Marc Gröling

E-Waste Tracking System with Public Permissioned Blockchain using Byzantine Fault Tolerance Consensus in Norway

Master's Thesis in Computer Science
Supervisor: Prof. Ibrahim A. Hameed, PhD
Co-supervisor: Prof. Deepti Mishra, PhD
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Engineering

NTNU

ABSTRACT

Electric and electronic equipment is the fastest-growing waste stream. These items contain many precious materials that can be recovered if they are properly recycled. Apart from recovering valuable finite materials, recycling also has economic and ecological benefits. However, the lack of reliable data and statistics on waste streams makes this process harder. Furthermore, the public does not fully trust the recycling process, and according to a survey done in this work, almost half (45.8%) the public is afraid that their personal data will be abused. This increase in e-waste items combined with the mediocre recycling of it, creates many problems. In this work, a decentralised system is proposed that enables the efficient tracking of electronic devices, or waste items in general. A barcode is attached to an item and can be scanned at recycling points, adding the locations and timestamps to where an item was scanned. This is aimed to gain the public's trust and provide data for the industry. Due to the requirements of the system, the data is stored on a blockchain. Multiple implementations of blockchain are examined and finally, Tendermint, a blockchain consensus framework that uses a Byzantine-fault tolerant algorithm, is chosen. The blockchain system is evaluated on a four-node test network that has a transaction throughput of about 130 transactions per second which is sufficient for the use case. Furthermore, a 25 GB hard drive may store up to 1.1 billion transactions, which while making assumptions would last for about four years. The system is practical and while there are still flaws, it can be considered as a first step towards practical e-waste tracking.

PREFACE

I would like to thank everyone who has helped me to get to this point and to complete this thesis. Thank you to both of my supervisors, for their valuable feedback and help. Many thanks to the other faculty members as well, who supported me throughout this thesis. I also want to thank my family, my mother, my father, and my sister for always being there for me and supporting me emotionally. Lastly, I would also like to express my gratitude to my friends, who helped me stay motivated and kept my spirits up high. Danke.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

List of all abbreviations in alphabetic order:

- **ABCI** Application Blockchain Interface
- **BFT** Byzantine Fault Tolerant
- **ECDSA** Elliptic Curve Digital Signature Algorithm
- **EEE** Electric and Electronic Equipment
- **NTNU** Norwegian University of Science and Technology
- **PBFT** Practical Byzantine Fault Tolerance
- **PoW** Proof of Work
- **Tx** Transaction
- **WEEE** Waste Electric and Electronic Equipment

CHAPTER 1

INTRODUCTION

In recent the years the demand for Electric and Electronic Equipment (EEE) has risen. This in combination with short lifespans of EEE products, due to technological advancements, has led to an increase of Waste Electronic and Electric Equipment (WEEE) [1]. It is the fastest growing waste stream and in 2019 approximately 53.6 million tons of e-waste were generated globally, of which only less than 13% was recycled [2]. The rest of it was discarded (much of it to landfills), which creates many environmental and health risks [2]. "Proper handling of e-waste and efficient recycling can enable the recovery of valuable raw materials, which not only reduces the demand for finite primary resources but also brings significant economic and environmental benefits." [3]

"One of the most critical gaps associated with e-waste management is the lack of reliable data and statistics on the quantities of e-waste being generated. Measuring and monitoring e-waste quantities on the national and international level is essential in addressing the e-waste challenge" [3]. Furthermore, tracking WEEE and publishing this data may increase the public's trust in the recycling chain and increase the willingness to recycle.

In this thesis, a survey was carried out to get a better understanding of the public's relation with WEEE. Furthermore, a tracking system that uses a blockchain was proposed. The research goals of this thesis can be summarised as:

- Analyse the public's relation with WEEE and identify: if and how much the public recycles their devices, what are the reasons for not recycling, and if a tracking system like the one proposed would increase the public's willingness to recycle.
- Develop a system that is tailored to the use case of e-waste tracking in Norway and analyse its suitability.

In Chapter 2 related work is examined and it is explained how this thesis' work differs from them. Later on in Chapter 3, an overview of blockchain technology is given, and design decisions (choice of consensus algorithm and digital signature algorithm) are presented and reasoned for. Following, Chapter 4 presents the survey that was conducted in this thesis and analyses its results. Next, Chapter 5 gives an overview of the tracking system that was developed in a previous project, explains the use case, extensions of the previous project, and why blockchain is suited for this use case. In the following chapter, Chapter 6 reviews current implementations of blockchain and gives reasons for their acceptance/rejection. Afterwards, Chapter 7 gives a detailed explanation of the implementation of the system that was developed in this thesis. The chapter is followed by Chapter 8, which provides a performance and security analysis of the developed system. Finally, Chapter 9 provides a summary of the completed work, as well as providing possible improvements for it.

CHAPTER 2

RELATED WORK

There have already been several works on tracking waste, with some using blockchain technology and some focused on e-waste tracking specifically:

In [4] the authors created a blockchain-based IoT-enabled system for monitoring EEE post-production. The system works on the Ethereum blockchain and supports smart contracts to be more autonomous. The system also employs a reputation system to minimise fraud. Their system is designed to work in smart cities and expects a level of infrastructure that is not feasible for implementation in the whole of Norway, which is the objective of this thesis.

In [5] a system similar to the one in [4] is developed. [5] lacks the destruction certificates of the other work though. Like [4], it is not feasible to implement this kind of system for all of Norway.

In [6] a blockchain-based system to automate forward supply chain processes regarding COVID-19 medical equipment is developed. They also use smart contracts on the Ethereum blockchain. Their work does not consider WEEE though.

The authors of [7] created a blockchain-based system that enables the remanufacturing and refurbishment of EEE products. Their work also deals with privacy-related issues that stem from old user data on these devices. They employ a private blockchain which does not address the public's mistrust in the recycling process.

In [8], hidden GPS location trackers were attached to various types of EEE products. They discovered that about one-third of their devices were sent overseas, which were likely unreported in official trade data. Employing GPS tracking like theirs is not feasible for a use case of a grander scale like this one, however.

The authors of [9] propose an incentive-based blockchain system that keeps track of the lifecycle of EEE products from manufacturing to recycling. They also propose a public-private partnership between the government and private players to make the process more organised. Like [7], they employ a private blockchain solution, which does not address the public's mistrust in the recycling process.

CHAPTER 3

BLOCKCHAIN TECHNOLOGY

This chapter gives an overview of blockchain technology and explains the basic concepts of it. Furthermore, two consensus algorithms (Proof of Work (PoW) and Byzantine Fault Tolerance (BFT) consensus) that were deemed suitable for this application are presented. It is concluded that BFT is the preferred algorithm for this use case. Next, explanations for the rejection of other consensus algorithms are given. Finally, digital signature algorithms are investigated and the Elliptic Curve Digital Signature Algorithm (ECDSA) [10] is deemed most suitable.

Blockchain is a decentralised, immutable database that is spread over many participants, which are typically referred to as nodes. The chain consists of an ordered number of blocks that each hold information. "Each Block $B_{i>0}$ is immutably connected to a single preceding block B_{i-1} through a cryptographic hash function $H(B_{i-1})$." [11] As such, if one wants to change the information of block B_i, they would need to update the hash values of all following blocks for the chain to be valid. This property makes the blockchain secure and assures immutability. To keep the chain consistent across nodes, they have to achieve consensus, which is one of the major challenges in a blockchain and discussed in detail in Section 3.1.

The most common use case of blockchain is to keep track of a virtual currency, with the most popular being Bitcoin [12]. There, each block stores several transactions that hold information about the sender and receiver of the transaction, as well as the amount of currency that is transferred. However, the data that blocks store can be arbitrary, thus enabling various use cases. Blockchain has been used in many areas, such as inventory, manufacturing, supply chain, IoT, finance, governance, and others [13].

Although blockchain has many benefits, there are also some drawbacks of it that one must consider. The first one is scalability, with an increasing number of nodes, more bandwidth, storage space, and power is required. Furthermore, the handling of transactions can be slow, depending on the consensus algorithm used. [11] The second drawback is that blockchains tend to use a lot of power [11], Bitcoin uses (as of February of 2023) 81.66 TWh annually, which is comparable to the power consumption of Chile.[1]

3.1 Consensus Algorithm

To keep the chain consistent across nodes and add new blocks, the nodes have to reach consensus. For this use case, two consensus algorithms were deemed especially suitable: Proof of Work and Byzantine Fault Tolerance consensus. First, the basic protocol of each of these is described and then they are compared to each other for this thesis' use

[1] `https://digiconomist.net/bitcoin-energy-consumption`

case. Finally, other consensus algorithms are presented and reasons for their rejection are given.

3.1.1 Proof of Work (PoW) Consensus

In PoW, nodes have to prove that a certain amount of work went into creating new blocks. This is accomplished by requiring the binary representation of the hash of a block to lead with a certain number of zeros (by changing an arbitrary field, which is referred to as *nonce*, as well as updating the timestamp accordingly). How many leading zeros are required is decided by the network and is described as network difficulty. The longest chain of blocks and thus the one where the highest amount of work went into is considered the valid one [12].

3.1.2 Byzantine Fault Tolerance (BFT) Consensus

BFT consensus protocols deal with Byzantine Faults, i.e. the Byzantine generals problem. The problem abstractly describes the situation where a computer system must deal with malfunctioning components that give conflicting information to different parts of the system. A more thorough description of the problem can be found in [14].

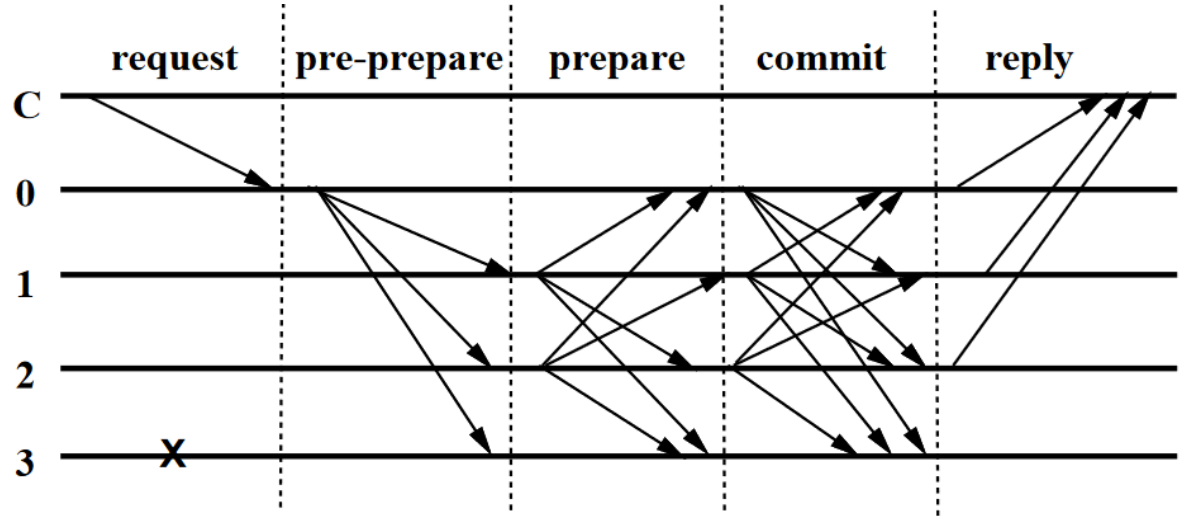


Figure 1: Normal Case Operation of PBFT [15], node 3 is Byzantine (faulty)

Practical Byzantine fault tolerance (PBFT) [15] was the first algorithm that could deal with Byzantine faults in an asynchronous environment and is practical in terms of performance. The algorithm can deal with f Byzantine (faulty) nodes with a total number of $3f + 1$ nodes. The normal case operation flow of the algorithm can be seen in Figure 1. First, the client sends a request to the leader (that is determined by the current view (which can change)), and then the leader broadcasts a *pre-prepare* message. When a secondary node receives the *pre-prepare* message, it checks its validity. If it considers the *pre-prepare* message valid, then it broadcasts a *prepare* message and waits until it receives $2f$ *prepare* messages from different secondary nodes. This phase ensures that non-faulty nodes agree on a total order for the requests within a view. Once it has received $2f$ *prepare* messages, it broadcasts a *commit* message. It then waits until it receives $2f$ valid *commit* messages from other nodes, after which it executes the request locally and

sends a reply to the client. Once the client has received at least $f + 1$ matching replies, it can view the replies as valid and can consider the request executed. A more detailed description of the algorithm can be found in [15].

3.1.3 Comparison of PoW and BFT Consensus

In Table 1, a high-level comparison between PoW and BFT consensus is given, which was made in [16]. For explanations of blockchain properties, please refer to their article.

Table 1: High-level comparison between PoW and BFT blockchain consensus families for a set of important blockchain properties. Entries in bold suggest desirable features and highlight the advantages of one consensus family over the other. [16]

	PoW consensus	BFT consensus
Node identity management	**open, entirely decentralized**	permissioned, nodes need to know IDs of all other nodes
Consensus Finality	no	**yes**
Scalability (no. of nodes)	**excellent (thousands of nodes)**	limited, not well explored (tested only up to $n \leq 20$ nodes)
Scalability (no. of clients)	**excellent (thousands of clients)**	**excellent (thousands of clients)**
Performance (throughput)	limited (due to possible of chain forks)	**excellent (tens of thousands tx/sec)**
Performance (latency)	high latency (due to multi-block confirmations	**excellent (matches network latency)**
Power consumption	very poor (PoW wastes energy)	**good**
Tolerated power of an adversary	$\leq 25\%$ computing power	$\leq 33\%$ voting power
Network synchrony assumptions	physical clock timestamps (e.g., for block validity)	**none for consensus safety** (synchrony needed for liveness)
Correctness proofs	no	**yes**

According to Table 1, PoW only outperforms BFT consensus in two blockchain properties: node identity management and scalability in terms of the number of nodes.

In Chapter 5, it was already concluded that the blockchain should be permissioned. As such the requirement of BFT that all nodes need to be known is of no concern for the choice of consensus algorithm and thus this property is irrelevant.

However, the scalability in terms of the number of nodes is of way higher concern for this use case. As one node is required for each waste collection site and the system might be deployed for all of Norway, a requirement for hundreds or even thousands of nodes should be expected. Since [16] was released, there have been new proposals, such as [17], which improve scalability through sharding. As such, BFT is the preferred option for the consensus algorithm in this thesis.

3.1.4 Other Consensus Algorithms

Other Consensus Algorithms that were investigated, but not found suitable are listed below. A more detailed explanation and comparison of these algorithms is given in [18].

- **Proof of Stake, Proof of Importance:** Requires a virtual currency for use.
- **Ripple Protocol Consensus Algorithm [19]:** Only tolerates up to 20% faulty nodes, by improving latency. As the expected number of transactions per second is probably in the 1-digit or 2-digit area, this decrease in security is not weighed out by the improvement in latency, however.
- **Stellar Consensus Protocol [20]:** Security analysis of this protocol is difficult, which makes it hard to determine how many faulty nodes the protocol can tolerate. As such, a consensus algorithm that can deliver guarantees is preferred.
- **Proof of Authority:** As discussed in [21], BFT protocols are preferred when data integrity is a priority, which is the case for this use case.

3.2 Digital Signature Algorithm

An important component of blockchains is digital signatures, which are used for authentication and verification of the integrity of a message. The general scheme of digital signatures can be seen in Figure 2. The originator generates a digest of the message using a cryptographic hash function, then this digest is encrypted using the private key p_{priv} (which corresponds to the public key p_{pub}). The message and the signature are then sent to the recipient, where the encrypted digest is decrypted using the public key p_{pub}, resulting in the expected digest of the message. The recipient then also generates the digest of the message using the same cryptographic hash function that the originator used and compares the two digests. If they match, the recipient can assume that the message was sent from a person that holds the private key p_{priv} and that the content of the message was not altered by a third party.

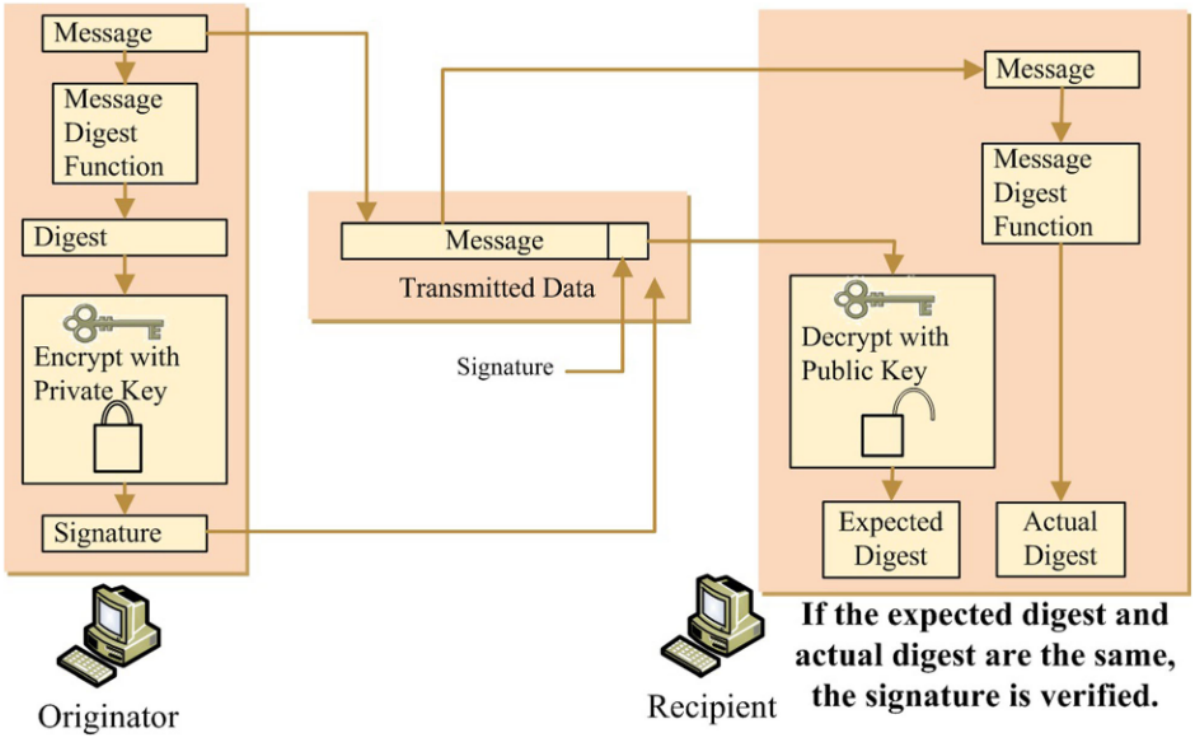


Figure 2: Digital Signature Scheme [22]

[22] reviews state-of-the-art digital signature algorithms. According to them, the Elliptic Curve Digital Signature Algorithm (ECDSA) [10] outperforms other digital signature algorithms (RSA [23] and DSA [24]), in terms of security, performance, and storage requirements. As such, ECDSA is the preferred option for the digital signature algorithm for this thesis.

CHAPTER 4

SURVEY

To get a better understanding of the public's relation with WEEE in Norway, a survey was conducted. This section shows the results of it, explains the survey's shortcomings, and discusses its results. The complete data of the survey is provided[1], as well as the form for it[2]. The survey was distributed through the e-mail network of the Norwegian University of Science and Technology (NTNU) and was answered by 114 respondents. This limits the survey's generality, however, made its distribution easier.

4.1 Results

In Figure 3a, one can see the age of respondents. The age groups are mostly evenly represented (with a maximum difference of 9.7%), except for the groups "0-18" and "60+".

Figure 3b shows how many old unused devices respondents keep at home. Only 20.2% keep one or zero devices at home, the rest of respondents keep two or more devices at home.

In Figure 4, potential reasons are listed why respondents do not send their items for recycling. As respondents might have multiple reasons, they were asked to select all that apply. Over the majority of the respondents (61.5%) lists that they think that they might need the device someday. The next highest reason (45.8%) for not submitting their devices is that people think that their data on these devices may be abused.

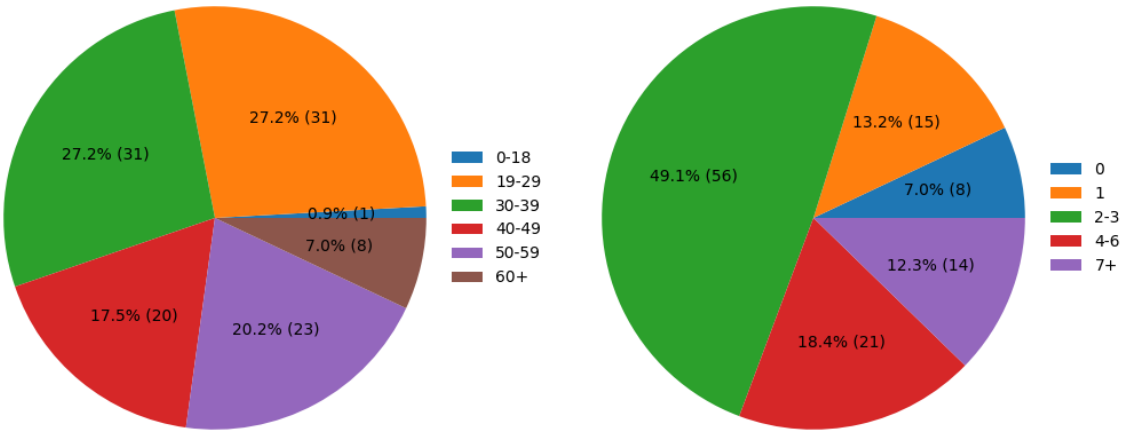
Figure 5a shows whether respondents have ever sent an item for recycling. The majority of respondents (55.4%) have never sent an item for recycling.

The system proposed in this thesis is supposed to help people build trust in the recycling process and whether it might be able to achieve that, can be seen in Figure 5b. 55.4% answered that a system like the one proposed in this thesis would encourage them to send their items to recycling.

Finally, Figure 6 displays reasons for not submitting devices of different age groups. The values refer to what percentage of that age group has given the reason as one of theirs. It seems that younger people are less afraid of data abuse (only 24.1% compared to the average of 45.8%). It is unclear why this is the case. Younger groups ("19-29" and "30-39") are about twice as likely to be unaware of where to submit their devices, compared to older age groups). These younger age groups are also more likely to think that they might need their device someday (about 70% think that way). Also, the age group "19-29" is the only one, where a significant amount of respondents (24.1%) are unaware that these devices are recyclable at all. All other age groups are below 1% in that category.

[1] `https://github.com/marc131183/e-waste-blockchain/blob/master/survey.csv`

[2] `https://forms.gle/PQ5aKS4AFYT2pj5X7`



(a) Ages of respondents

(b) Number of old unused devices respondents keep at home

Figure 3: Survey results

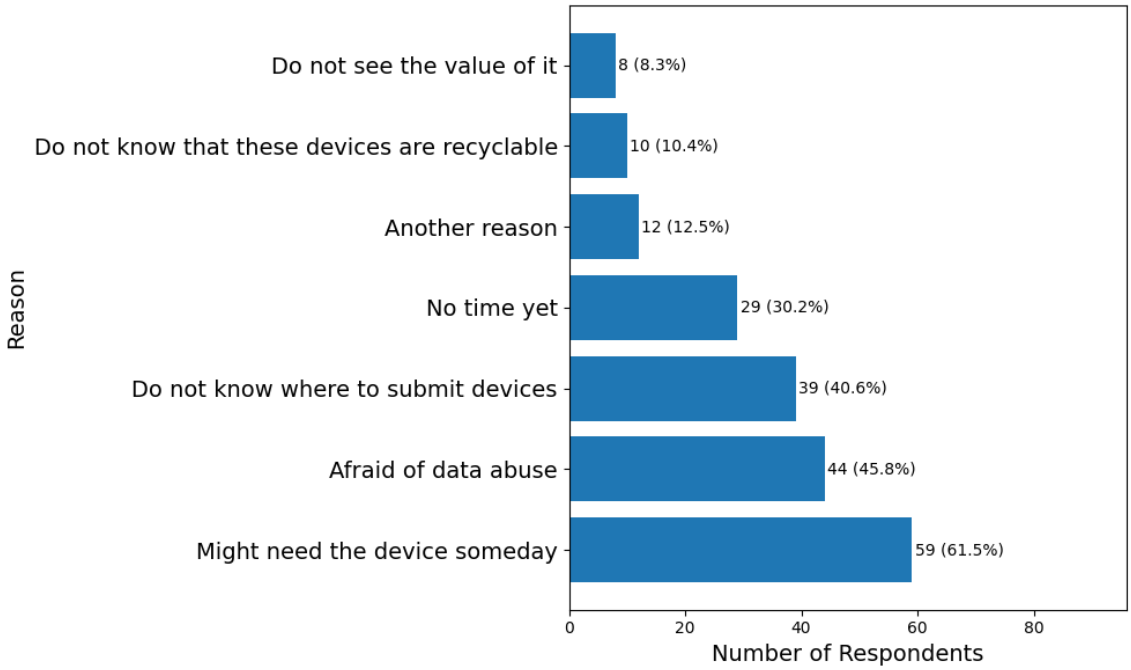
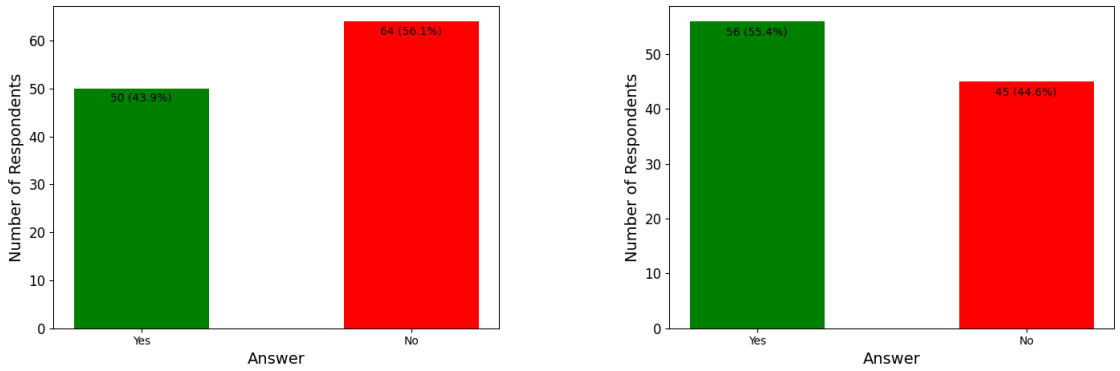


Figure 4: Reasons why respondents do not send items for recycling



(a) Results if respondents have ever sent a device for recycling

(b) Results if respondents would be encouraged to recycle more if there was a tracking system such as the one proposed in this thesis

Figure 5: Survey results

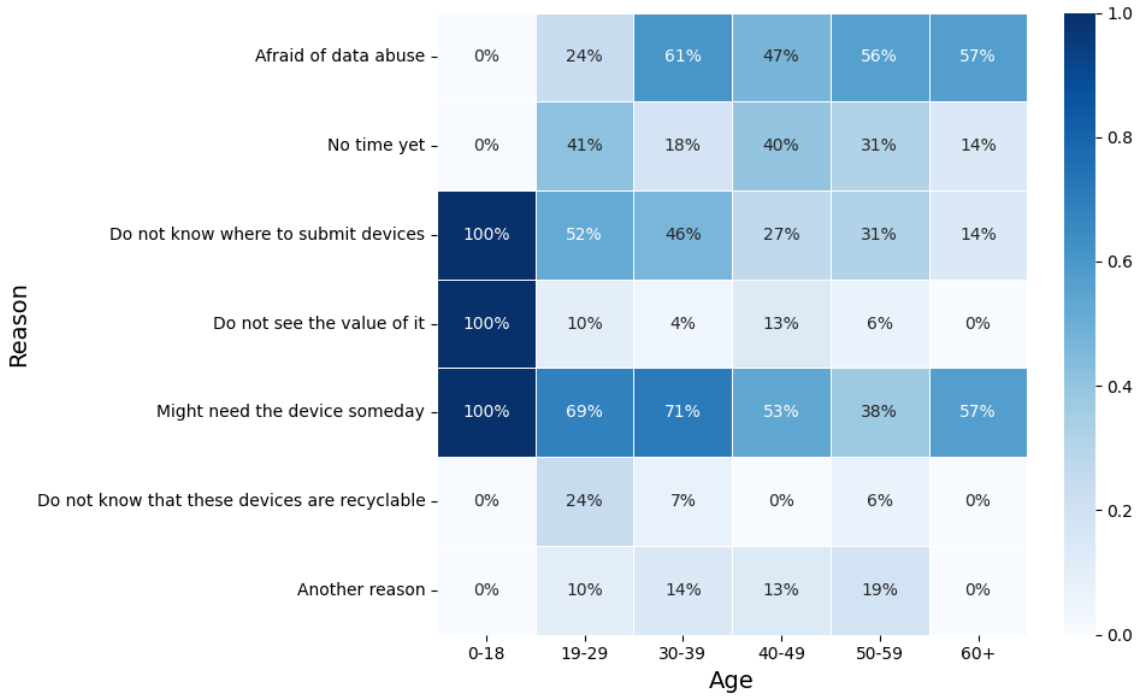


Figure 6: Reasons for not submitting devices of different age groups

4.2 Additional Remarks from Respondents

In the survey form, there was also a text field, where respondents could give additional information. The results of this are summarised in this section.

One respondent remarked, that it would be helpful to see the companies clear the hard drive when delivering the device, or at least explain the process of it. This could easily be added to the proposed system of this thesis, by adding an info section for each registered recycling location (or at least per company).

Another respondent believes that their e-waste will not be properly recycled and

instead shipped to the global south and burned down there. The system proposed in this thesis allows customers to see where their device ends up and thus should solve this issue.

Other respondents have remarked, that they feel attached to their old devices, as they hold sentimental value to them. Perhaps, clarifying the importance of recycling would encourage people who feel sentimental value for their old devices to still submit them.

Finally, another respondent said that they are afraid that their old device contains some useful data that they might need someday. For them, an easy-to-use routine to identify and transfer all (potentially) useful data on a device would be helpful.

4.3 Limitations of the Survey

While the survey conducted in this thesis can give an idea of the public's relation with WEEE, it is important to consider its limitations. First, the age groups "0-18" and "60+" are underrepresented. Second, the survey was distributed throughout the mail network of NTNU and as such, many respondents work at NTNU. As such, most respondents belong to the "middle class" and other socioeconomic groups are underrepresented. Third, most respondents likely live in one of three locations: Ålesund, Trondheim, or Gjøvik (the cities of the university's campuses). Although this makes the survey biased, it can still be considered an important sample.

4.4 Discussion

As seen in Figure 3b, the majority of respondents keep two or more old unused devices at home. As a result, there are plenty of devices that could be recycled, which would lead to economic and ecological benefits, as discussed in Chapter 1.

In Figure 4, the reasons for not recycling were listed. While the majority of reasons are captured, 12.5% responded with "Another reason". While this only represents about an eighth of the population, in the future, it should still be explored what these reasons are.

Furthermore, according to Figure 5a, the majority of respondents have never sent an item for recycling. As such, it would be beneficial if a higher number of the population would recycle their items and since the majority of people does not do it yet, incentives for more recycling should be given.

Figure 5b shows whether or not a system like the one proposed in this thesis would encourage people to recycle more. About half of the respondents (55.4%) responded with "yes". This represents a significant amount of the population. As such, implementing the system proposed in this thesis on a grand scale would help to combat the waste problem. However, it should be noted that other ways to encourage recycling should still be explored.

Lastly, as seen in Figure 6, some reasons are more prominent in some age groups. For example, younger people (19-39) are a lot more likely to be unaware of where to recycle items, compared to older people. As such, it would probably be beneficial to inform these age groups of where and how to recycle their devices. The reasons "Do not know where to submit devices", "Do not see the value of it", and "Do not know that these devices are recyclable" could be weighed down by launching campaigns to inform the public. To make these particularly effective, the most prominent reasons for age groups should be taken into consideration.

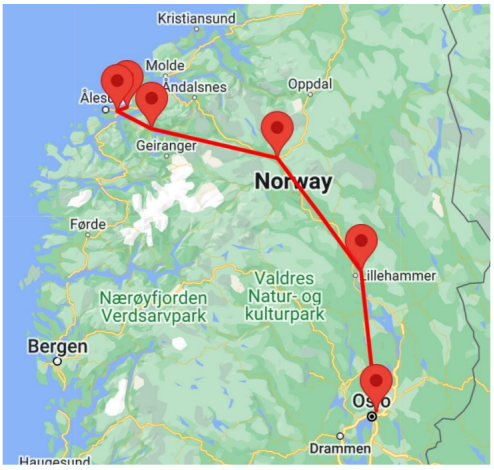
CHAPTER 5

TRACKING SYSTEM

This chapter gives an overview of the tracking system that is used with the blockchain. First, the previous project is briefly explained, and then a description of the use case is provided, which is followed by extensions that were added to the previous project. Finally, the validity and usefulness of using blockchain technology for this use case are explained.

In a previous project [25], a tracking system for WEEE products was developed. This system enables users to submit their devices and then see where the device currently is and where it has been. In this system, a unique QR code is generated for each device and attached to it, see Figure 7a. This QR code can then be scanned, which forwards one to a website. There, a checkpoint is added via the (scanning) device's current location. This checkpoint is then added for this device and saved in the cloud. The trajectory of the device (the checkpoints) can then be seen on the website, which can be seen in Figure 7b.

(a) Unique QR code is attached to a device [25]



(b) Example trajectory of a device [25]

Figure 7: E-Waste Tracking System

5.1 Use Case Description

The use case consists of two actors: recycling companies and customers (public citizens that recycle their devices). Customers submit items that should be recycled to recycling companies. Customers, therefore, submit their items to waste sites in their vicinity. Then recycling companies take care of the whole recycling process. Multiple companies may

be involved in this process and items may be sent to multiple recycling sites, which may be owned by different companies.

An example of how the proposed system could be incorporated into the use case could be: a customer sends their EEE device d for recycling and therefore submits it to recycling station r_1, which is owned by company c_1. There, an employee that works at recycling station r_1 orders the allocation of a new device id, prints the barcode out for it and attaches it to device d. This barcode is then also scanned and r_1 is added as the first location for device d. The customer receives this assigned id for the device, which they can later use to track their device. On recycling station r_1, the item is first stored and once enough items of the same category have been received there, it is forwarded to recycling station r_2, which is owned by company c_1. On recycling station r_2, the barcode of device d is then scanned and location r_2 is added. Also, the items are then sorted more thoroughly on recycling station r_2 and the device d is sent to recycling station r_3, which is owned by company c_2. There, the barcode of device d is then scanned again and as the item shall be destroyed there, a destruction certificate is added. Device d is then destroyed (recycled). During this process and after, the customer may at any point view the current location and status of their device d. They can access it through the device id that has been assigned to device d and given to them in the beginning, when they submitted their item. They may also view where their device has been so far (and at what times it was received there) and which companies have taken care of it. Furthermore, anyone may view the device trajectories for any device (as all are publicly available. As a result, companies can get an understanding of the waste flow. This whole process can also be seen in Figure 8.

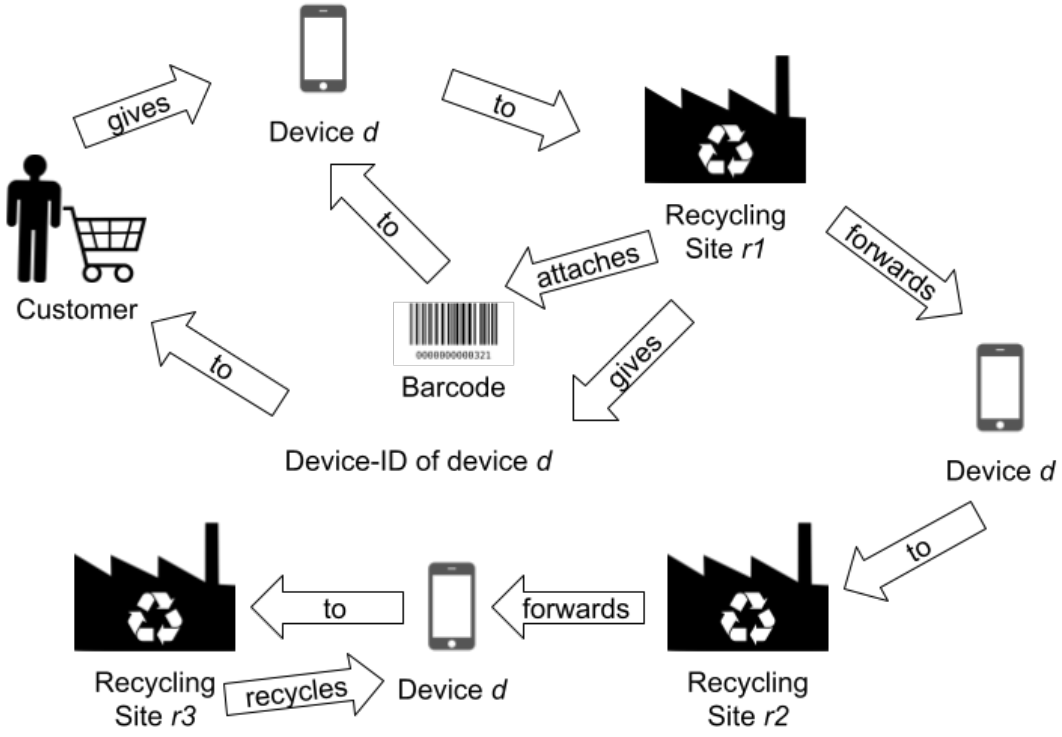


Figure 8: Example Recycle Chain with the Proposed System (Image Sources: Customer, Smartphone, and Recycling Site)

An important factor for the selection of a software solution for this use case is the number of recycling sites and the number of devices that flow through these in a time unit. Unfortunately, it is not clear how many waste sites exist in Norway and how many devices flow through these in a given time unit. As a result, further research needs to be done on this matter.

5.2 Non-Repudiation of Locations

In the original system, the location where a device was scanned was determined by accessing the (scanning) device's location. However, since a malicious user might alter this location, it would be beneficial to employ a scheme which aims to prevent this, or at least make it more difficult. To achieve this, digital signature algorithms are used, which are briefly explained in Section 3.2. When a new location is added as a verified waste management site, it is also assigned a public/private key pair. The private key is saved on the site's node (the computer that manages the site's connection) and the public key is broadcasted to all other nodes with the location. This key is then directly linked to the location of the site and saved as such in all other nodes. When a node now broadcasts that it has received a new device at its location, it will sign this message with its private key. Other nodes can now check the validity of the specified location by verifying the message with the corresponding public key.

5.3 Destruction Certificate for Devices

In addition to adding location-based digital signatures, destruction certificates were added. These location-coupled signatures act as certificates that a device was destroyed and taken out of the supply chain. A similar approach was employed in [4]. Destruction certificates ensure that only special sites can destroy items which increases the traceability of devices and makes it easier to understand whether a device was illegally taken out of the supply chain or was legally destroyed.

5.4 Use of Blockchain Technology

To make the current system more transparent, secure, and less trust-dependent, it was proposed to use a blockchain as a storage solution. To validate that it makes sense to use a blockchain, the guideline in [26] was followed. Their article also states which type of blockchain is required. After their scheme, a public permissioned blockchain is appropriate for this use case. The requirements for it are the following:

- *Need for storing a state:* The trajectories of each WEEE product should be stored.
- *Dealing with multiple writers:* There are multiple waste processing sites, which should all be able to write data, simultaneously.
- *Always-online trusted third party cannot be used:* Since there are multiple waste processing companies in Norway and they do not necessarily trust each other, no trusted third party can be appointed.

- *All writers are known:* Since a node needs to have a signature that corresponds to its location, to write data, they are known.
- *Not all writers are trusted:* Malicious users might try to delete data of other waste processing companies, for them to seem untrustful and thus sabotage a competitor.
- *Public verifiability is required:* Trajectories of devices should be known to the public to help build trust and also to make knowledgeable policy decisions for the government and waste industries.

Furthermore, [27] discusses the suitability of blockchain for waste management. They concluded that first, it would have to be ensured that data is entered correctly (since once it has been added, it cannot be changed). Since the use case in this thesis limits itself to scanning device ids, this is ensured, except for adding destruction certificates (which are manually added). Secondly, they concluded that blockchain ensures the prevention of loss of data, which combined with the reliability (due to immutability) is one of the main motives to employ blockchain for this use case.

CHAPTER 6

AVAILABLE IMPLEMENTATIONS

For public permissioned blockchains, there already exist many implementations. As discussed in Section 3.1.3, BFT consensus is the preferred algorithm for this use case and as such only implementations that use it are considered. This section describes the available implementations and provides the reason for choosing/rejecting them. Table 2 summarises implementations that are available at the time of evaluation (February of 2023) and provides reasons for their rejection. It is concluded that Tendermint[1] is the most suitable.

Table 2: Available Implementations and Reason for their Rejection

Name	Reason for their Rejection
Tendermint	accepted
NEO	complexity exceeds requirements
Stellar	tailored for digital currencies
EOS	complexity exceeds requirements
Omniledger	complexity exceeds requirements

6.1 Tendermint

Tendermint is a BFT engine for securely and consistently replicating state machine applications. It communicates with the application via a web socket and thus allows the developer to choose any programming language. For this, the developer must implement the Application Blockchain Interface (ABCI), which communicates with the Tendermint Engine.

The Tendermint consensus algorithm works similarly to the operation flow of PBFT [15], which it is based upon. The operation flow can be seen in Figure 9. Again, as in the PBFT protocol, it is assumed that there are $3f + 1$ nodes. The consensus algorithm works in a round fashion. First, the current leader proposes a block, then each validator (a registered node in the network) validates the block and if valid, broadcasts its prevote. If a node receives at least $2f + 1$ prevotes for a block, then it locks onto this block and broadcasts a precommit message for it. Once a node receives more than $2f + 1$ precommits for a block (and the node has received the block), it commits it locally. To summarise, the first vote (i.e. the prevote) is necessary to ensure that the network agrees that a block is valid and the second vote (i.e. the precommit) is necessary to ensure that the network commits the same block (or none) for a given round. Timeouts are

[1] `https://tendermint.com/core/`

implemented to ensure liveness, which makes Tendermint a weakly synchronous protocol. A more detailed description can be found in [28].

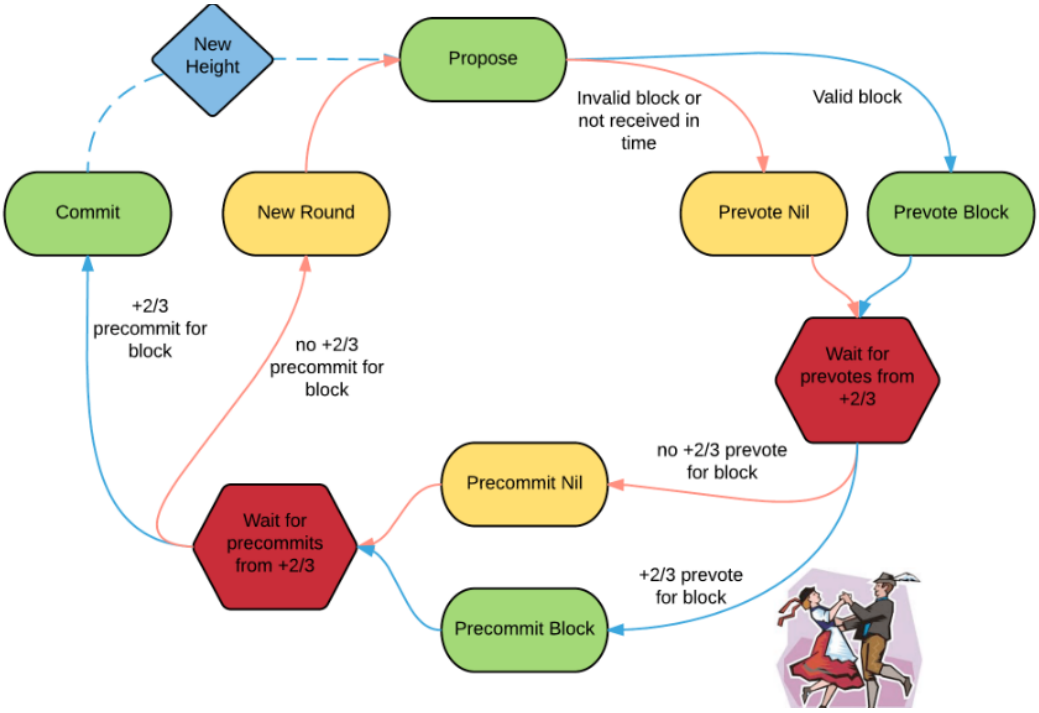


Figure 9: Consensus Protocol of Tendermint (image source)

An important factor in choosing a blockchain solution is its scalability. Unfortunately, it is uncertain how many nodes will be required for the proposed system and how many transactions need to be handled per second. Tendermint has done its own quality assurance with a test network of 200 nodes. This could handle 4449 transactions per minute or 74.15 per second. For more information on their quality assurance, please refer to their documentation[2].

6.2 NEO, EOS, and Omniledger

NEO[3], EOS[4], and Omniledger[17] all use smart contracts to allow for flexibility. [29] found that smart contracts, even ones of modest complexity, are relatively expensive. Furthermore, the amount of flexibility that smart contracts offer is simply not required for this use case, as transactions are relatively simple.

6.3 Stellar

Stellar[5] is a network optimised for payments and assets insurance. It is tailored for the use of cryptocurrencies or other forms of assets[6]. As such, it is not suitable for the use case of this thesis, which is a supply chain use case.

[2] `https://docs.tendermint.com/v0.34/qa/v034/`
[3] `https://neo.org/`
[4] `https://eos.io/`
[5] `https://stellar.org/`
[6] `https://developers.stellar.org/docs/tutorials/send-and-receive-payments`

CHAPTER 7

IMPLEMENTATION

As discussed in Chapter 6, Tendermint[1] is used as the blockchain engine. This chapter provides information on how the blockchain system was set up and implemented. First, an overview of the system is given, which is followed by a more detailed description of each component: the nodes, the Application Blockchain Interface (ABCI), the web server, and the smartphone app.

7.1 Overview

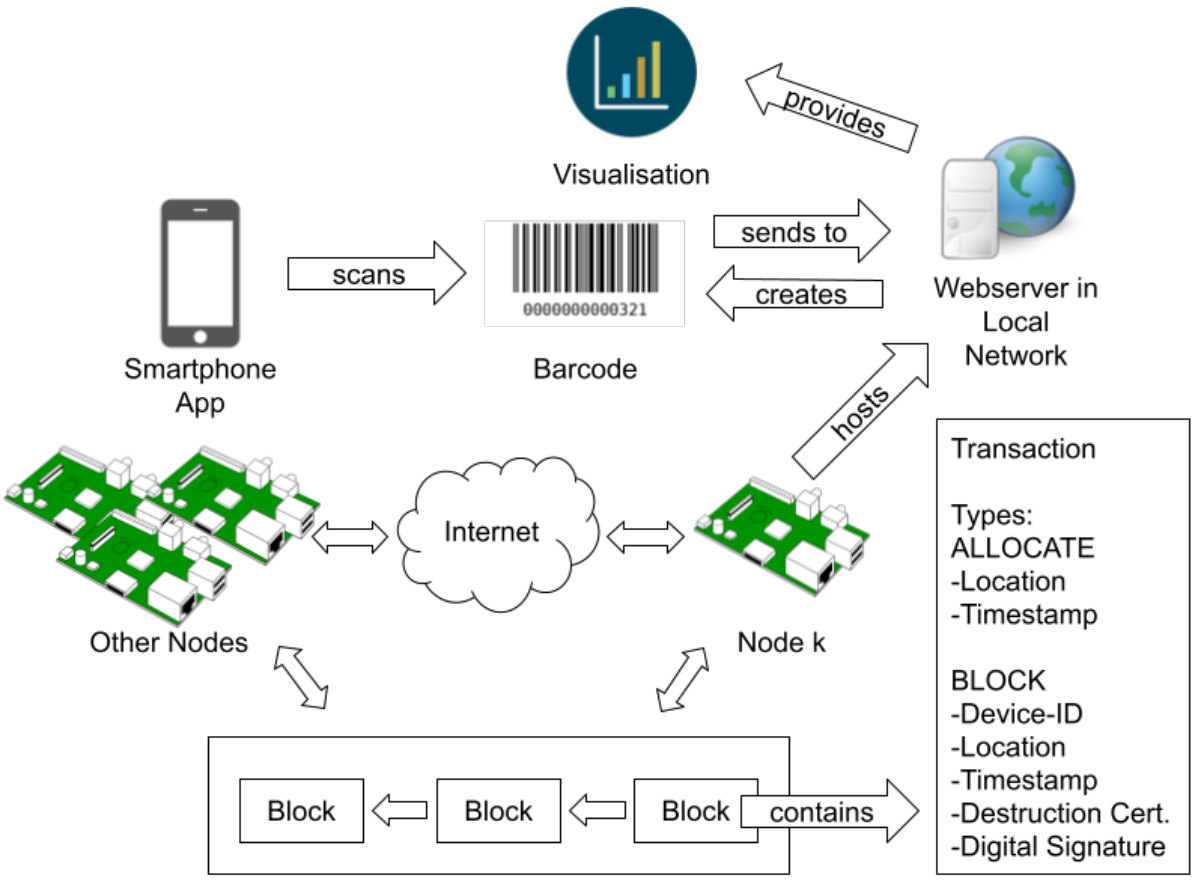


Figure 10: Overview of the Developed System (Image Sources: Smartphone, Node, Graph, and web server)

[1] https://tendermint.com/core/

An overview of the system can be seen in Figure 10. The blockchain network consists of n nodes, each having a local copy of the chain on its hard drive. Through communication over the internet, consensus is reached on how this chain should look like. There are two types of transactions: *ALLOCATE*, which saves that a new device id is allocated and *BLOCK*, which saves that a device has been scanned at a location. In addition, each blockchain node hosts a web server in the local network. On the website, one can allocate a new device id, creating a barcode for that device id, which may then be printed out and attached to the device. This barcode can then be scanned with the developed smartphone app, which upon successful scanning sends a message to the web server and a *BLOCK* transaction is created in the blockchain and broadcasted to the other nodes. Finally, the web server also provides visualisation, where one can see the trajectory of any device (by entering its assigned id).

7.2 Saved Information of Devices

One important decision is to select which attributes of devices shall be stored in the solution. Storing unnecessary attributes leads to a higher number of required disk space and may push the solution to be unpractical. The only attribute of devices that is stored right now is its device id, the locations (+ timestamps) it has been at and if and where it has been destroyed yet. Perhaps, it would be beneficial to store more attributes, however, such as the model of the device, the colour, etc. If attributes like this would be stored as well, this may prevent someone from for example removing the barcode of a device and attaching it to another lower-value device. The blockchain system may be expanded to store more information about devices. This could for example be done by creating a new transaction type that would be called after allocating a device id and then be used to submit it. This could then be stored in a simple table. This method is very likely preferred over including it in the *BLOCK* transaction type, as this would introduce unnecessary redundancy.

7.3 Nodes

As blockchain nodes, the Raspberry Pi 4 Model B was chosen. This device was chosen, as it was available to the author and in general is available at low retail prices (although, these prices have risen since 2021[2]). The proposed blockchain solution is not dependent on the Raspberry Pi architecture and as such, alternatives may be considered. Alternatives should support constant time uint64 multiplication to prevent leakage of private keys. Furthermore, they should fulfil the following requirements (as taken from the Tendermint documentation[3]):

Table 3: Hardware Recommendations for Blockchain Nodes

Component	Minimum	Recommended
RAM size	1 GB	2 GB
Storage space	25 GB	100 GB
CPU	1.4 GHz 64-bit	2.0 GHz 64-bit

[2] `https://picockpit.com/raspberry-pi/why-are-raspberry-pi-prices-so-high-will-it-improve/`
[3] `https://docs.tendermint.com/v0.34/tendermint-core/running-in-production.html`

7.4 Application Blockchain Interface

Connection of the application to the Tendermint engine is done using the Application Blockchain Interface (ABCI). It consists of the following (only implemented ones included, for a full list see[4]) methods:

- **check_tx:** Checks the validity of a transaction.
- **deliver_tx:** Prepares transactions for committing them to disk.
- **commit:** Commits the current state of the application to disk.
- **query:** Method used to query the state of the application.

All transactions are submitted to the blockchain using the provided API of Tendermint, which expects them in bytes. These are submitted in a predefined format, which consists of multiple parameters. These are separated with the "=" character. The first parameter of a transaction identifies the type of it. The following keywords are used for defining the format of it:

- **LOCATION:** A string identifier of a location.
- **TIMESTAMP:** A timestamp with a predefined format.
- **DEVICE-ID:** An integer that denotes the device id.
- **DESTRUCT:** A boolean that denotes whether the device shall be destroyed at the given location (used in the BLOCK transaction).
- **SIGNED-DIGEST:** The signed digest of the transaction (the hash-digest of the transaction, signed with the private key of the location's node)

Two types of transactions exist:

- **ALLOCATE:** This transaction type is used to allocate new device ids in the blockchain.

  Format: ALLOCATE=LOCATION=TIMESTAMP

- **BLOCK:** This transaction type is used to save that a device has been received at a location. More attributes regarding the device may be added if deemed beneficial.

  Format: BLOCK=DEVICE-ID=LOCATION=TIMESTAMP=DESTRUCT=SIGNED-DIGEST

  To be considered valid (checked in the check_tx method of the ABCI), it has to fulfil the following:

  - Digital signature must be correct. (ensures that the transaction was submitted by the location's node and that the content has not been altered)
  - If DESTRUCT is true, then the location must have a destruction certificate.
  - The device id must be assigned.
  - DESTRUCT must have not been true in any of the previous BLOCK transactions for that device id.
  - The last location of the device must not be the same as the given one.

[4] `https://github.com/tendermint/abci/blob/master/specification.md`

7.5 Web server

The web server is responsible for providing an easy-to-use interface to allocate new device ids and to access the trajectory of a device. It is also responsible for receiving and accepting messages from the smartphone app. It has a very simple interface that can be seen in Figure 11. It has two buttons, one for allocating a new device id, and another one for accessing the visualisation tool of the web server.

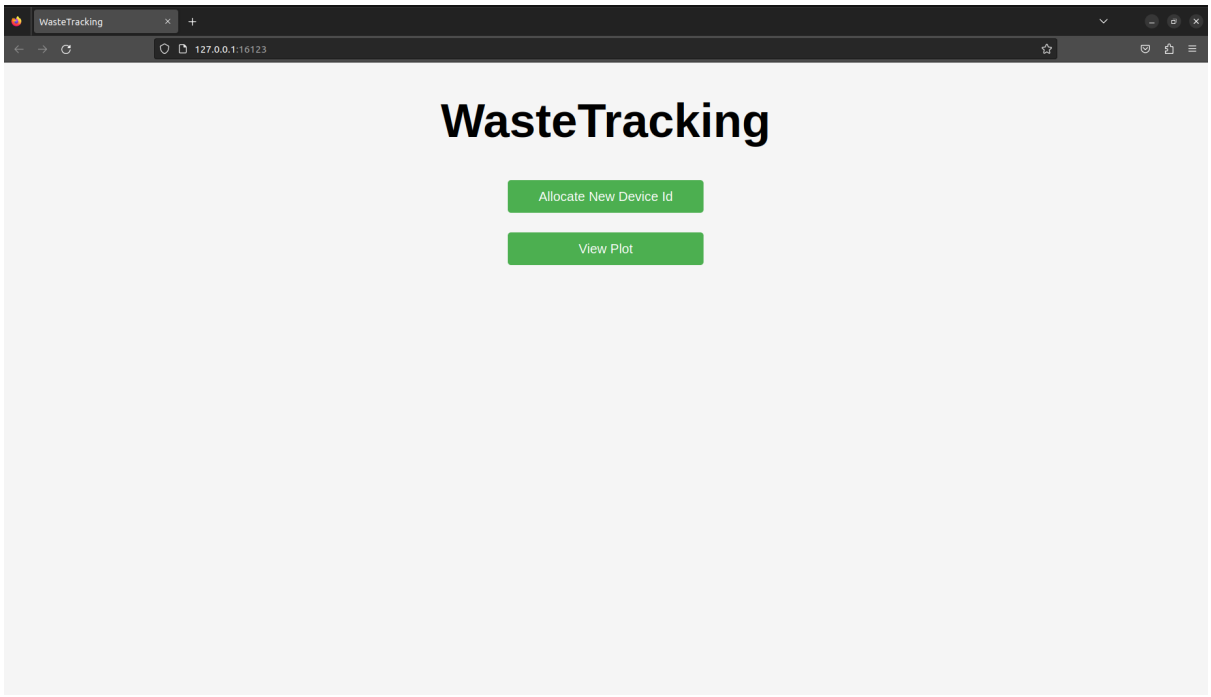


Figure 11: Website Interface

7.5.1 Allocating New Device-Ids

New device ids may be allocated by using the "Allocate New Device Id" button on the main page. This then sends a transaction to the blockchain to allocate a new device id. The blockchain then gives a response which includes the allocated device id. A barcode in the EAN13[5] format (with checksum) is then created and displayed in a new tab. This barcode may then be printed out and attached to a device for further scanning.

7.5.2 Visualisation

The web server also includes a visualisation tool. It features a single view which can be seen in Figure 12. On the top right of the site, there is an input field, which can be used to change the visualised device (its trajectory). Information on when a location was visited is displayed for each location, as well as the name of the location. Additional information, such as the name of the company that owns the recycling station of a location may be added in a later version of this system.

[5] `https://en.wikipedia.org/wiki/International_Article_Number`

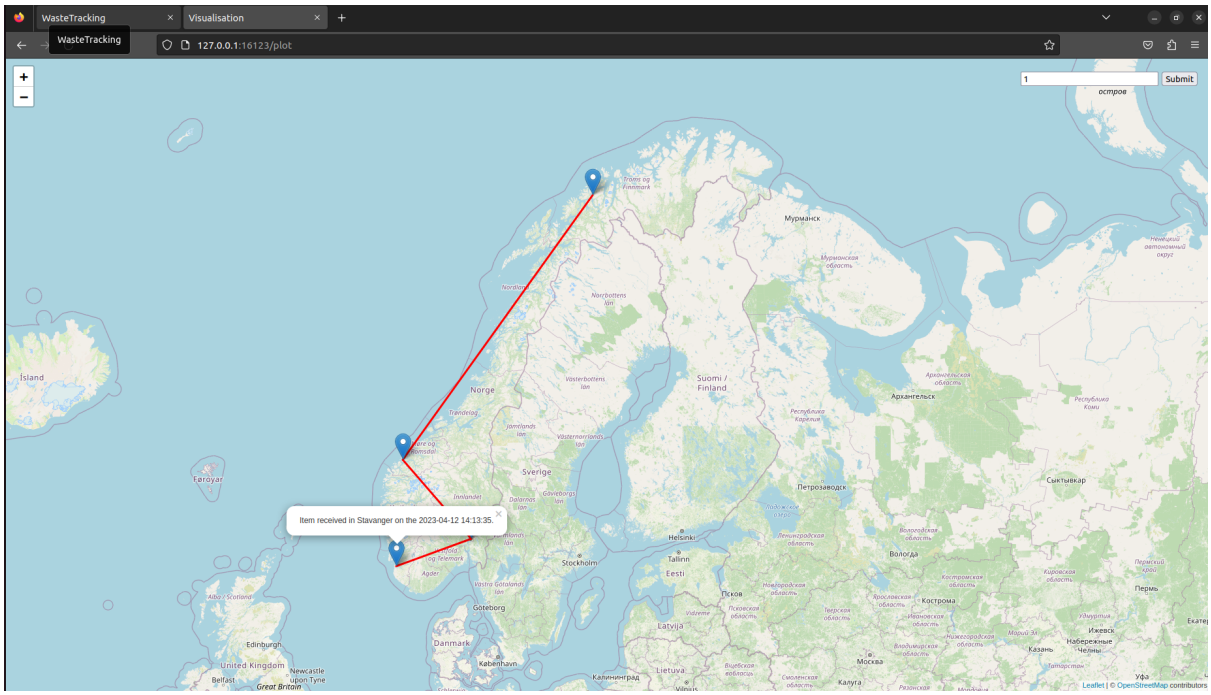


Figure 12: Visualisation Tool of the web server; Showing an Example Trajectory (Stavanger, Oslo, Ålesund, and Tromsø (in order))

7.6 Smartphone App

The smartphone app uses the device's camera to scan for any barcodes. When a barcode is scanned, the user is asked if the device shall be destroyed at the current location (which can be seen in Figure 13). After the user has selected one of the two options, this information and the barcode are sent to the web server. The web server then validates the barcode's checksum and submits a *BLOCK* transaction (which saves the intermediate location) to the blockchain.

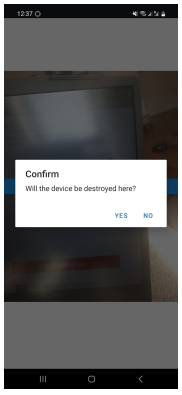


Figure 13: Smartphone App Interface (after scanning a barcode)

CHAPTER 8

EVALUATION

For analysing the performance of the blockchain and verifying that everything works, a test network of four nodes was used to test the system. As mentioned in Section 7.3, the Raspberry Pi 4 Model B was chosen. All tests were done with the nodes being in the same (local) network. This is a flaw in the evaluation and if possible should be expanded to be more realistic towards the use case. This was unfortunately not possible in the scope of this thesis.

8.1 Performance Analysis

For measuring the performance of the network, a high number of transactions was submitted to it and then it was measured how long it would take for the network to handle the transactions and how much space these make up for. It was also evaluated how the behaviour of *ALLOCATE* and *BLOCK* transactions differs.

8.1.1 Transaction Throughput

For measuring the transaction (tx) latency the experiment was conducted with 10,000 transactions. Each configuration was repeated three times to ensure reliability. The results can be seen in Table 4. The *ALLOCATE*% and *BLOCK*% indicates the likelihood of each transaction type (e.g. if *ALLOCATE*% is $\frac{1}{4}$, then approximately 2,500 transactions of the 10,000 submitted ones were of type *ALLOCATE*). The difference in mean of transactions per second (tx/s) only varies by a maximum of 0.788, which is less than 1%. This is surprising, as *BLOCK* transactions use digital signature verification and in general should be more expensive than *ALLOCATE* transactions, due to their validity checks. The standard deviation for configurations is a maximum of only 2.3% of the mean, which indicates that the system runs stable. If it is assumed that every device needs five transactions and the transaction throughput is 130 transactions per second (or about 4.1 billion per year). Then, every Norwegian (using the current population of Norway: 5.546 million) could recycle 147.94 devices each year, which may be considered sufficient.

Table 4: Transaction Throughput with 10,000 Transactions

ALLOCATE%	*BLOCK*%	Mean Tx/s	Stddev Tx/s
1/3	2/3	130.127	2.993
1/4	3/4	129.535	0.311
1/5	4/5	130.323	0.847

8.1.2 Storage Requirements

To measure how much memory the blockchain uses, 10,000 and 100,000 transactions were submitted to a single node (single-node network) and it was evaluated how much space is used. It is expected that the number of nodes in a network does not influence the storage space required of a single node with the same number of transactions. The results can be seen in Table 5. Again, different probabilities were assigned to *ALLOCATE* and *BLOCK* transactions, like in the experiment in Section 8.1.1. As expected, *BLOCK* transactions take up more space, since they contain more information. A hard drive with a storage capacity of 25 GB and $\frac{1}{5}$ ALLOCATE% can store approximately 1.1 billion transactions (assuming 100,000 transactions take up 2269.5 KB). Again, using the same assumptions as in Section 8.1.1 (each device needs five transactions, Norway has a population of 5.546 million) and every Norwegian recycling 10 devices per year. Then, a 25 GB hard drive would last for about 4 years.

Table 5: Storage Space Required for Transactions

ALLOCATE%	*BLOCK*%	Space for 10,000 Tx	Space for 100,000 Tx
1/3	2/3	233.3 KB	1945.6 KB
1/4	3/4	252.2 KB	2211.5 KB
1/5	4/5	275.2 KB	2269.5 KB

8.2 Security Analysis

Although blockchain is supposed to be a secure form of storing data across multiple nodes, like any other system, it is not completely secure. This section provides potential security risks and tries to make suggestions on how to reduce these risks further. It is important, however, to remember that this list is (very likely) incomplete.

8.2.1 Illegally-made Barcodes

Barcodes are used for scanning devices at a location. The barcode simply represents the device id (with a checksum). As a result, it is easy for an attacker to generate a new barcode for any device id. They may then scan it at any location. As a result, they could for example fraud the trajectory of a device.

One way to combat this, could be to use QR codes instead of barcodes and add a digital signature of the node (that allocated the device id) to the QR code. This way, attackers would be prevented from generating a barcode for any device id. However, they would still be able to copy and distribute a code for a device that they are in possession of. Perhaps, regularly changing the data on the QR code for a single device id could help prevent this, although, this would create the need to print out and attach new QR codes repeatedly.

8.2.2 Distribution of Private Keys of Nodes

Digital signatures are used to sign *BLOCK* transactions to verify the location. The private keys for this are stored on the hard drive of the node. An attacker could access this key and distribute it to other nodes. As a result, one node could have many private keys

then and be able to create a seemingly real trajectory of a device (adding multiple locations, even though they all came from the same node). In combination with the security risk discussed in Section 8.2.1, a single node could create transactions for all device ids for many locations.

One way to try to prevent this, could be to regularly force updates of digital signatures (private keys). Although, an attacker could simply redistribute the key every time it was changed.

Another way to reduce the security risk of this would be to make it harder to access keys. Currently, they are stored on the hard drive of a node. Storing the keys on specialised hardware would reduce this security risk, although it would introduce additional costs.

8.2.3 Controlling a Significant Percentage of Validators

Like in any blockchain environment, if an attacker controls enough nodes, they will be able to control the network. In the case of Tendermint, if an attacker controls more than $\frac{1}{3}$ of nodes, then they can stop the normal operation flow. If they control more than $\frac{2}{3}$ of nodes, then they can dictate the operation flow entirely.

To reduce this risk, there should be a sufficiently large number of nodes in the network, as this makes it harder for any attacker to control a significant percentage of validators.

Furthermore, as in this use case, nodes are located physically on the recycling centres of waste companies. As such, if a company wanted to, they could seize control of all of their nodes (which are located on their recycling centres) and choose them to behave arbitrarily. As a result, no company should control more than $\frac{1}{3}$ of the nodes. Although, multiple companies could always reach an agreement and thus reach the critical percentage of controlled nodes.

8.2.4 Encryption Breakdown

Current encryption methods rely on the fact that current computing capabilities are not strong enough to break them. It is however theoretically possible [30]. Quantum computers could in the future provide these computing capabilities to break encryption methods [31]. Although this is only a theoretical concern, that may never come true, it is still important to consider.

8.2.5 Message Flooding (DDOS Attack)

A node could flood the network with transactions and thus stop the normal flow (at least temporarily) of operations. Transactions may get lost in the process, which an attacker could abuse.

Implementing a safeguard for nodes that limits how many transactions they are allowed to send could be used to limit this threat. However, it should be noted that this threshold should not be set too low, as otherwise, this might hinder the normal operation flow.

8.2.6 Human Errors

In the process of adding locations for a node, a human has to scan a barcode to add it. They then have to specify whether the device shall be destroyed at the scanned location.

They could accidentally enter the wrong value here and thus add incorrect data to the blockchain. And since one of the validity checks of a *BLOCK* transaction (see Section 7.4) is to check that a device has not been destroyed yet, this would prevent the addition of any further locations for that device. Even if the human does not make this mistake, there could still be a change of plans, which would make the data on the blockchain invalid.

A way to deal with this issue could be to add a mechanism which enables changing of data. Although, this could introduce security risks and as such must be handled carefully.

CHAPTER 9

CONCLUSION AND FUTURE WORK

In this thesis, a survey was conducted to evaluate the relationship of the public with electronic waste. Additionally, a blockchain system to track electronic waste was developed.

The survey was answered by 114 people and it was concluded that more than 80% of people keep two or more old unused devices at home. Furthermore, it was asked why respondents keep their devices at home, for which they could list multiple reasons. Most people (61.5%) think that they might need the device someday, the next highest reason was that they were afraid of data abuse (45.8%). The waste tracking system developed in this thesis is supposed to help gain the public's trust in the recycling process. 55.4% of people answered that a tracking system like the one proposed in this thesis would encourage them to submit their devices. While this percentage is significant, further ways to encourage recycling should be explored in the future.

The blockchain system developed in this thesis is built on top of the blockchain consensus engine Tendermint which uses a Byzantine fault-tolerant consensus algorithm. The blockchain was tested on a test network with four nodes and achieved a transaction throughput of about 130 transactions per second. This allows every Norwegian to submit about 148 devices every year, which is considered sufficient. Furthermore, it was approximated that a 25 GB hard drive could store 1.1 billion transactions, which under assumptions would last for about four years. The system can be considered as an efficient way of tracking WEEE.

Unfortunately, the system still has flaws and as such, further refinement and testing of the system would be beneficial. Improvements could be, but are not limited to the following:

9.1 Scalability Testing

The network to test transaction throughput in this thesis is only limited to a network of four nodes (see Section 8.1.1). In the proposed use case, however, it can be expected that the number of nodes is in the hundreds or perhaps even thousands area. Furthermore, they are expected to be all over the country, while in the test, they were in the same network. As such, it would be beneficial to test how the transaction throughput changes with more realistic conditions and if it is still practical. (although a similar system has been tested with 200 nodes and achieved a transaction throughput of 74.15 transactions per second, see Section 6.1)

9.2 Security Flaws

In Section 8.2, security flaws are examined and possible solutions are given. Unfortunately, it was not possible to implement these within the scope of this thesis. Furthermore, the list of security flaws is (very likely) incomplete and the proposed solutions are not 100% effective. As such, it would be advantageous to conduct a more thorough security analysis and implement solutions to combat these flaws.

9.3 User Testing

The system's interaction points with users are designed to be easy to use. However, this has not been tested with any users. As such, it might be flawed. As a result, it would be valuable to conduct an experiment with users. That way, it would be possible to see if they can correctly interact with the system and to find out what they think needs improvement.

REFERENCES

[1] Rahul Rautela et al. "E-waste management and its effects on the environment and human health". In: *Science of the Total Environment* 773 (2021), p. 145623.

[2] Lynda Andeobu, Santoso Wibowo, and Srimannarayana Grandhi. "An assessment of e-waste generation and environmental management of selected countries in Africa, Europe and North America: A systematic review". In: *Science of the Total Environment* 792 (2021), p. 148078.

[3] Sohani Vihanga Withanage and Komal Habib. "Life cycle assessment and material flow analysis: two under-utilized tools for informing E-waste management". In: *Sustainability* 13.14 (2021), p. 7939.

[4] Atta Ur Rehman Khan and Raja Wasim Ahmad. "A blockchain-based IoT-enabled E-Waste tracking and tracing system for smart cities". In: *IEEE Access* 10 (2022), pp. 86256–86269.

[5] Neha Gupta and Punam Bedi. "E-waste management using blockchain based smart contracts". In: *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE. 2018, pp. 915–921.

[6] Raja Wasim Ahmad et al. "Blockchain-based forward supply chain and waste management for COVID-19 medical equipment and supplies". In: *Ieee Access* 9 (2021), pp. 44905–44927.

[7] Thomas K Dasaklis, Fran Casino, and Constantinos Patsakis. "A traceability and auditing framework for electronic equipment reverse logistics based on blockchain: The case of mobile phones". In: *2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA*. IEEE. 2020, pp. 1–7.

[8] David Lee et al. "Monitour: Tracking global routes of electronic waste". In: *Waste management* 72 (2018), pp. 362–370.

[9] Amit Dua et al. "Blockchain-based E-waste management in 5G smart communities". In: *IEEE INFOCOM 2020-IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE. 2020, pp. 195–200.

[10] Don Johnson, Alfred Menezes, and Scott Vanstone. "The elliptic curve digital signature algorithm (ECDSA)". In: *International journal of information security* 1 (2001), pp. 36–63.

[11] Fabian Knirsch, Andreas Unterweger, and Dominik Engel. "Implementing a blockchain from scratch: why, how, and what we learned". In: *EURASIP Journal on Information Security* 2019 (2019), pp. 1–14.

[12] Satoshi Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: *Decentralized business review* (2008), p. 21260.

[13] Bela Shrimali and Hiren B Patel. "Blockchain state-of-the-art: architecture, use cases, consensus, challenges and opportunities". In: *Journal of King Saud University-Computer and Information Sciences* 34.9 (2022), pp. 6793–6807.

[14] Leslie Lamport, Robert Shostak, and Marshall Pease. "The Byzantine generals problem". In: *Concurrency: the works of leslie lamport*. 2019, pp. 203–226.

[15] Miguel Castro, Barbara Liskov, et al. "Practical byzantine fault tolerance". In: *OsDI*. Vol. 99. 1999. 1999, pp. 173–186.

[16] Marko Vukolić. "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication". In: *Open Problems in Network Security: IFIP WG 11.4 International Workshop, iNetSec 2015, Zurich, Switzerland, October 29, 2015, Revised Selected Papers*. Springer. 2016, pp. 112–125.

[17] Eleftherios Kokoris-Kogias et al. "Omniledger: A secure, scale-out, decentralized ledger via sharding". In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 583–598.

[18] Leo Maxim Bach, Branko Mihaljevic, and Mario Zagar. "Comparative analysis of blockchain consensus algorithms". In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Ieee. 2018, pp. 1545–1550.

[19] David Schwartz, Noah Youngs, Arthur Britto, et al. "The ripple protocol consensus algorithm". In: *Ripple Labs Inc White Paper* 5.8 (2014), p. 151.

[20] David Mazieres. "The stellar consensus protocol: A federated model for internet-level consensus". In: *Stellar Development Foundation* 32 (2015), pp. 1–45.

[21] Stefano De Angelis. "Assessing security and performances of consensus algorithms for permissioned blockchains". In: *arXiv preprint arXiv:1805.03490* (2018).

[22] Weidong Fang et al. "Digital signature scheme for information non-repudiation in blockchain: a state of the art review". In: *EURASIP Journal on Wireless Communications and Networking* 2020.1 (2020), pp. 1–15.

[23] Ronald L Rivest, Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[24] PUB FIPS. "Digital Signature Standard, Federal Information Processing Standards Publication 186". In: *US Department of Commerce, National Institute of Standards and Technology (NIST), National Technical Information Service. Springfield, Virginia* (1994).

[25] Sara Artang and Afshin Ghasemian. "E-WASTE TRACKER: A PLATFORM TO MONITOR E-WASTE FROM COLLECTION TO RECYCLING". In: ().

[26] Karl Wüst and Arthur Gervais. *Do you need a Blockchain?* URL: `https://eprint.iacr.org/2017/375.pdf` (visited on 02/05/2023).

[27] Guido Ongena et al. "Blockchain-based smart contracts in waste management: a silver bullet?" In: (2018).

[28] Jae Kwon. "Tendermint: Consensus without mining". In: *Draft v. 0.6, fall* 1.11 (2014).

[29] Andreas Unterweger et al. “Lessons learned from implementing a privacy-preserving smart contract in ethereum”. In: *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE. 2018, pp. 1–5.

[30] Zach Kirsch and Ming Chow. “Quantum computing: The risk to existing encryption methods”. In: *Retrieved from URL: http://www. cs. tufts. edu/comp/116/archive/fall2015/zkirsch. pdf* (2015).

[31] Dorothy E Denning. “Is Quantum Computing a Cybersecurity Threat? Although quantum computers currently don’t have enough processing power to break encryption keys, future versions might”. In: *American Scientist* 107.2 (2019), pp. 83–86.

APPENDIX

A - GITHUB REPOSITORY

The source code is available in the corresponding GitHub repository: `https://github.com/marc131183/e-waste-blockchain`