>. The 7th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2016)/The 6th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2016)/Affiliated Workshops.

[95] AIGANG. "Autonomous insurance network - fully automated insurance for IoT devices and a platform for insurance innovation built around data". Disponível em: <https://docs.google.com/document/d/1tJPnmI1_6HbhSASVt8b8OlRUrvSrso9Y7FGG--KkXRI/edit>.

[96] ALI, M. S., DOLUI, K., ANTONELLI, F. "IoT Data Privacy via Blockchains and IPFS". In: *Proceedings of the Seventh International Conference on the Internet of Things*, IoT '17, pp. 14:1–14:7, New York, NY, USA, 2017. ACM. ISBN: 978-1-4503-5318-2. doi: 10.1145/3131542.3131563. Disponível em: <http://doi.acm.org/10.1145/3131542.3131563>.

[97] NAIK, D. R., DAS, L. B., BINDIYA, T. S. "Wireless Sensor networks with Zigbee and WiFi for Environment Monitoring, Traffic Management and Vehicle Monitoring in Smart Cities". In: *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, pp. 46–50, 2018. doi: 10.1109/CCCS.2018.8586819.

[98] GARCÍA-GARCÍA, L., JIMÉNEZ, J. M., ABDULLAH, M. T. A., et al. "Wireless technologies for IoT in smart cities", *Network Protocols and Algorithms*, v. 10, n. 1, pp. 23–64, 2018.

[99] EGLI, P. R. "LPWAN - OVERVIEW OF EMERGING TECHNOLOGIES FOR LOW POWER WIDE AREA NETWORKS IN INTERNET OF THINGS AND M2M SCENARIOS". Disponível em: <https://indigoo.com/dox/itdp/12_MobileWireless/LPWAN.pdf>.

[100] PANDORAFMS. "Waht is LPWAN". Disponível em: <https://pandorafms.com/blog/what-is-lpwan/#>.

[101] EBI, C., SCHALTEGGER, F., RÜST, A., et al. "Synchronous LoRa Mesh Network to Monitor Processes in Underground Infrastructure", *IEEE Access*, v. 7, pp. 57663–57677, 2019. doi: 10.1109/ACCESS.2019.2913985.

[102] AGIWAL, M., ROY, A., SAXENA, N. "Next Generation 5G Wireless Networks: A Comprehensive Survey", *IEEE Communications Surveys Tu-*

torials, v. 18, n. 3, pp. 1617–1655, 2016. doi: 10.1109/COMST.2016.2532458.

[103] MORGADO, A., HUQ, K. M. S., MUMTAZ, S., et al. "A survey of 5G technologies: regulatory, standardization and industrial perspectives", *Digital Communications and Networks*, v. 4, n. 2, pp. 87–97, 2018. ISSN: 2352-8648. doi: https://doi.org/10.1016/j.dcan.2017.09.010. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352864817302584>.

[104] LORENZ MEIER, MYLÈNE F. JACQUEMART, B. B. B. A. "Real-Time Avalanche Detection with Long-Range, Wide-Angle Radars for Road Safety in Zermatt, Switzerland". Last Visited in 3/09/2022. Disponível em: <https://arc.lib.montana.edu/snow-science/item/2284>.

[105] YANG, C., LIANG, P., FU, L., et al. "Using 5G in smart cities: A systematic mapping study", *Intelligent Systems with Applications*, v. 14, pp. 200065, 2022. ISSN: 2667-3053. doi: https://doi.org/10.1016/j.iswa.2022.200065. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2667305322000060>.

[106] KHAN, F., PI, Z., RAJAGOPAL, S. "Millimeter-wave mobile broadband with large scale spatial processing for 5G mobile communication". In: *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1517–1523, 2012. doi: 10.1109/Allerton.2012.6483399.

[107] BLUETOOTH.COM. "Bluetooth.com". Last Visited in 3/09/2022. Disponível em: <https://www.bluetooth.com/blog/wireless-connectivity-options-for-iot-applications-technology-comparison/>

[108] KHAREL, J., REDA, H. T., SHIN, S. Y. "Fog Computing-Based Smart Health Monitoring System Deploying LoRa Wireless Communication", *IETE Technical Review*, v. 36, n. 1, pp. 69–82, 2019. doi: 10.1080/02564602.2017.1406828. Disponível em: <https://doi.org/10.1080/02564602.2017.1406828>.

[109] ZAHMATKESH, H., AL-TURJMAN, F. "Fog computing for sustainable smart cities in the IoT era: Caching techniques and enabling technologies - an overview", *Sustainable Cities and Society*, v. 59, pp. 102139, 2020. ISSN: 2210-6707. doi: https://doi.org/10.1016/j.scs.2020.102139. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2210670720301268>.

[110] CHA, S., CHEN, J., SU, C., et al. "A Blockchain Connected Gateway for BLE-Based Devices in the Internet of Things", *IEEE Access*, v. 6, pp. 24639–24649, 2018. ISSN: 2169-3536. doi: 10.1109/ACCESS.2018.2799942.

[111] C.M.S.FERREIRA. "LoRaBLE". Last Visited in 19/09/2022. Disponível em: <https://github.com/celiomarcio/lorable>.

[112] HTTPS://WWW.LORASERVER.IO. "Lora Server". Last Visited in 30/11/2018. Disponível em: <https://www.loraserver.io>.

[113] HTTPS://GITHUB.COM/PYCOM/PYCOM-LIBRARIES/TREE/MASTER/EXAMPLES/L NANO GATEWAY. "LoraWan Nano Gateway". Last Visited in 30/11/2018. Disponível em: <https://github.com/pycom/pycom-libraries/tree/master/examples/lorawan-nano-gateway>.

[114] LORASIM. "LoRa Simulator". Disponível em: <https://www.lancaster.ac.uk/scc/sites/lora/lorasim.html>.

[115] C.M.S.FERREIRA. "lorasimlogs". Last Visited in 19/09/2022. Disponível em: <https://drive.google.com/drive/folders/1SF5aRCvSqh33P3hVwVQPidkNEuUambxm?usp=sharing>.

[116] CHINCHILLA-ROMERO, N., NAVARRO-ORTIZ, J., MUÑOZ, P., et al. "Collision Avoidance Resource Allocation for LoRaWAN", *Sensors (Basel)*, v. 21, n. 4, fev. 2021.

[117] BLOG.SEMANTECH.COM. "LoRa combined with BLE creates complementary hybrid iot connectivity". Last Visited in 07/04/2019. Disponível em: <https://blog.semtech.com/lora-combined-with-ble-creates-complementary-hybrid-iot-connectivity>.

[118] CALBIMONTE, J.-P., SARNI, S., EBERLE, J., et al. "XGSN: An Open-source Semantic Sensing Middleware for the Web of Things", 2014.

[119] ZHANG, R., XUE, R., LIU, L. "Security and Privacy on Blockchain", *ACM Comput. Surv.*, v. 52, n. 3, jul 2019. ISSN: 0360-0300. doi: 10.1145/3316481. Disponível em: <https://doi.org/10.1145/3316481>.

[120] HANADA, Y., HSIAO, L., LEVIS, P. "Smart Contracts for Machine-to-Machine Communication: Possibilities and Limitations". In: *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, pp. 130–136, 2018. doi: 10.1109/IOTAIS.2018.8600854.

[121] IBBA, S., PINNA, A., SEU, M., et al. "CitySense: Blockchain-Oriented Smart Cities". In: *Proceedings of the XP2017 Scientific Workshops*, XP '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN: 9781450352642. doi: 10.1145/3120459.3120472. Disponível em: <https://doi.org/10.1145/3120459.3120472>.

[122] TAHMASEBI, S., HABIBI, J., SHAMSAIE, A. "A Scalable Architecture for Monitoring IoT Devices Using Ethereum and Fog Computing". In: *2020 4th International Conference on Smart City, Internet of Things and Applications (SCIOT)*, pp. 66–76, 2020. doi: 10.1109/SCIOT50840.2020.9250193.

[123] GARCIA-FONT, V. "SocialBlock: An architecture for decentralized user-centric data management applications for communications in smart cities", *Journal of Parallel and Distributed Computing*, v. 145, pp. 13 – 23, 2020. ISSN: 0743-7315. doi: https://doi.org/10.1016/j.jpdc.2020.06.004. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0743731520303117>.

[124] DE OLIVEIRA NETO, J. S., KOFUJI, S. T. "Inclusive Smart City: Expanding design possibilities for persons with disabilities in the urban space". In: *2016 IEEE International Symposium on Consumer Electronics (ISCE)*, pp. 59–60, Sept 2016. doi: 10.1109/ISCE.2016.7797370.

[125] HUH, S., CHO, S., KIM, S. "Managing IoT devices using blockchain platform". In: *2017 19th International Conference on Advanced Communication Technology (ICACT)*, pp. 464–467, Feb 2017. doi: 10.23919/ICACT.2017.7890132.

[126] KUMAR, P., KUMAR, R., GUPTA, G. P., et al. "A Distributed framework for detecting DDoS attacks in smart contract-based Blockchain-IoT Systems by leveraging Fog computing", *Transactions on Emerging Telecommunications Technologies*, v. n/a, n. n/a, pp. e4112. doi: https://doi.org/10.1002/ett.4112. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4112>.

[127] IBARRA, M. J., ALCARRAZ, E., TAPIA, O., et al. "NFT-I technique using IoT to improve hydroponic cultivation of lettuce". In: *2020 39th International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1–7, 2020. doi: 10.1109/SCCC51225.2020.9281277.

[128] BANERJEE, M., LEE, J., CHOO, K.-K. R. "A blockchain future for internet of things security: a position paper", *Digital Communications and Net-*

works, v. 4, n. 3, pp. 149 – 160, 2018. ISSN: 2352-8648. doi: https://doi.org/10.1016/j.dcan.2017.10.006. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2352864817302900>.

[129] PAN, J., WANG, J., HESTER, A., et al. "EdgeChain: An Edge-IoT Framework and Prototype Based on Blockchain and Smart Contracts", *arXiv e-prints*, art. arXiv:1806.06185, Jun 2018.

[130] NAZ, S., LEE, S. U.-J. "Why the new consensus mechanism is needed in blockchain technology?" In: *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*, pp. 92–99, 2020. doi: 10.1109/BCCA50787.2020.9274461.

[131] YUN, J., GOH, Y., CHUNG, J.-M. "Analysis of Mining Performance Based on Mathmatical Approach of PoW". In: *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–2, 2019. doi: 10.23919/ELINFOCOM.2019.8706374.

[132] CHICAIZA, S. A. Y., CHAFLA, C. N. S., ÁLVAREZ, L. F. E., et al. "Analysis of information security in the PoW (Proof of Work) and PoS (Proof of Stake)blockchain protocols as an alternative for handling confidential nformation in the public finance ecuadorian sector". In: *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–5, 2021. doi: 10.23919/CISTI52073.2021.9476382.

[133] PAWAR, A., BARTHARE, D., RAWAT, N., et al. "BlockAudit 2.0: PoA blockchain based solution for secure Audit logs". In: *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, pp. 1–6, 2021. doi: 10.1109/ISCON52037.2021.9702378.

[134] HIMANSHI. "Proof of Capacity (PoC) in Blockchain". Last Visited in 5/02/2022. Disponível em: <https://www.naukri.com/learning/articles/proof-of-capacity-in-blockchain/>.

[135] PAUL MERRILL, THOMAS AUSTIN, J. T. Y. P. J. R. "Lock and Load: A Model for Free Blockchain Transactions through Token Locking". In: *Proceedings of 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, DAPPCON 2019, 2019. ISBN: 978-1-7281-1264-0. doi: 10.1109/DAPPCON.2019.00013.

[136] JAYABALAN, J., N, J. "A Study on Distributed Consensus Protocols and Algorithms: The Backbone of Blockchain Networks". In: *2021 International*

Conference on Computer Communication and Informatics (ICCCI), pp. 1–10, 2021. doi: 10.1109/ICCCI50826.2021.9402318.

[137] BEZ, M., FORNARI, G., VARDANEGA, T. "The scalability challenge of ethereum: An initial quantitative analysis". In: *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pp. 167–176, 2019. doi: 10.1109/SOSE.2019.00031.

[138] HARRIS, C. G. "The Risks and Challenges of Implementing Ethereum Smart Contracts". In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 104–107, 2019. doi: 10.1109/BLOC.2019.8751493.

[139] PUSTIŠEK, M., KOS, A. "Approaches to Front-End IoT Application Development for the Ethereum Blockchain", *Procedia Computer Science*, v. 129, pp. 410 – 419, 2018. ISSN: 1877-0509. doi: https://doi.org/10.1016/j.procs.2018.03.017. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050918302308>. 2017 INTERNATIONAL CONFERENCE ON IDENTIFICATION,INFORMATION AND KNOWLEDGEIN THE INTERNET OF THINGS.

[140] WEB3.JS. "Web3.js". Disponível em: <https://web3js.readthedocs.io/en/v1.7.3/>.

[141] SAYED, A. I. E., AZIZ, M. A., AZEEM, M. H. A. "Blockchain Decentralized IoT Trust Management". In: *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, pp. 1–6, 2020. doi: 10.1109/3ICT51146.2020.9311998.

[142] HELLANI, H., SLIMAN, L., SAMHAT, A. E., et al. "Tangle the Blockchain:Towards Connecting Blockchain and DAG". In: *2021 IEEE 30th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 63–68, 2021. doi: 10.1109/WETICE53228.2021.00023.

[143] MARCIO, C., Last Visited in 28/09/2022. Disponível em: <https://github.com/celiomarcio/fogmqttethereum>.

[144] ZOLLER, T., Last Visited in 28/09/2022. Disponível em: <https://github.com/javahippie/geth-dev.git>.

[145] SOLIDITY, Last Visited in 28/09/2022. Disponível em: <https://docs.soliditylang.org/en/v0.8.17/>.

[146] MOSQUITTO, E., Last Visited in 28/09/2022. Disponível em: <https://mosquitto.org/>.

[147] MARCIO, C., Last Visited in 28/09/2022. Disponível em: <https://docs.google.com/spreadsheets/d/13bDiOVd8CsA3ROn5vb-0afnTOn-WnZRLFUS4NIYnwNQ/edit?usp=sharing>.

[148] ASHIK, M. H., MASWOOD, M. M. S., ALHARBI, A. G. "Designing a Fog-Cloud Architecture using Blockchain and Analyzing Security Improvements". In: *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1–6, 2020. doi: 10.1109/ICECCE49384.2020.9179374.

[149] KHALEEJTIMES. "khaleejtimes.com". Disponível em: <https://www.khaleejtimes.com/nation/dubai/dubai-to-embrace-blockchain>.

[150] KARATURK, E., KOCYIGIT, E. "Security Concepts in Smart Cities". In: *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–7, 2020. doi: 10.1109/HORA49412.2020.9152605.

[151] NAWAZ, A., PEÑA QUERALTA, J., GUAN, J., et al. "Edge Computing to Secure IoT Data Ownership and Trade with the Ethereum Blockchain", *Sensors*, v. 20, n. 14, pp. 3965, Jul 2020. ISSN: 1424-8220. doi: 10.3390/s20143965. Disponível em: <http://dx.doi.org/10.3390/s20143965>.

[152] TAHMASEBI, S., HABIBI, J., SHAMSAIE, A. "A Scalable Architecture for Monitoring IoT Devices Using Ethereum and Fog Computing". 2020.

[153] REN, Q., MAN, K. L., LI, M., et al. "Using Blockchain to Enhance and Optimize IoT-based Intelligent Traffic System". In: *2019 International Conference on Platform Technology and Service (PlatCon)*, pp. 1–4, 2019. doi: 10.1109/PlatCon.2019.8669412.

[154] RAHMAN, M. A., RASHID, M. M., HOSSAIN, M. S., et al. "Blockchain and IoT-Based Cognitive Edge Framework for Sharing Economy Services in a Smart City", *IEEE Access*, v. 7, pp. 18611–18621, 2019. doi: 10.1109/ACCESS.2019.2896065.

[155] ARLI DENNI, VAN ESCH PATRICK, B. M. L. A. "Do consumers really trust cryptocurrencies?" *Marketing Intelligence Planning*, 2020. doi: 10.1108/MIP-01-2020-0036.

[156] MISTRY, I., TANWAR, S., TYAGI, S., et al. "Blockchain for 5G-enabled IoT for industrial automation: A systematic review, solutions, and challenges", *Mechanical Systems and Signal Processing*, v. 135, pp. 106382, 2020. ISSN: 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2019.106382. Disponível em: <http://www.sciencedirect.com/science/article/pii/S088832701930603X>.

[157] HAKAK, S., KHAN, W. Z., GILKAR, G. A., et al. "Securing Smart Cities through Blockchain Technology: Architecture, Requirements, and Challenges", *IEEE Network*, v. 34, n. 1, pp. 8–14, 2020. doi: 10.1109/MNET.001.1900178.

[158] METAMASK.IO. *Metamask Crypto Wallet*, 2020 (accessed December 30, 2020). https://metamask.io.

[159] WEB3.JS. *web3.js - Ethereum Javascript API*, 2020 (accessed December 30, 2020). https://github.com/ssimunic/iot-device-management.

[160] ENGINE, I.-F. *IoT-Framework Engine*, 2013 (accessed December 30, 2020). https://github.com/EricssonResearch/iot-framework-engine.

[161] IOT-FRAMEWORK GUI. *iot-framework-gui*, 2013 (accessed December 30, 2020). https://github.com/EricssonResearch/iot-framework-gui.

[162] TRUFFLESUITE.COM. "trufflesuite.com". Last Visited in 3/03/2022. Disponível em: <https://trufflesuite.com/>.

[163] FERREIRA, C. M. S. *Code of the work, IoT Registration and Authentication in Smart City Applications using Blockchain*, 2020 (accessed December 30, 2020). https://github.com/celiomarcio/iotregauthbc.

[164] FERREIRA, C. M. S. *Deploy version of IoT Device managemente*, 2020 (accessed December 30, 2020). https://bciotmanager.herokuapp.com.

[165] NOTHEISEN, B., HAWLITSCHEK, F., WEINHARDT, C. "Breaking down the Blockchain Hype - towards a Blockchain Market Engineering Approach". In: *ECIS*, 2017.

[166] PETERS, G. W., PANAYI, E. "Understanding Modern Banking Ledgers Through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money". In: Tasca, P., Aste, T., Pelizzon, L., et al. (Eds.), *Banking Beyond Banks and Money: A Guide to Banking Services in the Twenty-First Century*, pp. 239–278, Cham, Springer International Publishing, 2016. ISBN: 978-3-319-42448-4. doi:

10.1007/978-3-319-42448-4_13. Disponível em: <https://doi.org/10.1007/978-3-319-42448-4_13>.

[167] BHUTTA, M. N. M., KHWAJA, A. A., NADEEM, A., et al. "A Survey on Blockchain Technology: Evolution, Architecture and Security", *IEEE Access*, v. 9, pp. 61048–61073, 2021. doi: 10.1109/ACCESS.2021.3072849.

[168] ETHON. "EthOn: Ethereum Ontology". Disponível em: <https://ethon.consensys.net/>.

[169] SPECIFICATION, E. "EthOn Specification". Disponível em: <https://consensys.github.io/EthOn/EthOn_spec.html>.

[170] ONTOLOGY, E. "EthExtras Ontology". Disponível em: <https://ethon.herokuapp.com/ethextras.owl#>.

[171] UNIVERSITY, S. "Protégé". Disponível em: <https://protege.stanford.edu/>.

[172] FLASK.PALLETSPROJECTS.COM. "Flask". Last Visited in 07/04/2022. Disponível em: <https://flask.palletsprojects.com/en/2.1.x/>.

[173] RDFLIB.READTHEDOCS.IO. "RDFLib". Last Visited in 07/04/2022. Disponível em: <https://rdflib.readthedocs.io/en/stable/>.

[174] HTTPS://GITHUB.COM/CELIOMARCIO/SEMANTICETHON. "https://github.com/celiomarcio/semanticethon". Disponível em: <https://ethon.herokuapp.com/>.

[175] HTTPS://ETHON.HEROKUAPP.COM/. "https://ethon.herokuapp.com/". Disponível em: <https://ethon.herokuapp.com/>.

[176] SUNY, M. F. I., FAHIM, M. M. R., RAHMAN, M., et al. "IoT Past, Present, and Future a Literary Survey". In: *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*, Springer, pp. 393–402, 2021.

[177] RESEARCH, G. V. "Industrial Internet of Things (IIoT) Market Size, Share Trends Analysis Report By Component, By End Use (Manufacturing, Energy Power, Oil Gas, Healthcare, Logistics Transport, Agriculture), And Segment Forecasts, 2019 - 2025". Available: https://www.grandviewresearch.com/industry-analysis/industrial-internet-of-things-iiot-market., 2019.

[178] KHAN, M. A., SALAH, K. "IoT security: Review, blockchain solutions, and open challenges", *Future generation computer systems*, v. 82, pp. 395–411, 2018.

[179] ZHONG, S., ZHONG, H., HUANG, X., et al. *Security and Privacy for Next-Generation Wireless Networks*. Springer, 2019.

[180] NIŽETIĆ, S., ŠOLIĆ, P., GONZÁLEZ-DE, D. L.-D.-I., et al. "Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future", *Journal of Cleaner Production*, v. 274, pp. 122877, 2020.

[181] KHAN, M., WU, X., XU, X., et al. "Big data challenges and opportunities in the hype of Industry 4.0". In: *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6. IEEE, 2017.

[182] CHEN, M., MAO, S., LIU, Y. "Big data: A survey", *Mobile networks and applications*, v. 19, n. 2, pp. 171–209, 2014.

[183] DE SOUZA MOREIRA, F., LOPES, M. P. C., DE FREITAS, M. A. V., et al. "Future scenarios for the development of the desalination industry in contexts of water scarcity: A Brazilian case study", *Technological Forecasting and Social Change*, v. 167, pp. 120727, 2021.

[184] AL SADAWI, A., HASSAN, M. S., NDIAYE, M. "A Survey on the Integration of Blockchain With IoT to Enhance Performance and Eliminate Challenges", *IEEE Access*, v. 9, pp. 54478–54497, 2021.

[185] GARROCHO, C. T. B., KLIPPEL, E., MACHADO, A. V., et al. "Blockchain-based Machine-to-Machine Communication in the Industry 4.0 applied at the Industrial Mining Environment". In: *2020 X Brazilian Symposium on Computing Systems Engineering (SBESC)*, pp. 1–8. IEEE, 2020.

[186] WANG, Y. "A blockchain system with lightweight full node based on dew computing", *Internet of Things*, v. 11, pp. 100184, 2020.

[187] HTTPS://ETHERSCAN.IO. "https://etherscan.io". Disponível em: <`https://etherscan.io`>.

[188] LOSADA, M., CORTÉS, A., IRIZAR, A., et al. "A Flexible Fog Computing Design for Low-Power Consumption and Low Latency Applications", *Electronics*, v. 10, n. 1, 2021. ISSN: 2079-9292. doi: 10.3390/electronics10010057. Disponível em: <`https://www.mdpi.com/2079-9292/10/1/57`>.

[189] PERUZZI, G., POZZEBON, A. "Combining LoRaWAN and NB-IoT for Edge-to-Cloud Low Power Connectivity Leveraging on Fog Computing", *Applied Sciences*, v. 12, n. 3, 2022. ISSN: 2076-3417. doi: 10.3390/app12031497. Disponível em: <https://www.mdpi.com/2076-3417/12/3/1497>.

[190] YANG, H., SHEN, Y., CETIN, M., et al. "Supporting Transportation System Management and Operations Using Internet of Things Technology". May 2021. Disponível em: <https://rosap.ntl.bts.gov/view/dot/56310>. Tech Report.

[191] BHATTACHARJEE, P., ROY, S., BISWAS, S., et al. "Design of an Energy-Efficient Probabilistic Algorithm for a Hybrid Healthcare Network". In: Sikdar, B., Prasad Maity, S., Samanta, J., et al. (Eds.), *Proceedings of the 3rd International Conference on Communication, Devices and Computing*, pp. 499–512, Singapore, 2022. Springer Singapore. ISBN: 978-981-16-9154-6.

[192] KULIK, V., PHAM, V. D., KIRICHEK, R. "Methods and Models for Using Heterogeneous Gateways in the Mesh LPWANs". In: Vishnevskiy, V. M., Samouylov, K. E., Kozyrev, D. V. (Eds.), *Distributed Computer and Communication Networks*, pp. 137–148, Cham, 2020. Springer International Publishing. ISBN: 978-3-030-66471-8.

[193] PEREIRA, V., HADJIELIAS, E., CHRISTOFI, M., et al. "A systematic literature review on the impact of artificial intelligence on workplace outcomes: A multi-process perspective", *Human Resource Management Review*, p. 100857, 2021. ISSN: 1053-4822. doi: https://doi.org/10.1016/j.hrmr.2021.100857. Disponível em: <https://www.sciencedirect.com/science/article/pii/S105348222100036X>.

[194] JAVAID, M., HALEEM, A., PRATAP SINGH, R., et al. "Blockchain technology applications for Industry 4.0: A literature-based review", *Blockchain: Research and Applications*, v. 2, n. 4, pp. 100027, 2021. ISSN: 2096-7209. doi: https://doi.org/10.1016/j.bcra.2021.100027. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2096720921000221>.

[195] ROCHA-JÁCOME, C., CARVAJAL, R. G., CHAVERO, F. M., et al. "Industry 4.0: A Proposal of Paradigm Organization Schemes from a Systematic Literature Review", *Sensors*, v. 22, n. 1, 2022. ISSN: 1424-

8220. doi: 10.3390/s22010066. Disponível em: <https://www.mdpi.com/1424-8220/22/1/66>.

[196] MAMDOUH, M., AWAD, A. I., KHALAF, A. A., et al. "Authentication and Identity Management of IoHT Devices: Achievements, Challenges, and Future Directions", *Computers Security*, v. 111, pp. 102491, 2021. ISSN: 0167-4048. doi: https://doi.org/10.1016/j.cose.2021.102491. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167404821003151>.

[197] ROIG, P. J., ALCARAZ, S., GILLY, K., et al. "Modeling an Edge Computing Arithmetic Framework for IoT Environments", *Sensors*, v. 22, n. 3, 2022. ISSN: 1424-8220. doi: 10.3390/s22031084. Disponível em: <https://www.mdpi.com/1424-8220/22/3/1084>.

[198] SAJID, M., ULLAH, S., JAVAID, N., et al. "Exploiting Machine Learning to Detect Malicious Nodes in Intelligent Sensor-Based Systems Using Blockchain", *Wireless Communications and Mobile Computing*, 12 2021. doi: 10.1155/2022/7386049.

[199] ZHANG, H., ZHANG, X., GUO, Z., et al. "Secure and Efficiently Searchable IoT Communication Data Management Model: Using Blockchain as a new tool", *IEEE Internet of Things Journal*, pp. 1–1, 2021. doi: 10.1109/JIOT.2021.3121482.

[200] ISTIAQUE AHMED, K., TAHIR, M., HADI HABAEBI, M., et al. "Machine Learning for Authentication and Authorization in IoT: Taxonomy, Challenges and Future Research Direction", *Sensors*, v. 21, n. 15, 2021. ISSN: 1424-8220. doi: 10.3390/s21155122. Disponível em: <https://www.mdpi.com/1424-8220/21/15/5122>.

[201] ALI, R. A., ALI, E. S., MOKHTAR, R. A., et al. "Blockchain for IoT-Based Cyber-Physical Systems (CPS): Applications and Challenges". In: De, D., Bhattacharyya, S., Rodrigues, J. J. P. C. (Eds.), *Blockchain based Internet of Things*, pp. 81–111, Singapore, Springer Singapore, 2022. ISBN: 978-981-16-9260-4. doi: 10.1007/978-981-16-9260-4_4. Disponível em: <https://doi.org/10.1007/978-981-16-9260-4_4>.

[202] CHEN, Y., LU, Y., BULYSHEVA, L., et al. "Applications of Blockchain in Industry 4.0: a Review", *Information Systems Frontiers*, Feb 2022. ISSN: 1572-9419. doi: 10.1007/s10796-022-10248-7. Disponível em: <https://doi.org/10.1007/s10796-022-10248-7>.

[203] YU, X., SHI, X. "Smart City Medical Resource Allocation System Based on Big Data". In: Atiquzzaman, M., Yen, N., Xu, Z. (Eds.), *2021 International Conference on Big Data Analytics for Cyber-Physical System in Smart City*, pp. 441–447, Singapore, 2022. Springer Singapore. ISBN: 978-981-16-7466-2.

[204] PATTEWAR, G., MAHAMUNI, N., NIKAM, H., et al. "Management of IoT Devices Security Using Blockchain—A Review". In: Shakya, S., Balas, V. E., Kamolphiwong, S., et al. (Eds.), *Sentimental Analysis and Deep Learning*, pp. 735–743, Singapore, 2022. Springer Singapore. ISBN: 978-981-16-5157-1.

[205] AZROUR, M., MABROUKI, J., GUEZZAZ, A., et al. "Internet of Things Security: Challenges and Key Issues", *Security and Communication Networks*, v. 2021, pp. 5533843, Sep 2021. ISSN: 1939-0114. doi: 10.1155/2021/5533843. Disponível em: <https://doi.org/10.1155/2021/5533843>.

[206] ZHOU, X., KRAFT, M. "Blockchain Technology in the Chemical Industry", *Annual Review of Chemical and Biomolecular Engineering*, v. 13, n. 1, pp. null, 2022. doi: 10.1146/annurev-chembioeng-092120-022935. Disponível em: <https://doi.org/10.1146/annurev-chembioeng-092120-022935>. PMID: 35363506.

[207] JING, S., ZHENG, X., CHEN, Z. "Review and Investigation of Merkle Tree's Technical Principles and Related Application Fields". In: *2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA)*, pp. 86–90, 2021. doi: 10.1109/CAIBDA53561.2021.00026.

[208] DE CAMARGO, E. T., SPANHOL, F. A., CASTRO E SOUZA, Á. R. "Deployment of a LoRaWAN network and evaluation of tracking devices in the context of smart cities", *Journal of Internet Services and Applications*, v. 12, n. 1, pp. 8, Oct 2021. ISSN: 1869-0238. doi: 10.1186/s13174-021-00138-7. Disponível em: <https://doi.org/10.1186/s13174-021-00138-7>.

[209] FOUNDATION, H. "Hyperledger Fabric". Last Visited in 04/04/2022. Disponível em: <https://www.hyperledger.org/use/fabric>.

[210] POLKADOT. "Polkadot Network". Last Visited in 04/04/2022. Disponível em: <https://polkadot.network/parachains/>.