NOVA IMS Information Management School

MDSAA

Master Degree Program in
Data Science and Advanced Analytics

Blockchain and Electronic Voting

An Electronic Voting Hyperledger Fabric System based on the Portuguese electoral voting system

Nuno Reis Henriques Dias

Project Work

presented as partial requirement for obtaining a Master's Degree in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

Blockchain and Electronic Voting

An Electronic Voting Hyperledger Fabric System based on the Portuguese electoral voting system

by

Nuno Reis Henriques Dias

Project Work presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a specialization in Data Science

Supervised by

Prof. Filipe Montargil, PhD, Nova IMS, Information Management School

Eng. Dino Coutinho, MSc, INM, Innovation Makers

July 2024

Statement of Integrity

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

[Lisbon, 15th of July of 2024]

Acknowledgments

I would like to express my deepest gratitude to my supervisors, Prof. Filipe Montargil and Eng. Dino Coutinho, for their invaluable guidance, support, and encouragement throughout the course of my research. Their expertise and insights have been instrumental in shaping this work, and I am profoundly grateful for their mentorship.

My sincere thanks go to the Innovation Makers team, especially Eng. Paulo Nunes Filipe, for their collaboration and support. Your contributions have significantly enriched my research experience.

My heartfelt thanks go to my parents, whose unwavering support and belief in me have been the bedrock of my journey. Their love and sacrifices have made this achievement possible. To my sister, thank you for your constant encouragement and understanding. Your presence has been a source of strength and comfort.

I am also deeply grateful to my extended family, including my uncles, aunts, cousins, and especially my grandparents, who would have loved to see me accomplish this and have been a constant source of inspiration. Your collective support, love, and faith in my abilities have been a continuous source of motivation.

Lastly, I would like to acknowledge my friends for their companionship and support. Your encouragement and the moments of respite you provided have been invaluable in maintaining my focus and motivation.

Thank you all for being a part of this journey with me.

Abstract

Blockchain technology, with its decentralized and secure nature, has emerged as a promising solution for enhancing the integrity and transparency of Electronic Voting systems. This thesis explores the application of blockchain in Electronic Voting, specifically within the context of the Portuguese electoral system. It begins with a comprehensive overview of blockchain technology, including its core features such as decentralization, immutability, and transparency, as well as consensus protocols like Proof of Work and Proof of Stake. These foundational elements underscore blockchain's potential to revolutionize Electronic Voting by addressing existing challenges such as security vulnerabilities, susceptibility to fraud, and lack of transparency. The core of this work involves the design and implementation of a blockchain-based Electronic Voting system using Hyperledger Fabric, a permissioned blockchain framework tailored to the specific requirements of the Portuguese electoral process. The system ensures voter privacy, vote integrity, and transparency through smart contracts that automate and secure voting procedures, from voter identification to vote tallying. Biometric data integration further enhances voter authentication, reducing the risk of fraud. Despite promising results, challenges such as scalability, secure voter devices, and balancing transparency with privacy remain. Overall, the findings suggest that blockchain-based Electronic Voting systems have the potential to make elections more secure, transparent, and reliable, offering valuable insights for future research and development in this field.

KEYWORDS

Blockchain; Electronic Voting; Hyperledger Fabric; Smart Contracts; Chaincode

Table of Contents

List of Tables

List of Figures

Introduction

Blockchain has emerged as a revolutionary technology, reshaping transaction management by eliminating the need for third-party intermediaries (Di Pierro, 2017). This transformative capability finds a promising application in Electronic Voting, a domain that has been under study for over two decades (Montargil, 2004). Despite facing challenges that led to a pause in its evolution, Electronic Voting is experiencing renewed interest, thanks to the decentralization, security, and speed attributes introduced by Blockchain.

In this work, we attempt to develop an on-site Electronic Voting system using Blockchain for a small-scale election, drawing upon the insights gained from an extensive review of both Blockchain and Electronic Voting with Blockchain. Our contributions encompass a comprehensive state-of-the-art analysis in these domains and the synthesis of methodologies to inform the creation of our Electronic Voting with Blockchain system.

Our objective is to leverage existing knowledge to implement a pilot that closely resembles the traditional voting process in Portugal. By doing so, we aim to facilitate a gradual transition to this new technology, ensuring that the adoption process can be smooth and does not require a significant leap from the current system. This approach allows for the incremental introduction of Blockchain technology into the voting process, maintaining user familiarity and trust while progressively enhancing security and transparency.

The document's overview commences with an exploration of Blockchain in Chapter 1, covering its origin and characteristics (Section 1.1), types of Blockchain (Section 1.2), consensus protocols (Section 1.3), challenges and issues (Section 1.4), and Blockchain frameworks (Section 1.5). Chapter 2 then delves into the state of the art of Electronic Voting with Blockchain, starting with the definitions, advantages, and types of Electronic Voting (Section 2.1). It addresses the introduction of Electronic Voting, how Blockchain can improve Electronic Voting (Section 2.2), limitations that Electronic Voting with Blockchain still presents (Section 2.3), applications of Electronic Voting with Blockchain (Section 2.4), Blockchain Frameworks for Electronic Voting (Section 2.5), Cryptography for Electronic Voting (Section 2.6), and Electronic Voting in Portugal (Section 2.7).

Chapter 3 engages in a detailed discussion on the creation of a conceptual Electronic Voting with Blockchain system, based on the Portuguese electoral voting system. It includes sections on Voting Laws in Portugal (Section 3.1), Hyperledger Fabric (Section 3.2), Encryption (Section 3.3), System Architecture (Section 3.4), Working Process (Section 3.5), and Smart Contracts and Chaincode (Section 3.6).

1. Blockchain

Blockchain technology, with its decentralized and transparent nature, has gained widespread popularity in recent years, following the emergence of Bitcoin cryptocurrency, by far the most successful Blockchain application (Ghiro et al., 2021). It revolutionizes the traditional concept of centralization, where one party is in control of everything, offering enhanced efficiency and accessibility (Rajasekaran et al., 2022). The chapter is systematically divided into five key sections to cover the foundational concepts, different types, underlying protocols, associated challenges, and available frameworks. The structure is designed to ensure a logical flow and to facilitate a deeper understanding of the subject.

1.1. Concept and Key Features

Blockchain was first introduced by Nakamoto (2008) in the paper "Bitcoin: A Peer-to-Peer Electronic Cash System." This system allows online payments to be sent directly from one party to another without the need for a trusted third party by using a peer-to-peer distributed timestamp server. In a Bitcoin system, transactions are secured through a chain of digital signatures. Each owner digitally signs the previous transaction's hash along with the next owner's public key, allowing payees to verify the chain of ownership. To prevent double-spending, the system implements a timestamp server that publicly timestamps data through hashing, forming a chain of timestamps. This ensures a transparent record of transactions.

To secure the distributed timestamp server on a peer-to-peer basis, a proof-of-work system is employed. This mechanism, detailed further in subsequent sections, also establishes the majority decision in the network. The longest chain, representing the most significant proof-of-work effort, becomes the accepted chain. This prevents malicious actors from tampering with the system, as they would need to redo the proof-of-work for each block and surpass the work of honest nodes. In the Bitcoin network, new transactions are broadcast to all nodes, which then collect these transactions into a block. Nodes work on finding a difficult proof-of-work for the block, and once this proof-of-work is found, the block is broadcast to the network. The block is accepted only if all transactions within it are valid. Nodes continually work on extending the longest chain. In the event of simultaneous conflicting versions of the Blockchain, they switch to the longer branch when the next proof-of-work is found, as argued by Nakamoto (2008).

The security of the system is maintained by the proof-of-work system, which ensures the implementation of the distributed timestamp server on a peer-to-peer basis. After the computational effort has been made to meet the proof-of-work requirements, altering a block becomes impractical without redoing all subsequent work. Since blocks are sequentially chained, modifying a single block would require redoing the work for all subsequent blocks in the chain.

Blockchain, as a technology in constant evolution, presents an ongoing challenge in defining its features, as diverse authors like Xinyi et al., Bhutta et al., and Zarrin et al. offer distinct perspectives on this technology. Among the key features emphasized by a number of scholars, as shown in Figure 1.1, are decentralization, distributed consensus, immutability, anonymity, transparency, programmability, and security.

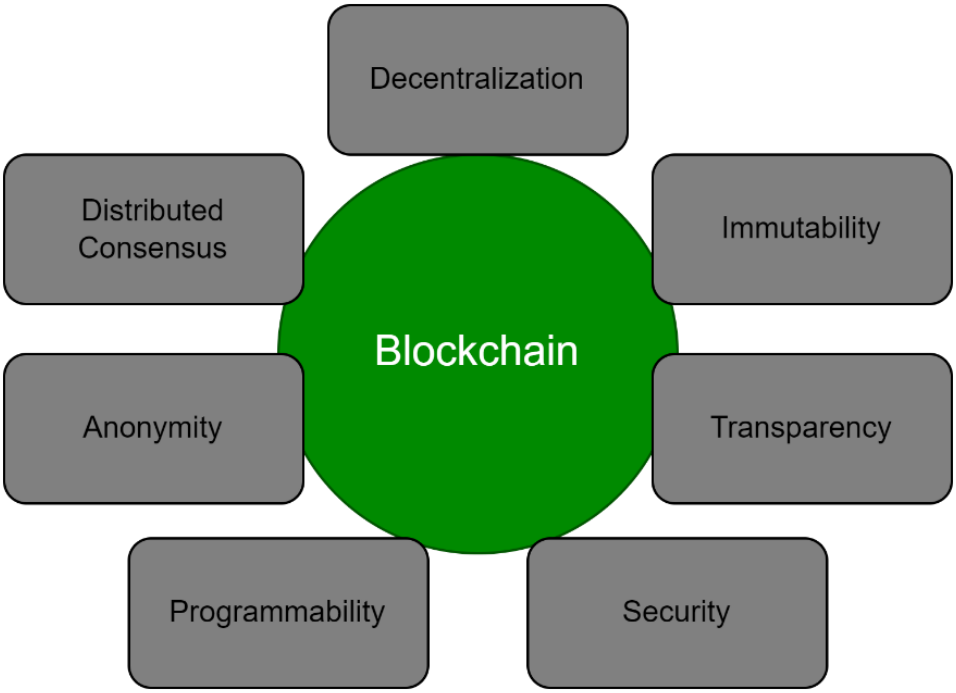

Figure 1.1: Blockchain Characteristics

Regarding decentralization, the most important and essential feature of Blockchain according to Xinyi et al. (2018, p. 3) and Bhutta et al. (2021, p. 4), the system uses a distributed structure for recording, storing, updating, transmission, verification, maintenance, and several other processes related to the information in the Blockchain network. Operating on multiple computers, often referred to as nodes that work in a peer-to-peer manner, each transaction is conducted by two nodes at a time, allowing the system to be non-reliant on a central authority. This essentially enables nodes to have equal voting rights within the network (Zarrin et al., 2021, p. 7). Chu & Wang (2018, p. 2) and Xinyi et al. (2018, p. 3) argue that decentralization is a key component of Blockchain, democratizing trust through the use of pure mathematical methods for establishing relationships between chain nodes.

Appending new transactions to the Blockchain relies on a distributed consensus protocol (Hjalmarsson et al., 2018, p. 1). For any proposed block of entries to become a permanent part of the Blockchain, a consensus among the majority of the network must be reached. Chatterjee & Chatterjee (2017, p. 1) highlighted that in a distributed system, a copy of the Blockchain is present with all its members, making it more secure than centralized institutions like financial banks. By distributing vital data across the network, the Blockchain becomes more resilient against potential attacks. In case an attacker compromises the

Blockchain, it is highly unlikely they could successfully alter it (Twesige, 2015, p. 5). Consensus can be reached in different ways, depending on the technical approach used, such as proof-of-work and proof-of-stake (Zīle & Strazdiņa, 2018, p. 2), which help nodes make the right decisions as argued by (Panwar & Bhatnagar, 2020, p. 2).

Immutability is a characteristic guaranteed in Blockchain. The system employs a chain structure composed of timestamped data blocks closely connected in chronological order for the storage of information. Each block is encrypted using a hash algorithm and permanently recorded upon generation, making it resistant to tampering or deletion unless over 51% of the nodes within the entire system collude (Xinyi et al., 2018, p. 3). The time-stamped and hash-encrypted structure of Blockchain entries ensures permanence, as any attempt at tampering will generate a different hash, which can be detected through a consensus of the majority of nodes within the system, as explained by Bhutta et al. (2021, p. 5).

Anonymity, as outlined by Bhutta et al. (2021, pp. 5–6), supports privacy, defined as protection from unauthorized intrusion or observation. This anonymity is achieved by validating transactions without disclosing personal information about the involved parties. Data is exchanged between nodes using a defined algorithm that establishes trust, hence the information of the nodes does not need to be revealed or verified, allowing information transfer to be carried out anonymously. Users in a Blockchain system can interact with generated Blockchain addresses to keep their real identities hidden. Zarrin et al. (2021, p. 7) note that each miner uses a generated address as a unique identifier, reinforcing the core principle of maintaining anonymity within the network. While not all Blockchains are entirely anonymous, with some practicing pseudo-anonymity (such as Ethereum and Bitcoin), the fundamental objective remains to ensure users can remain anonymous within the network.

Transactions on Blockchain, as argued by Bhutta et al. (2021, pp. 4–5), are completely transparent, meaning anyone can see the details and history of any transaction. This level of transparency is unique to Blockchain technology and provides high accountability and integrity to the information, ensuring that nothing is unduly altered, deceitfully added, or removed. The decentralized nature of the Blockchain network, with several validating peer nodes and no centralized authority, contributes to this level of transparency. The holdings and transactions associated with each public address are accessible and open for viewing by anyone, creating traceable and transparent transaction records. The significance of transparency extends beyond financial systems, finding applications in healthcare, clinical trials, supply chain management, and public elections. According to Bhutta et al. (2021, p. 5), transparency ensures high visibility, easy tracking, and verification of transactions, contributing to the auditability of large companies.

Blockchain technology, known for its programmability, offers a dynamic script code system, providing users the flexibility to create advanced smart contracts, currencies, or other decentralized applications, as highlighted by Xinyi et al. (2018, p. 3). This scripting capability not only optimizes transaction formats but also enhances the overall economic efficiency of society. Bhutta et al. (2021, p. 6) emphasize the open-source nature of Blockchain technology, allowing users to develop applications through a shared application programming interface. The programmability of Blockchain can be tailored to meet specific

needs, such as developing a financial payment system by individuals with software development skills (Twesige, 2015, p. 5).

Blockchain systems inherently prioritize security through the use of asymmetrical cryptography, as outlined by Bhutta et al. (2021, p. 5). This cryptographic approach involves a set of public keys visible to anyone and a set of private keys visible only to the owner, ensuring the ownership and immutability of transactions. Security in the Blockchain system encompasses aspects of integrity, confidentiality, and authorization. The distributed nature of the Blockchain system, relying on a peer-to-peer consensus mechanism, eliminates a single point of failure for data, in stark contrast to centralized storage, which is more vulnerable to compromise. Xinyi et al. (2018, p. 3) contribute to this understanding by emphasizing the security and immutability ensured by the chain structure of data blocks with timestamps, encrypted with hash algorithms, and permanently recorded.

1.2. Types of Blockchain

Blockchain technology can be classified into three distinct types, each designed to fulfill specific organizational needs and operational criteria. The three primary types, as shown in Figure 1.2—Public, Private, and Consortium—offer varied structures and functionalities.

Figure 1.2: Types of Blockchain

In the realm of permissionless or Public Blockchains, participants enter the network without requiring any permission, embracing a truly decentralized structure. Participants engage in the consensus process, transact, and uphold the shared ledger, fostering an environment where new blocks are published, accessed, and validated by all. Security in Public Blockchains is fortified through computationally intensive consensus mechanisms, such as puzzle-solving or staking one's cryptocurrency, ensuring the tamper-proof nature of block contents through hashes and decentralized consensus, as highlighted by Bhutta et al. (2021, p. 9). Public Blockchains, such as Bitcoin, Ethereum, Bitshares, Ripple, or Hyperledger, are completely decentralized, distributed, and open digital ledgers accessible worldwide. While these Blockchains ensure data security through encryption technology, operational costs and transaction speed pose challenges, according to Xinyi et al. (2018, p. 2). They exhibit a transparent record of transactions, allowing anyone worldwide to verify the ledger's correctness and participate in the validation process. However, the anonymity of users poses security concerns, addressed by stringent consensus protocols like the Proof of Work (PoW) mechanism in Bitcoin and Ethereum. Despite this, transaction throughput is limited compared to traditional systems like Visa, as outlined by Ghiro et al. (2021, p. 2).

In the realm of permissioned or Private Blockchains, participants are allowed to join the network by invitation and play specific roles to maintain the Blockchain in a decentralized manner, as Bhutta et al. (2021, p. 9) highlighted. Private Blockchains differ from Public Blockchains as only authorized entities are allowed to join the network and maintain the blocks. Security in Private Blockchains is fortified as only known participants can join the network. Private nodes from an organization or group, restricted from the public, handle the verification and consensus processes, according to Zarrin et al. (2021, p. 8). Not every node can participate in these processes, even if they belong to the same organization. Open research issues in Private Blockchains include the potential tampering of blocks and network hacking by internal authorized participants. Strict management of writing and reading authority is required for data access and usage, introducing a degree of centralization control. Xinyi et al. (2018, p. 2) exemplify R3 Corda as a representative of Private Blockchains, which find applications in scenarios such as database management, internal audits within enterprises, or government statistics.

Private Blockchains, designed by large corporations or banks, rely on classical authentication mechanisms for security, deviating from the consensus protocols of Public Blockchains (Ghiro et al., 2021, pp. 2–3). The trust model in Private Blockchains allows the use of faster and more efficient consensus protocols, vital for supporting critical business operations. Unlike permissionless Blockchains, permissioned ledgers free users from traditional trusted authorities, such as Public Key Infrastructure (PKI) or banks, while maintaining transparency in public validation processes without revealing user identities.

Consortium Blockchains, as Bhutta et al. (2021, p. 9) argue, are designed for multiple organizations, exclusively permit invited and trusted participants to join and maintain the network, ensuring heightened security measures against tampering and hacking due to the involvement of participants from various organizations. Labeled as a permissioned Blockchain, consortium Blockchains select nodes from public or private branches to manage verification and consensus processes, representing a hybrid between Public and Private Blockchains. Examples, such as Hashed Health and IBM/Maersk in the financial and health industries, illustrate the application of Consortium Blockchains, according to Zarrin et al. (2021, p. 8). Internally pre-selecting nodes as bookkeepers, Consortium Blockchains control the consensus process, requiring authorization for entry and exit. While other nodes can participate in transactions, they lack accounting rights. Fabric serves as a representative Consortium Blockchain, offering higher transaction processing speed, lower transaction and maintenance costs, and superior data security compared to Public Blockchains as outlined by Xinyi et al. (2018, p. 2). Sankar et al. (2017, p. 2) distinguish Consortium Blockchains from fully Private Blockchains, emphasizing their centralized structure and decision-making authority, with the consensus process controlled by a single entity.

1.3. Consensus Protocol

The consensus mechanism is used to validate transactions and reach a consensus for updates in the Blockchain. The most commonly used mechanisms, as noted by various authors, are Proof of Work (PoW), first mentioned by Nakamoto (2008), and Proof of Stake (PoS).

Proof of Work (PoW) is a pivotal consensus mechanism in Blockchain technology, notably employed by Bitcoin, as highlighted by Bhutta et al. (2021, p. 14). It functions by initiating a competition among network nodes to achieve consensus in a decentralized and distributed Blockchain network. The consensus algorithm determines how nodes agree to append a new block to the chain, requiring the initiator node to employ cryptographic algorithms to produce a winning value lower than the network's predetermined threshold. If multiple nodes produce a value, the network resolves this by selecting the maximum PoW value, signifying the highest amount of work performed by a node. This node is then authorized to add a block and earn a reward, making PoW suitable for scalable Blockchain networks. However, challenges include the high cost of mining equipment, a low transaction rate, and susceptibility to attacks.

PoW involves cryptographic computations, particularly determining a nonce that results in a block hash with a specified number of leading zeros. Miners solve this problem by appending a block to the chain, and others validate the correctness of the work. Despite being resource-intensive, PoW ensures accuracy and verifiability. However, controlling a subset of miners can confirm illegal blocks, and the difficulty of finding the nonce cannot be arbitrarily small to prevent excessive forking and double spending (Gao et al., 2018, p. 4). Its major flaw lies in significant power wastage for calculations, hindering IoT devices from competing due to high computing power requirements, according to Zarrin et al. (2021, pp. 11–12). As argued by Oyinloye et al. (2021), while the node discovering a block receives rewards, the process consumes immense energy, equivalent to powering millions of households. Drawbacks include low throughput, decentralization challenges, and susceptibility to the 51% attack, where an adversary accumulates superior computing assets.

The core idea of Proof-of-Stake (PoS), as argued by Ferdous et al. (2020, pp. 15–16), revolves around nodes demonstrating ownership of a specific quantity of coins to participate in the block creation process. These nodes must commit a designated amount of their currency, referred to as stake, to an escrow account to qualify for involvement. The stake serves as a pledge, ensuring obedience to protocol rules. The node placing its stake in escrow is referred to in PoS terminology as the stakeholder, leader, forger, or minter, and risks losing their stake in the event of misconduct.

By escrowing its stake, a stakeholder becomes a member of an exclusive group with the privilege to participate in the block creation process. When given the opportunity to create a new block, the stakeholder is rewarded either by collecting transaction fees within the block or receiving a specified amount of currency, functioning as interest against their stake. This incentive structure, combined with potential punitive measures, can grant security comparable to the PoW algorithm. PoS stands out as a notable alternative to PoW with

distinct advantages. Bhutta et al. (2021, p. 15) emphasize the cost-effectiveness of PoS, as it eliminates the need for expensive mining equipment. Instead, nodes engage in validation based on the number of coins they hold, promoting a more inclusive approach to network participation. (Lashkari & Musilek (2021) stress PoS's role as an energy-efficient alternative, particularly citing its use in transitioning from PoW in Blockchain evolution. Ferdous et al. (2020, pp. 15–16) highlight benefits such as energy efficiency, mitigation of centralization, and explicit economic security through penalty mechanisms.

1.4. Challenges and Issues

Despite the advantages Blockchain can offer compared to central authorities, this technology still faces many challenges and issues that need to be solved or improved, as highlighted by various authors. Blockchain systems currently grapple with scalability challenges, characterized by low throughput, high transactional latency, and escalating resource demands. As noted by Bhutta et al. (2021, p. 7), as transaction volumes increase, so does the demand for storage space. For instance, Bitcoin's size reached 158 GB in September 2017, leading to prolonged bootstrap times for new nodes. This burgeoning storage requirement raises concerns about the potential concentration of control among large businesses, risking fraudulent activities. Panwar & Bhatnagar (2020, p. 3) underscore scalability challenges in permissionless Blockchains like Bitcoin, where block size constraints limit transaction processing speed. Yang et al. (2019, p. 17), discuss the scalability trilemma, which balances scalability with decentralization and security, and suggest integrating edge computing and off-mainchain protocols as an effective solution for scalable data storage and computation. Gao et al. (2018, p. 9) emphasize the continuous growth of Blockchain size and its adverse effects on storage costs and network distribution speed, presenting a persistent challenge.

Blockchain systems also face significant performance challenges, including throughput bottlenecks, transaction latency, and storage constraints, as highlighted by Bhutta et al. (2021, p. 7). The execution of smart contracts in current Blockchains is hindered by a serial process, limiting throughput. Zarrin et al. (2021, p. 10) highlight the need for innovative solutions, such as the PvScheme system, which employs a probabilistic verification scheme to reduce forking occurrences, thereby minimizing network delay and mitigating forking attacks. Performance bottlenecks resulting from the extended verification time of limited block sizes present critical challenges. Panwar & Bhatnagar (2020, p. 3) emphasize scalability and transaction speed issues in permissionless Blockchains like Bitcoin, which can process only 4-8 transactions per second due to block size limitations. These insights underscore the multifaceted nature of performance issues in Blockchain systems and advocate for innovative solutions to enhance scalability and transaction speed.

The energy-intensive nature of the Proof of Work (PoW) consensus mechanism, particularly in the context of Bitcoin, raises significant environmental concerns. Bhutta et al. (2021, p. 7) highlight the staggering energy consumption of almost 15.77 terawatt-hours, equivalent to 0.08% of the world's electricity consumption, required for consensus through PoW. The

primary culprit behind this energy expenditure is the irreversible SHA256 hashing function used in the PoW approach. This resource-intensive design poses a threat to the global climate and contributes to greenhouse gas emissions. Panwar & Bhatnagar (2020, p. 4) elaborate on the electricity consumption aspect, emphasizing that permissionless or public Blockchains employing PoW consensus algorithms consume vast amounts of electricity for extensive computing processes, particularly in mining activities. These insights underscore the urgent need to explore more sustainable consensus mechanisms to mitigate the environmental impact of Blockchain systems.

While Blockchain systems initially offer robust security and privacy, they face evolving challenges that require careful consideration. Zarrin et al. (2021, p. 10) reveal vulnerabilities in Blockchain anonymity, where both public and private keys can be compromised, leading to the extraction of users' private information and potential erasure of stored data in nodes. Additionally, identity tracing becomes a concern due to the continuous use of the same false address by nodes, as Blockchain lacks mechanisms to refresh new false addresses. Panwar & Bhatnagar (2020, p. 4) emphasize privacy challenges in public Blockchain systems like Bitcoin or Litecoin, where despite data encryption and hidden user identities, privacy remains a significant concern. Gao et al. (2018, p. 9) highlight privacy issues, noting the traceability of transactions through IP addresses and potential risks associated with third-party applications tracking user profiles and data. Yang et al. (2019, p. 17) address security and privacy challenges introduced by outsourcing services at the edges in integrated Blockchain systems, noting controversies in off-chain solutions and efforts to address lost transactions in extreme cases of node crashes. These insights underscore the evolving nature of security concerns in the ever-expanding Blockchain ecosystem.

1.5. Blockchain Frameworks

Blockchain has garnered global attention due to its various characteristics, as discussed earlier. Among the three most common Blockchains—Bitcoin, the pioneer, Ethereum, and Hyperledger—each exhibits distinct aspects.

Bitcoin introduced the groundbreaking concept of Blockchain-based peer-to-peer (P2P) digital currency systems, marking a seminal contribution to the cryptocurrency landscape, as argued by Bhutta et al. (2021, p. 12). This research laid the foundation for direct digital payments, eliminating the reliance on trusted third-party financial systems. Bitcoin's implementation involves digital signatures and network timestamps, utilizing a hashing chain and hash-based Proof of Work (PoW). The interconnected blocks form an immutable distributed database, ensuring the integrity of records by requiring the redo of PoW for any changes. As a decentralized system, Bitcoin operates by having users request transactions and miners validate and commit these transactions to the ledger. This process involves solving PoW crypto puzzles, ensuring the construction of a new block approximately every ten minutes on a global scale. The Nakamoto (2008) consensus, as elucidated by Chu & Wang (2018, p. 2), plays a crucial role in achieving consensus among open pools of miners. Miners

are economically incentivized to append to the longest chain, and the consensus tolerates Byzantine faults as long as over 50% of computing power is controlled by honest nodes. This historical perspective underscores Bitcoin's foundational role in shaping decentralized digital currencies.

Ethereum, conceived as an open-source public Blockchain platform, stands out as a versatile tool for implementing decentralized applications, as highlighted by Bhutta et al. (2021, p. 12). Unlike Bitcoin, Ethereum goes beyond cryptocurrencies and empowers users to create complex applications using smart contracts. These smart contracts, small immutable programs stored in the Ethereum network, represent various resources, including currency, land, and houses. The Ethereum Virtual Machine (EVM) plays a central role in executing these decentralized applications, processing complex algorithms written in the contract-oriented programming language Solidity. Chatterjee & Chatterjee (2017, p. 2) Ethereum's distinctiveness lies in its digital contracts known as "smart contracts," which facilitate agreements between clients and end-users. Major multinational companies, including IBM and Amazon, have actively utilized Ethereum, foreseeing its continued integration in various sectors Introduced in 2015, Ethereum, as detailed by Polge et al. (2021, p. 2), is not only a public permissionless Blockchain but also supports private configurations. Initially relying on a PoW consensus protocol, Ethereum has evolved to explore alternatives like the Clique Proof-of-Authority (PoA) consensus protocol, addressing the security concerns and computational demands associated with PoW. Ethereum's adaptability is evident in its support for smart contracts and the EVM, allowing nodes to execute programs in any language. Ethereum's native cryptocurrency, Ether, further enhances its utility in various applications, emphasizing its multifaceted nature and pivotal role in shaping decentralized applications and Blockchain technology.

Hyperledger Fabric emerges as a pioneering permissioned distributed operating system introduced by IBM, according to Bhutta et al. (2021, p. 12). Distinguished by its programmable framework, Hyperledger Fabric allows users to execute distributed applications independently of native cryptocurrencies. The framework introduces the innovative execute-order-validate Blockchain architecture, addressing the limitations of the order-execute architecture and gaining significant traction within the Blockchain community. However, concerns highlighted by Chu & Wang (2018, p. 2) include the inability to use permissionless Blockchain and the limited number of programmers versed in Hyperledger Fabric application development. Operating under a modular umbrella approach, this project spans infrastructure development, framework implementation (e.g., Fabric, Iroha, Sawtooth, Burrow, Grid, and Indy), and tool support (e.g., Composer, Explorer, Caliper, Cello, Quilt, and Ursa). Hyperledger Fabric, distinct from fully open Blockchains, is designed to build permissioned Blockchains and support smart contracts in traditional programming languages. It lacks economic incentives and uses traditional consensus algorithms like Practical Byzantine Fault Tolerance (PBFT), making it particularly favored by closed consortiums, including financial institutions, according to Chu & Wang (2018, p. 2). Hosted by the Linux Foundation, Hyperledger Fabric supports the execution of distributed applications written in general-purpose programming languages like Go, Java, or Node.js. It relies on a membership service provider to manage node identities. Fabric's nodes include clients proposing

transactions, peers maintaining ledgers and states, and ordering service nodes establishing transaction order. Employing smart contracts, or chaincode, to implement application logic, Fabric natively uses Solo, a voting-based consensus protocol, while accommodating alternative protocols like PBFT, Raft, or Kafka. Notably, Fabric distinguishes itself by lacking an underlying cryptocurrency as Polge et al. (2021, pp. 1–2) highlighted.

2. Electronic Voting with Blockchain

Electronic Voting has been a subject of research over the past few decades. Compared to traditional paper-based voting, it offers several advantages, including environmental sustainability, real-time counting and processing, and reduced error rates. These benefits have the potential to enhance overall voter turnout (Liu & Wang, 2017, p. 1).

2.1. Electronic Voting

As argued in Montargil (2004) work, the adoption of Electronic Voting systems can be supported by various arguments. One commonly cited point is the potential reduction in the costs associated with organizing and managing elections. This includes the elimination of printed ballots and streamlined logistics, leading to lower overall expenses. Another argument emphasizes the ecological benefits, suggesting that Electronic Voting could contribute to environmental conservation by reducing the need for millions of paper ballots. However, the frequency of elections is noted as a crucial factor in evaluating this argument.

The acceleration of result tabulation is another argument, with the idea that Electronic Voting systems could significantly reduce the time required for processing election results. This claim, however, may be context-dependent, as some electoral systems already have quick manual counting processes. The reduction of errors in result tabulation forms a fourth argument. Advocates argue that electronic systems can minimize mistakes, particularly in converting votes to mandates. However, the relevance of this argument depends on the complexity of the existing electoral process.

Enhanced mobility is suggested as a fifth argument. The idea is that citizens could vote at any convenient location, irrespective of their registered residence, by leveraging Electronic Voting systems. This, however, requires a fully computerized electoral registration management process. Challenges associated with non-presential voting encompass both political and technical aspects. Political considerations include whether non-presential voting should be universally applicable, while technical challenges involve ensuring the correct presentation of ballots to remote voters.

Montargil (2004) also presented the concept of Electronic Voting, incorporating two additional concepts: poll-site Electronic Voting and remote Electronic Voting. Poll-site

Electronic Voting involves utilizing electronic tools for engaging in various forms of political participation within a given system. The defining aspect of this concept lies in the nature of the technology used, independent of its impact on citizens' mobility. Therefore, if a poll-site Electronic Voting experience employs electronic means for political engagement, it doesn't necessarily eliminate existing in-person constraints, such as those associated with in-person voting.

Poll-site Electronic Voting is instrumental and relative. It specifically relates to the utilization of electronic tools for existing forms of political participation, without necessarily altering those forms. Approaching a democratic regime in this manner characterizes electronic democracy, wherein citizens have the opportunity to engage in established forms of political participation using electronic means.

Remote Electronic Voting refers to the use of methods that enable citizens to engage in political participation from different locations within a given political system. The essential element in defining remote Electronic Voting is the freedom of movement for citizens. In political systems where face-to-face interactions are significant, remote Electronic Voting anticipates a decrease in face-to-face constraints and associated costs, regardless of the specific technology employed. Remote Electronic Voting often suggests the possibility of citizens participating in politics from their homes or without the requirement of physical presence in predefined spaces. In this context, political involvement becomes, at its extreme, independent of a specific location.

Similar to poll-site Electronic Voting, remote Electronic Voting is instrumental and relative, shaped within the context of a specific political system. Its limits and potential variations relate to how existing forms of political participation are executed, rather than altering the fundamental forms themselves. Teledemocracy, as an extension, occurs when democracy enables the exercise of existing forms of participation with mobility, without necessarily altering those forms.

However, it's essential to note that remote Electronic Voting isn't solely about instrumental changes or altering how the political system functions. The shift in the relationship between specific forms of political participation and the territorial contexts where they occur could bring about profound changes in the ritual dimension and social significance of political involvement. It could lead to the transformation of "public spaces" into a "virtual public space," redefining the connection between political participation and physical space. Furthermore, remote Electronic Voting inherently offers an advantage to citizens, as the possibility of participating remotely implies a reduction in participation costs. In contrast, electronic participation, involving the use of technology with specific features, does not inherently provide a similar advantage to citizens.

Table 2.1: Electronic Voting Examples. Source: elaborated by the author, using the mentioned references.

Country	Voting Method	Mandatory	How it works	Remarks	Problems	Origin
Estonia	Remote Electronic Voting	No	Voting via PC Application	Voters can change their vote during voting period	Election administration, Limited impact on Participation	(Ehin et al., 2022)
Norway	Remote Electronic Voting	No	Voting via PC browser	Return code of the party voted via SMS	SMS veracity, Malicious on user-end, false reports return codes	(Stenerud & Bull, 2011)
Switzerland	Mixed Electronic Voting	No	Voting via mail, online, in-person and SMS	Voters receive all official documents 3 weeks prior	Participation gap, Technological Concerns and Voting Quality	(Gerlach & Gasser, 2009)
Brazil	Poll-site Electronic Voting	Yes	Vote in-person on an electronic ballot	Candidate picture, no hand-written mistakes and accelerates vote tallying	Ballot privacy, recount is impossible, blind faith in the election authorities	(Aranha & van de Graaf, 2018)

There are several examples of countries that have embraced Electronic Voting systems, such as Estonia, the first in the world to adopt this technology for its national elections, as well as Switzerland, Norway and Brazil, as seen in Table 2.1.

Estonia, as argued by Ehin et al. (2022), renowned for its democratic practices, adopted an innovative Electronic Voting system in 2005. With a population of 1.3 million, the nation employs a competitive multi-party democracy, featuring elections for the 101-member Riigikogu every four years. The system, utilizing open-list proportional representation, enables voters to cast their ballots for candidates on a party list. Noteworthy features include a voting age of 18, extended to 16- and 17-year-olds in local elections since 2017. The Population Register ensures automatic voter registration, eliminating additional steps. Traditional paper voting occurs on Sundays, with early voting options and provisions for special voter needs and citizens abroad.

A pioneering aspect in Estonia's system is internet voting, available during a designated early voting period, allowing global voting from any internet-connected computer. The process, taking under two minutes on average, involves downloading a voting application, authentication, candidate selection, encryption, and digital signature confirmation. Crucially, internet voters can change their votes during the early voting period, prioritizing voting secrecy. Initially excluded from voting on Election Day, internet voters can cast a paper ballot since 2021, showcasing Estonia's commitment to accessible and secure electoral processes, ensuring democratic integrity.

The Norwegian Electronic Voting system, as argued by Stenerud & Bull (2011), provides a straightforward experience for voters. Using the widely used MinID two-factor

authentication, voters access a point-and-click interface displaying the ballot. After making selections, the voter's choices are sent to central voting servers through a Java applet on their client PC, ensuring encryption and digital signatures. Post-vote, the voter promptly receives a text message with a unique 4-digit return code, which corresponds to a party on the poll card sent by mail before the voting period. The return code serves as a verification tool, allowing voters to confirm that their vote matches the chosen party on the poll card. This mechanism, known as a cast-as-intended proof, ensures the accurate receipt of the vote by the server. To prevent coercion, voters can cast an unlimited number of internet ballots and even switch to paper voting if needed. The decision to send return codes via SMS, instead of displaying them on the screen, adds an extra layer of security, preventing attackers from learning the meaning of the codes and replacing votes without the voter's awareness. Checking the return code is optional, and the poll card is not used for authentication, allowing voters without the card to still participate but without the ability to verify the SMS return code.

In Switzerland, as argued by Gerlach & Gasser (2009), pilot experiments in the cantons of Geneva and Zurich were employed to determine whether Electronic Voting can be used for democratic purposes (Gerlach & Gasser, 2009). The Geneva Electronic Voting system operates with a mail-based distribution of official documents to voters three weeks before the vote date. Accompanied by a personal voting card, voters can choose to submit their ballots by mail, online, or at the ballot box, with the card ensuring the "one man, one vote" principle. The Electronic Voting procedure in Geneva involves four stages. The voter, using a secure server, verifies eligibility by entering an individual identification number found on the ballot sheet. Subsequently, they make their vote, confirm choices, and verify their identity through date of birth and a hidden PIN code on the ballot sheet. After confirmation, voters receive acknowledgment of their accepted and recorded vote. In Zurich, an enhanced Electronic Voting model includes internet-based voting, text message voting (discontinued in 2007), and interactive television system (ITV) voting. The Zurich system comprises several elements spread across different locations, maintained by various entities. Components include the pre-existing voting system, the Electronic Voting platform, data centers of communities, and user interfaces like PCs, cell phones, or interactive television systems, connected via a special public data network. Similar to Geneva, Zurich voters receive registration information containing a user-ID, hidden PIN code, browser certificate "fingerprint," and a security symbol. For internet voting, users enter their user-ID, verify the security symbol, input date of birth and the PIN received in the mail, and submit the ballot, receiving a confirmation message. Zurich's SMS-based voting entails sending an SMS with the user-ID, poll code, and vote code, followed by another SMS with the PIN and date of birth upon receiving a response. The Zurich system incorporates security measures, including encrypted communications, a firewall-protected server, encrypted storage of cast votes, and the use of Write Once Read Multiple (WORM) medium for added security. PINs are safeguarded by a security seal, and voter registration data is transient, generated and deleted as needed to maintain privacy.

Brazil, as argued by Aranha & van de Graaf (2018), one of the world's largest countries by area and population, faces unique challenges in organizing elections across its vast territory (Aranha & van de Graaf, 2018). The country transitioned from paper ballots to nationwide Electronic Voting, completing the shift in 2000. The Superior Electoral Tribunal (TSE) oversees election matters, managing the audacious Electronic Voting project. The Electronic

Voting process in Brazil involves multiple stages. Initially, during the voter identification phase, individuals present their identification documents to poll workers who input the details into the voting equipment. In certain instances, fingerprint identification adds an extra layer of authentication. Following this, the voter engages in candidate selection, as shown in Figure 2.1, by inputting the digits associated with their preferred candidate or opting for "BRANCO" to register an abstention vote. The subsequent step entails confirmation or correction, where the voting machine displays a photo of the chosen candidate or flags the vote as invalid. Voters can then press "CONFIRMA" to affirm their choice or "CORRIGE" to rectify any errors. This iterative process of selection and confirmation ensures thorough participation of voters in the electoral process.

Figure 2.1: Brazil Electronic Voting Machine (Bond, 2022)

The Electronic Voting system offers advantages, such as providing visual confirmation of the chosen candidate, reducing ambiguity, and accelerating the vote tallying process. The digital representation allows for faster computation of results, often determining the winner before midnight. While proponents highlight these advantages, the system's security is paramount. Ensuring the integrity of the election process involves addressing security concerns, particularly in remote areas reachable only by plane and boat. The logistics of the Electronic Voting system, known as “urna eletrônica,” demonstrate its resilience in diverse conditions. Understanding the concept of security in the election context involves defining and meeting specific criteria to ensure a fair and reliable electoral process.

2.2. How Blockchain can improve Electronic Voting

Electronic Voting systems have gained attention for their potential to revolutionize the democratic process, offering efficiency and accessibility. However, several drawbacks and security concerns have been extensively discussed by various authors, leading to a cautious approach in the global implementation of Electronic Voting systems.

As argued by Ben Ayed (2017), the centralization of Internet Voting systems makes them susceptible to Distributed Denial of Service (DDOS) attacks, potentially rendering elections inaccessible to voters. Security vulnerabilities, as discussed by Taş & Tanrıöver (2020) and Mursi et al. (2013), pose major challenges in Electronic Voting systems, introducing risks such as identity theft, malware, server penetration attacks, and spoofing. Specifically, centralized Electronic Voting systems are exposed to various cyber-attacks, compromising system usability, privacy, and authentication. The potential for manipulation and coercion in online voting systems, both in terms of vote count and electoral fraud, further diminishes their reliability (Taş & Tanrıöver, 2020). Coercion resistance, as argued by Bokslag & de Vries (2016), is particularly challenging in I-voting, where voters can be coerced into voting under observation.

Taş & Tanrıöver (2020) also argued that the lack of trustworthiness in current Electronic Voting systems stems from the absence of proofs or confirming evidence of honesty. They suggested that assuring voters of their intended votes requires the implementation of paper receipts, but this approach comes with its own set of challenges, including a heightened risk of bribery and coercion. According to Mursi et al. (2013), spoofing attacks add another layer of vulnerability to internet-based voting systems, where voters may unknowingly connect to imposter sites, leading to the potential undetected loss or manipulation of votes. Large-scale attacks are also facilitated in Electronic Voting systems due to the common use of the same software across a country, allowing a small group of attackers to potentially influence election outcomes (Bokslag & de Vries, 2016).

In addition to these challenges, Taş & Tanrıöver (2020) highlighted the difficulty in checking the accuracy of votes recorded in Electronic Voting results. The general lack of transparency in voting processes further contributes to the skepticism surrounding Electronic Voting. Major security risks identified in specific Electronic Voting applications, including unsafe security protocols and weak passwords, have led to the discontinuation of certain systems.

Electronic Voting has encountered significant challenges that have hampered its widespread adoption globally. However, the integration of Blockchain technology holds the potential to address these challenges and bring about a transformative shift in the landscape of Electronic Voting, both in on-site/poll-site and internet-based scenarios. Drawing insights from Tu et al. (2023), Blockchain brings meticulous verification of voter eligibility in on-site Electronic Voting. By utilizing digital signatures, facial recognition, and data from official entities, the technology ensures that only eligible voters with valid credentials participate, significantly reducing the risk of including ineligible voters. Moreover, Blockchain guarantees the validity of each cast ballot through digital signatures and smart contracts. This not only upholds the sanctity of the voting process but also securely records each ballot on the Blockchain, allowing for a reliable final tally. The decentralized nature of Blockchain enhances transparency in Electronic Voting. Consortium members and external entities can securely examine voting operations, fostering trust in the fairness and accuracy of the electoral system.

In the realm of internet-based Electronic Voting, as discussed by Indrason et al. (2021), Blockchain introduces decentralization, enabling voters to cast their votes from anywhere. This addresses concerns related to undue influence and enhances overall security and transparency. Blockchain's security features, including hash functions and secure public

ledgers, are crucial for maintaining the integrity of the voting process, preventing unauthorized access, and ensuring a robust and trustworthy Electronic Voting system.

Agbesi & Asante (2019) emphasize the revolutionary impact of Blockchain on election result recording. The decentralization aspect, facilitated by a distributed peer-to-peer network, aims to enhance trust throughout the election process. Transparency can be promoted through a public Blockchain, granting comprehensive access and oversight, allowing stakeholders to validate each step. Immutability can be achieved through digital signatures, ensuring that results on the Blockchain remain unalterable, contributing to the security of the election results.

Khan et al. (2020) highlight the integration of Blockchain and smart contracts in on-site Electronic Voting within physical polling booths. This addresses security concerns associated with Electronic Voting machines, providing an immutable and transparent system. The organized data structure, secured through hashing and distributed across a peer-to-peer network, acts as a defense against tampering. Smart contracts contribute to efficient control between parties and the management of tokenized assets within the Electronic Voting system.

A et al. (2018) position Blockchain as a comprehensive solution to challenges in the current voting process. Transparency is a key focus, allowing voters to track and verify their votes in real-time. The reliability of the voting process is improved through the creation of an immutable record of votes, mitigating risks associated with manual errors. Accessibility is enhanced by empowering people to verify information, reducing the potential for administrative interference.

2.3. Limitations Electronic Voting with Blockchain still presents

While Blockchain technology has introduced significant improvements to Electronic Voting systems, certain limitations and challenges persist, requiring careful consideration and ongoing development.

Ben Ayed (2017) argues that one key limitation arises from the assumption that voters will use secure devices to cast their votes. Despite the overall system security, the potential for hackers to exploit malicious software on voters' devices remains a concern, allowing unauthorized casting or alteration of votes. Additionally, the inability to change a vote in case of user mistakes poses another drawback, as users are restricted to casting their votes only once.

The implementation of Blockchain-based Electronic Voting systems, as highlighted by Taş & Tanrıöver (2020), faces specific challenges that can impact their advantages. Issues related to smart contracts and election rule violations need resolution to ensure the integrity of the process. The transparency-security trade-off arises when considering the accessibility of published data on public Blockchains, necessitating a careful balance, potentially through the use of private Blockchains. Time-dependent disclosure mechanisms for election results present additional challenges to prevent potential interventions based on intermediate results.

Kshetri & Voas (2018) argue that scalability emerges as a significant concern for Blockchain-based Electronic Voting systems. As the number of users increases, scalability problems

intensify, impacting the cost and time consumption of transactions, especially during large-scale elections. Parallelizing Blockchain through techniques like sharding is proposed as a solution to enhance scalability. However, the inherent scalability issues of Blockchain operations demand further improvement in terms of latency, throughput, and cost-effectiveness (Taş & Tanrıöver, 2020).

User identity protection remains a challenge, as discussed by Kshetri & Voas (2018), as utilizes pseudonyms, and user identities can potentially be revealed through the public nature of transactions. Ensuring transactional privacy in an election system, where anonymity is crucial, requires a delicate balance, possibly involving a third-party authority for privacy checks and balances. Energy efficiency in Blockchain processes, including consensus mechanisms, peer-to-peer communication, and encryption, presents another challenge. To make Blockchain-based Electronic Voting more sustainable, energy-efficient consensus methods and modifications to peer-to-peer protocols are suggested.

The immaturity of Blockchain technology is acknowledged as a limitation, leading to technical issues in adoption. Current implementation flaws and the lack of widespread understanding contribute to challenges in evaluating the full potential of Blockchain in Electronic Voting. Acceptance by the public and political leaders is crucial for the success of Blockchain-based Electronic Voting. Blockchain's complexity may hinder its acceptance among the general public, requiring extensive awareness campaigns to highlight its benefits. Additionally, political leaders, accustomed to the existing centralized election processes, may resist the shift towards decentralized Blockchain-based systems (Jafar et al., 2021).

2.4. Electronic Voting with Blockchain Applications

We will now explore various insights provided by different authors, as seen in the table, in the domain of Electronic Voting with Blockchain, both for Remote Electronic Voting and Poll-site Electronic Voting, shedding light on the diverse applications and innovative approaches within this field, as shown in Table 2.2 we can have an overview on different Electronic Voting with Blockchain Applications.

Kshetri and Voas (2018) provide insightful examples of Blockchain-based Electronic Voting applications in their research. For instance, Voatz conducted tests of its mobile-phone-based system during various events, including student government elections, church-group voting, and non-profit organization balloting. This demonstrates the system's versatility in informal and consultative voting scenarios.

Table 2.2: Electronic Voting with Blockchain Examples. Source: elaborated by the author, using the mentioned references.

Type	Applied to	How it works	Remarks	Origin
Remote	Various events	Mobile-phone-based system		(Kshetri & Voas, 2018)
Remote	South Korea community projects	Blockchain platform, smart contracts	Nine thousand residents voted, no central authority was involved	(Kshetri & Voas, 2018)
Remote	Corporate-governance-related decisions in Estonia	Log in using their verified national online ID	Voting system issues voting-right assets and voting token assets to shareholders	(Kshetri & Voas, 2018)
Poll-site	Proposed System	Voters present credentials and undergo verification	The system relies on Blockchain, biometrics, and cryptographic techniques	(Tu et al., 2023)
Poll-site	Proposed System for Africa and Ghana	Participants, including political parties act as full nodes	Emphasizes decentralization, transparency, and tokenization	(Agbesi & Asante, 2019)
Poll-site	Proposed System for Indonesia	Voting includes preparation, verification and real-time monitoring	Utilizes a Hyperledger Fabric with Admin Server, Polling Station, and KPPS roles	(Seftyanto et al., 2019)

In South Korea, the province of Gyeonggi-do pioneered the use of a Blockchain-based Electronic Voting system in March 2017 for the Ddabok Community Support Project. This marked a significant milestone as the first application of Blockchain technology for voting in the country. The system, developed by the Korean fintech startup Block, employed smart contracts and stored votes, results, and relevant data on the Blockchain, eliminating the need for central authority involvement.

Estonia, known for its forward-looking approach to technology, extended the use of Blockchain-based Electronic Voting to corporate governance. Shareholders of the Estonian technology company LVH Group, who are Estonian citizens or e-residents, utilize Blockchain-based Electronic Voting to make decisions at the annual general meeting. Estonia plans to integrate Blockchain in various sectors, including e-residency projects and healthcare.

Tu et al. (2023) proposed a Poll-site Electronic Voting system, where voters navigate through three key phases: Before Voting, During Voting, and After Voting. Before voting, the Central Election Commission (CEC) initiates the election, recording essential details on the Blockchain. The Household Registration Agency (HRA) provides voter data, verified by the CEC, and generates voting credentials. These credentials, accompanied by digital signatures, are delivered to voters by the Election Commission (EC). Voters can verify their credentials on the Blockchain using public keys. During voting, voters bring their credentials, digital signatures, and a QR code to the polling station for verification. Inside the voting booth, an

electronic machine facilitates candidate selection, ensuring secure recording of votes on the Blockchain. The system automatically tallies votes through a smart contract, addressing irregularities like spoiled ballots. Polling stations signal the end of voting on the Blockchain. After voting, the smart contract concludes the process. The EC reads the tally sheet from the Blockchain, enabling authorized institutions to inspect and query results. The EC officially announces the election outcome based on the tally sheet. This proposed Electronic Voting system integrates Blockchain, biometrics, and cryptographic methods, ensuring a secure, transparent, and efficient electoral process. It represents a shift toward a more advanced and trustworthy approach to conducting elections.

Agbesi & Asante (2019) proposed a Blockchain-based election result recording system for Africa and Ghana that leverages Ethereum's permissionless Blockchain architecture to ensure transparency, immutability, and decentralized control. The system involves political party nodes and Electoral Commission (EC) nodes, each serving as full nodes with an identical copy of the distributed Blockchain database. The design eliminates the need for a trusted third party, promoting trust in the collation process. Tokenization is introduced to represent vote counts, utilizing fungible tokens to ensure the uniformity and security of the data. These tokens are created at polling stations, representing the vote count, and transferred through collation points until they reach the EC main office. This approach prevents unauthorized modifications to the election results at intermediate collation points. The process flow involves interactions between full nodes (political party nodes, EC) and light nodes (polling stations). Each polling station, equipped with public and private keys, initiates transactions via a Blockchain API. The transaction includes the polling station ID, parliamentary and presidential results as tokens, and a verification code by party agents. The transaction is signed using the station's private key. Once the transaction is initiated, a new block is created, containing relevant data and signatures, and broadcast to all nodes in the network. EC and political party nodes validate the transaction by verifying signatures, confirmation codes, and computing hash values. Validated blocks are added to the Blockchain, ensuring the integrity of the recorded results. This process repeats as results move through collation points, with constituency centers receiving tokens and transferring them to subsequent collation centers until reaching the EC headquarters. The goal is to establish a secure, transparent, and tamper-resistant election result recording system, reducing the reliance on centralized authorities and enhancing overall trust in the electoral process.

Seftyanto et al.(2019) proposed a Poll-site Electronic Voting system for Indonesia that is built on a Hyperledger Fabric Blockchain framework with a decentralized architecture. The design employs a multiple organizations model, with each server serving as an organization. This approach ensures heightened security by creating a Blockchain duplicate for each server, making hacking attempts challenging and enhancing public trust. Within the electronic election system, each server operates with three distinct participant types. The Admin Server assumes a pivotal role, managing the provincial-level system with comprehensive privileges, including the creation, reading, updating, and deletion of participant types for both Polling Stations and KPPS (Officers). Additionally, Admins enjoy unrestricted access to all resources within the business network. Meanwhile, the Polling Station acts as a voting venue, accommodating an average of three voting booths. It holds the unique privilege of creating

ballots throughout the election process. Lastly, the KPPS (Officers) play a crucial role in monitoring incoming ballots, overseeing the election process, and tracking activities within Polling Stations and KPPS. Officers, constituting KPPS participants, possess read-only privileges. Notably, both KPPS and Polling Station participants can only be created by the Admin Server. The proposed Electronic Voting system introduces the concept of assets and transactions within the Hyperledger Blockchain. Ballots are considered assets and are designed with unique values, including Ballot ID, Information of Polling Station (Officer ID, Village ID, District ID, and Regency ID), and Candidate ID. Transactions in the election process involve creating new ballots, a task exclusively performed by participants in the Voting Booth. To define the business network, the Hyperledger Fabric Framework incorporates a business network archive that collectively governs the interactions between participants, assets, and transactions.

The electronic election procedure unfolds through distinct stages at a polling station, ensuring a systematic and secure voting process. In the initial stage, registered voters present themselves with an identity card and invitation during the Voter Preparation Stage. Subsequently, the Voter Verification Stage employs a two-part verification process, utilizing electronic identity cards and fingerprints. This comprehensive procedure aims to guarantee the authenticity of voters and ascertain their absence from the polling station. Moving to the Voting Stage, KPPS officers facilitate the process by issuing smartcards to voters, who then proceed to make their selections within the booth. Following the vote, they receive audit receipts containing the unique Ballot ID. In the concluding Voice Monitoring Stage, the transparency of the election process is enhanced. Votes become accessible online, and the real-time counting process can be observed through KPPS Officer accounts or on the General Election Commission website, providing a comprehensive overview of the election proceedings.

2.5. Blockchain Frameworks

In this chapter, we will explore different Blockchains and their applications to Electronic Voting. Many existing Blockchains can already be used as solutions, as seen in Table 2.3, Ethereum, Hyperledger Fabric, Corda, Tezos, and Avalanche.

Ethereum as stated by Singh et al. (2023), is a public Blockchain well-suited for a transparent voting system. It supports smart contracts and has a high adoption rate and developer community. Ethereum currently uses a Proof-of-Work consensus mechanism but is evolving to Ethereum 2.0, which will use Proof of Authority as its consensus mechanism. Hyperledger Fabric as stated by Díaz-Santiso and Fraga-Lamas (2021), is a private Blockchain suitable for enterprise or government-led Electronic Voting systems. Developed by the Linux Foundation, it offers a modular architecture with pluggable consensus. For smart contracts, it supports chaincode, which can be upgraded and modified over time.

Table 2.3: Blockchain Frameworks for Electronic Voting. Source: elaborated by the author, using the mentioned references.

Name	Type	Developers	Remark	Source
Ethereum	Public	non-profit organization	High adoption and developer community	(Ethereum, 2024)
Hyperledger Fabric	Private	Linux Foundation	Modular architecture with pluggable consensus	(*Hyperledger Fabric*, 2024)
Corda	Private	R3	Emphasizes privacy and scalability	(Corda, 2024)
Tezos	Public	Arthur Breitman and Kathleen Breitman	Focus on security and upgradability	(Tezos, 2024)
Avalanche	Both	Researchers at Cornell University	Highly scalable with sub-second transaction finality	(Avalanche, 2024)

Corda is a private Blockchain suitable for enterprise-level Electronic Voting systems that require privacy and scalability. Developed by R3, it supports interaction with other systems and has a pluggable consensus mechanism. Tezos is a public Blockchain focused on security and upgradability, making it suitable for transparent and secure voting systems. It is a self-amending Blockchain with on-chain governance and utilizes the Liquid Proof-of-Stake (LPoS) consensus mechanism. Avalanche can be both a public and private Blockchain with high scalability, suitable for Electronic Voting systems with sub-second transaction finality. It supports custom Blockchains (subnets) and interoperability, utilizing the Avalanche consensus mechanism.

2.6. Cryptography in Electronic Voting with Blockchain

In this chapter, we'll explore four key cryptographic methods used in Electronic Voting with Blockchain systems: Blind Signature, Elliptic Curve Cryptography, Short-Linkable Ring Signature, and RSA Cryptography with Optimal Asymmetric Encryption Padding (OAEP). Cryptography plays a crucial role in Electronic Voting, ensuring that votes remain anonymous and secure. By understanding these cryptographic techniques, we can better appreciate how they protect the integrity of the voting process, giving voters confidence in the security of their votes.

As argued by Carcia et al. (2021), a digital signature serves as a mathematical protocol to ensure the authenticity and integrity of a message. However, in the realm of blind signatures, the process diverges. Here, the originator initiates a blind displacement of the message, concealing its content before passing it on to the signing entity. The signing entity then signs the blinded message and returns it to the originator, who subsequently unveils the blind-signed message to reveal the signing entity's signature.

Blind signature protocols support two fundamental security properties: blindness and forgery resistance. Blindness prevents a malicious signer from discerning the order in which two

messages were signed, ensuring privacy. Meanwhile, forgery resistance stops any attempts by an attacker to create more signed messages than their interactions with the honest signer would permit. In the context of Electronic Voting systems, blind signature protocols serve to remove the link between voters and their respective ballots. Voters can obtain a blind signature on their message from the Election Authority, affirming their eligibility without divulging personal information or vote choices. This signed message allows voters to prove eligibility to the entity responsible for storing the ballots. Although the authority's signature in the message can be verified, the timing of the signature and the sender's identity remain hidden. Thus, the implementation of blind signatures ensures the privacy and anonymity crucial for an Electronic Voting system.

As argued by Ahmad et al. (2009), Elliptic Curve Cryptography (ECC) is a type of public key cryptography that offers several advantages over traditional methods like RSA. One significant advantage is the smaller key size required for an equivalent level of security. For instance, ECC keys of 224 bits and 256 bits are comparable in security to RSA keys of 2048 bits and 3072 bits respectively. This characteristic makes ECC particularly suitable for use in constrained devices, where resources such as memory and processing power are limited. ECC operates within two finite fields: the prime number field (GF(p)) and the extended binary field (GF(2m)). Among these, the prime number field is more commonly used in software implementations due to its smaller key sizes for equivalent security levels. An elliptic curve in the prime number field can be represented as a set of points satisfying a specific equation. This equation, along with domain parameters such as the prime number, coefficients, generator, point order, and cofactor, defines the curve's properties.

In ECC, operations such as point addition and scalar multiplication play crucial roles. Point addition involves adding two points on the curve, resulting in a third point. Scalar multiplication, on the other hand, involves multiplying a point by an integer, effectively adding the point to itself multiple times. As scalar multiplication typically consumes a significant portion of ECC execution time, the efficiency of ECC implementations depends largely on optimizing this operation. The strength of ECC lies in the difficulty of finding the scalar given the public key and base point. The private key, which is a random number within a certain range, is used to derive the public key through scalar multiplication with the generator point. Additionally, shared secret keys between parties can be computed by multiplying one party's private key with the other party's public key or vice versa.

As argued by Tornos et al. (2015), Short linkable ring signatures are cryptographic techniques used to ensure anonymity within a group while still enabling authentication. Initially, methods like shared keys were employed, granting access to authorized users but lacking robustness as anyone could share the key with unauthorized individuals. Group signature schemes addressed this by allowing only group members' signatures to be valid without revealing their identities, although a group manager was necessary for key issuance and potential signer identification.

Ring signatures were a significant advancement, removing the need for a manager and enabling users to sign as part of a group while preserving anonymity. Linkable ring signatures took this further, allowing different signatures from the same user to be linked without

disclosing their identity. However, a limitation of traditional linkable ring signatures was their linear increase in length with the number of voters in the ring. The challenge of increasing signature length with the number of voters in the ring was tackled by short linkable ring signatures. These signatures exhibit several key properties. First, they ensure unforgeability, meaning that an attacker cannot produce a valid signature without possessing the correct pair of public and private keys. Second, they provide L-anonymity, making it impossible for an attacker to identify which group member signed a specific message. Lastly, they offer linkability, enabling the determination of whether two signatures originate from the same signer.

RSA (Rivest-Shamir-Adleman), introduced by Rivest et al. (1978), is a widely used asymmetric cryptosystem known for its application in secure data transmission. RSA utilizes two keys, a public key for encryption and a private key for decryption. The security of RSA is based on the difficulty of factoring large integers into their prime components. Despite its widespread use, textbook RSA has vulnerabilities, particularly its deterministic nature, making it susceptible to chosen plaintext attacks where an attacker can deduce the plaintext from the ciphertext by comparing it to known encrypted texts, as argued by Zhong (2022).

To address these vulnerabilities, RSA is often combined with Optimal Asymmetric Encryption Padding (OAEP), a padding scheme that adds random padding to the plaintext before encryption, converting RSA from a deterministic scheme to a probabilistic one. This padding prevents attackers from inferring any structure from the plaintext and provides semantic security against chosen ciphertext attacks. OAEP was introduced by Bellare and Rogaway (1995) and has since become a standard in enhancing RSA's security. The combination of RSA with OAEP ensures that even if an attacker captures the ciphertext, they cannot deduce the original plaintext without the private key, thus enhancing the security of Electronic Voting systems by safeguarding the integrity and confidentiality of votes (Zhong, 2022).

2.7. Electronic Voting in Portugal

Portugal has been exploring Electronic Voting for nearly 30 years. Although it is not yet the main voting method, recent years have seen numerous tests in various elections to facilitate the transition to this modern voting system.

According to Montargil (2004) the first Electronic Voting experiment was made in October 1997, by Omron Electronics Portugal, a company specialized in industrial automation, developed the Omron Electronic Voting System (SVE). This system was designed to facilitate Electronic Voting using advanced technology, thus enhancing the voting process's efficiency and security. The initial project presentations were made to various entities, including the Technical Secretariat for the Electoral Process (STAPE), the Social Democratic Party (PSD), and the Calouste Gulbenkian Foundation.

The SVE comprises three fundamental components. The first component is a chip card, which is similar to those used in Multibanco or Telecom Card systems. The second

component is the Electronic Voting booth, which is equipped with a touchscreen terminal and a card reader/writer. The third component is the electronic ballot box, used for securely storing the votes.

The Electronic Voting process using the SVE involves several steps. The procedure starts with the voter receiving a chip card at the voting table after being identified. The voter then inserts the card into the reader/writer at the voting booth. If the card is inserted incorrectly, the system alerts the voter. The touchscreen terminal, as shown in Figure 2.2, displays the electronic ballot, and the voter selects their option by touching the screen. A cross appears in the corresponding square, and options for blank and null votes require explicit selection.

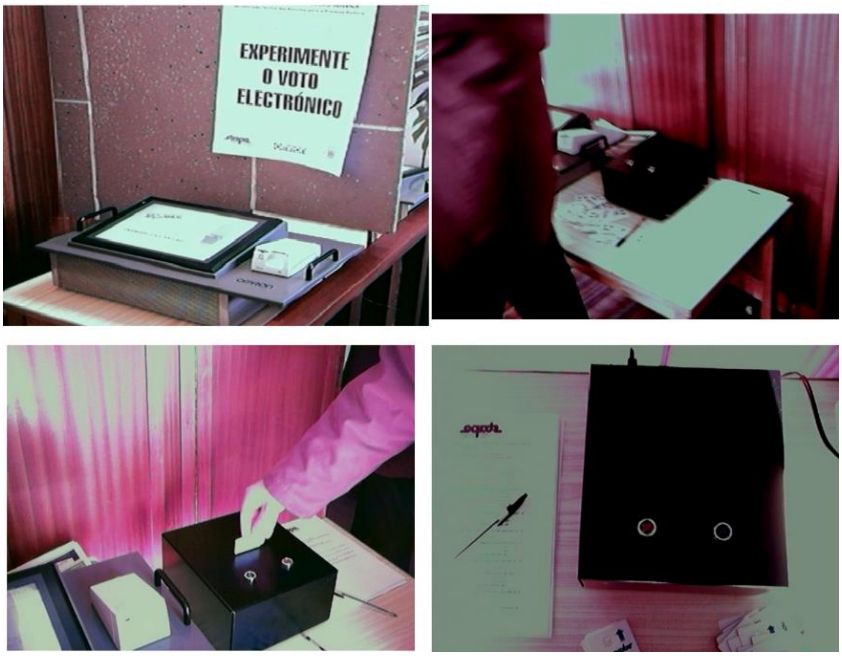

Figure 2.2: Voting Machines 2001 (STAPE, 2001)

The voter has a set time, defaulting to 5 seconds, to confirm their choice. If they change their mind, they can alter their selection within the allotted time. Once the selection period is over, the vote is recorded on the chip card. If there are multiple votes, the system cycles through the steps for each vote. The voter then returns to the voting table to insert the card into the electronic ballot box, which confirms the vote with a green light. If there's an error indicated by a red light, the voter must reinsert the card. After the vote is successfully recorded, the card is wiped clean and can be used by another voter.

The SVE allows access to two types of information. During the election, voting turnout can be monitored using a special control card. After the election, final results can be accessed using an end-of-vote card. The data can also be transmitted via fixed or mobile networks. It was tested in several pilot projects, beginning with the 1997 local elections in S. Sebastião da Pedreira, Lisbon, where it was implemented with voluntary participation after traditional voting. In 1998, the SVE was exclusively used in the XX National Congress of PSD for voting on strategic motions and the election of national organs. Another test occurred during

the 2001 local elections in the parishes of Sobral de Monte Agraço and Campelo, following a similar procedure to the 1997 local elections.

The pilot tests demonstrated the system's capability to handle Electronic Voting processes efficiently. However, issues such as system delays and initial configuration problems were observed. For instance, in the PSD Congress, the omission of Marcelo Rebelo de Sousa's motion from the initial results display caused suspicion and controversy.

In conclusion, the Omron Electronic Voting System (SVE) represents a significant advancement in electoral processes, providing a foundation for secure and efficient Electronic Voting. The pilot tests conducted across various elections and political events in Portugal have highlighted both the potential benefits and challenges of implementing such a system on a broader scale.

According to Montargil (2004) an Electronic Voting experiment was conducted during the European Parliament elections on June 13, 2004. This experience was organized by the Unidade de Missão Inovação e Conhecimento (UMIC) in partnership with STAPE and under the supervision of the Comissão Nacional de Eleições (CNE). Similar to the experiences conducted in the local elections of 1997 and 2001, Electronic Voting was carried out in-person at polling stations after voters exercised their right to vote traditionally. This participation was invitational and non-binding, meaning the results of the Electronic Voting experiment were not considered in the official electoral process.

Three IT entities were invited to participate in this experiment: Indra, a Spanish company; the Multicert/PT Inovação consortium; and Unisys, in partnership with PT Corporate and supported by Elections Systems and Software (ES&S). The experiment was carried out in nine parishes, with each company implementing its system in three different parishes. The parishes included Mangualde (Viseu), Mirandela (Bragança), Paranhos (Porto), S. Bernardo (Aveiro), S. Sebastião (Setúbal), Salir (Faro), Salvador (Beja), Santa Maria de Belém (Lisboa), and Sé (Portalegre). Indra used its Point-&Vote system in Mangualde, Santa Maria de Belém, and Sé; Unisys used the iVotronic system in São Sebastião, Salvador, and Salir; and Multicert used the voto@PT prototype in Mirandela, Paranhos, and São Bernardo.

The Electronic Voting experiment was monitored and audited by several universities, including the University of Minho, University of Porto, University of Lisbon, University of Aveiro, and the Technical University of Lisbon. Each university produced reports on their findings, which were made available on the UMIC website. From the voter's perspective, the systems operated similarly to previous Electronic Voting experiences: voters received a card at the voting table, moved to a voting booth to cast their vote via a touchscreen terminal, and then returned to the voting table to submit their card into an electronic ballot box. The vote was then downloaded from the card, and the voter received confirmation that their vote was successfully recorded.

A notable difference from previous experiments was the use of an electronic voter roll system in all parishes, developed by Multicert, which managed the voter rolls electronically, unlike previous instances where paper rolls were used. This electronic management is crucial for enabling voter mobility in future experiments.

Operational challenges were identified, including issues with system reliability and usability. Some voting systems involved lengthy voting processes, and there were software issues in some parishes that caused delays at the start of the experiment. Despite these issues, the general operational process was similar across systems, with some differences in user interfaces noted, such as Indra’s interface being more akin to traditional ballot presentations than Unisys's. Statistical data from the experiment showed varying voter turnout and participation rates in Electronic Voting across different parishes. For instance, Paranhos had the lowest Electronic Voting participation rate at 5%, while no incidents or difficulties were reported in the audit reports for this parish.

The audit reports also highlighted different interpretations of electoral concepts like vote uniqueness and voter anonymity. A trend was noted in considering null votes as a result of involuntary voter actions, potentially distorting their political significance. One report emphasized that an advantage of the system was to "avoid null votes," which might oversimplify the intentional political expression null votes can represent. This is crucial to consider in the design and testing of Electronic Voting systems to ensure they capture the full range of voter intentions accurately.

In conclusion, while the 2004 Electronic Voting experiment demonstrated significant progress and potential in using Electronic Voting systems, it also underscored the importance of addressing reliability, usability, and accurate representation of voter intentions in future developments.

The implementation of Electronic Voting in the district of Évora during the European Parliament elections on May 26, 2019, marked a significant step towards modernizing electoral processes in Portugal. The Ministry of Internal Administration spearheaded the development of a pioneering Electronic Voting system that adhered to fundamental electoral principles, including confidentiality, reliability, and the personal and in-person exercise of voting rights.

The Electronic Voting system in Évora was designed to ensure that all voters in the district were included, maintaining the confidentiality of each vote and ensuring that each voter could only vote once. The system was built to provide exact results that accurately reflected the voters' intentions and to ensure the personal, in-person exercise of voting rights as required by the Portuguese Constitution.

The pilot project involved the installation of 47 Electronic Voting tables across 25 parishes in 14 municipalities within the Évora district. The system integrated Electronic Voting with dematerialized electoral rolls, allowing voters to vote electronically at any of the 47 tables or traditionally at their registered section. The system adopted the Voter-Verifiable Paper Audit Trail (VVPAT) model, which issued a paper receipt similar to traditional ballots. Voters would fold the receipt and hand it to the table president for insertion into a ballot box.

On election day, the procedures began with the verification and autotest of all components before opening the electronic ballot boxes using a smartcard and PIN. Voters received a smartcard to activate their voting session, selected their vote on a touchscreen, and confirmed it, resulting in a printed receipt. The receipt was folded and handed to the table president. Special provisions were made for visually impaired voters through audio instructions.

Figure 2.3: Voting Process, Évora 2019 (SGMAI, 2019)

The pilot project saw a participation rate of 33.29% of voters in Évora using Electronic Voting. Despite some equipment failures and brief power outages, these issues were promptly resolved. Notably, Electronic Voting sections had significantly more voters per section compared to traditional sections.

According to Ordem dos Advogados (OA) (2022), it was implemented Electronic Voting for their 2023-2025 elections, including both the OA and the Caixa de Previdência dos Advogados e Solicitadores (CPAS). The system enables online voting with secure procedures for obtaining and recovering voting credentials. The integrity of the election process is ensured through audits. Updated contact information is essential for credential recovery, and support measures are available during election days to assist voters. This initiative aims to modernize the election process and enhance accessibility for members.

The three major sports clubs in Portugal—Sporting Clube de Portugal (Sporting), Futebol Clube do Porto (Porto), and Sport Lisboa e Benfica (Benfica)—are already adopting this new voting technology. In their most recent elections for governing bodies, Benfica, according to *Regulamento Eleitoral do Sport Lisboa e Benfica* (2021), implemented Internet Electronic Voting for members residing in the Azores, Madeira and abroad. Members only need to request a specific code for Electronic Voting from the General Assembly Board five days before the election. After authentication, members can cast their votes, including the option to vote blank. Upon registration, the member is prohibited from voting on-site. After submission, the member will receive a confirmation message indicating whether their vote was successful or not.

In November 2023, according Jornal de Notícias (2023), FC Porto approved the use of Electronic Voting in its elections during a general assembly. However, it was noted that they might not implement it in the next elections, which had not yet occurred at the time of this research. In October 2023, according to Renascença (2023), during a general assembly, Sporting tried to approve a proposal to allow Electronic Voting for members at a distance. Currently, their voting method is only available in Lisbon at Estádio José Alvalade. The proposal received nearly 70% support but needed 75% to be approved. The club president stated they intend to revisit this topic sooner or later.

Many tests of Electronic Voting have been conducted in Portugal over the past years, but it has not yet become the primary method. To this day, there is no record of Electronic Voting tests using Blockchain technology in Portugal.

3. Electronic Voting with Blockchain system

In this chapter we will focus on developing and implementing an Electronic Voting with Blockchain system, trying to follow the Portuguese laws on voting, our objective is to start to make a smooth transition from a paper-based voting system to an Electronic Voting System.

3.1. Voting Laws in Portugal

In Portugal from Legislação Eleitoral | Comissão Nacional de Eleições (2024), we can join the more important requirements for voting in Portugal and the ones that can be transformed in Electronic Voting with varying levels of importance. We have elaborated Table 3.1 to consolidate and present this information clearly. Each parish corresponds to a voting assembly, which may be divided into voting sections if it has more than 1,000 voters. In each assembly or section, there is a delegate and a respective substitute from each party running in the elections. Additionally, there are tables responsible for overseeing and facilitating electoral operations, each composed of a president, a substitute president, a secretary, and two tellers.

Table 3.1: Voting requirements in Portugal. Source: elaborated by the author, based on the voting requirements in Portugal.

Voting Requirements in Portugal	**Notes**	**Value**
Unicity	Each voter can only vote once	1
Secrecy	Votes must be secret and without a timestamp	1
Urn Closure	The urn must remain closed until the end of the election to prevent access to results	1
Display of Empty Urn	The empty urn must be displayed half an hour before the start of the election to members of the table and presented to voters	2
Post-Urn Voting	Voters arriving before the end time of the election must be allowed to vote	3
Consult at any time the copies of the electoral rolls	Any member from the table or party delegates can consult	2
Option of Blank vote		1
Option of Null vote		1
Electoral rolls with only the voters from that table	Voters have only one table where they can vote	3
Identification of the voter through an official document	Citizen card, driving license, passport, documents stay on the table until the vote ends.	1
An exemption, voters can vote without the official document	Two voters can confirm their identity under oath of honor, or table agreement	2
The table president says out loud the voter’s name and civil identification number.		4
Capability to vote in Braille	Braille matrix to be used over the ballot paper	4
Record to register all electoral operations		4

During the election process, table members and party delegates have priority to vote, provided they are registered in that assembly or section. Party delegates have the authority to replace any election operation, while members must address any doubts or queries from the voting assembly, whether during the voting or clearance phases. Decisions are made by an absolute majority of present table members, with the president holding the tiebreaker vote. In cases where a voter is affected by a notable illness or physical disability, rendering them unable to vote independently, they may be accompanied by another voter of their choice. However, if the table deems the voter capable of voting independently, a medical certificate must be presented at the time of voting.

After the election, the urn is only opened after the votes on the electoral roll have been counted. Any discrepancies between the number of voters counted in the election rolls and the ballot papers counted in the urn are resolved by prioritizing the latter for counting purpose Our system prioritizes key aspects of the traditional electoral process while transitioning to Electronic Voting. We ensure that each voter can cast only one ballot, preserving the principle of unicity, and guaranteeing the secrecy of each vote. Access to the urn's results is restricted until the conclusion of the election to maintain integrity.

We incorporate options for Blank and Null votes within the electronic interface. Voter identification relies solely on official documents, eliminating the need for manual verification processes. Table members and party delegates have real-time access to electoral rolls, which contain only the relevant voters assigned to their respective tables. Voters arriving before the designated end time can still cast their ballots, facilitating participation.

Our system maintains consistency with the traditional process while omitting complexities such as voting without official documents, braille voting, and manual registration, ensuring a smooth transition to Electronic Voting.

3.2. Hyperledger Fabric

To meet our requirements, we will use Hyperledger Fabric as we can explore in Hyperledger Fabric Docs (2023), a modular Blockchain platform offering highly confidential, resilient, flexible, and scalable distributed ledger solutions. Launched by the Linux Foundation's Hyperledger project in 2015, it fosters Blockchain technology development through a collaborative approach. Key features include a private and permissioned network, requiring participants to enroll through a trusted Membership Service Provider (MSP). Its modular architecture offers pluggable components, including consensus mechanisms and MSPs. Fabric supports the creation of channels, allowing subsets of participants to maintain separate ledgers for increased privacy.

The ledger structure consists of two components: the world state (a database representing the ledger's current state) and the transaction log (recording all transactions that have modified the world state). By default, the world state uses LevelDB, a key-value store database. Smart contracts in Fabric, known as chaincode, are written in languages like Go, Node.js, and Java. They are primarily used to interact with the world state. Fabric addresses privacy through channels and allows various consensus mechanisms to be chosen based on network requirements, ranging from structured to peer-to-peer relationships.

Hyperledger Fabric is designed to be a comprehensive enterprise Blockchain solution, featuring several innovative elements. Assets within Fabric can represent a range of items with monetary value, from tangible to intangible, enabling diverse exchanges over the network. Chaincode, which defines and manipulates these assets, is executed separately from transaction ordering, enhancing scalability and performance. The ledger, comprising an immutable transaction log and a world state database, ensures a tamper-resistant record of all transactions and supports SQL-like queries for efficient auditing.

Privacy is a cornerstone of Fabric, achieved through channels and private data collections. Channels allow specific participants to maintain separate ledgers, while private data collections enable confidential data sharing among subsets of organizations within a channel. Additionally, encryption of chaincode data further secures transaction information. Security is reinforced through permissioned membership, where all participants have known identities verified by cryptographic certificates. This structure facilitates precise data access control and ensures that transactions can be traced by authorized regulators and auditors.

Consensus in Hyperledger Fabric extends beyond transaction ordering, encompassing the entire transaction lifecycle from proposal to commitment. This process includes multiple verification steps, such as endorsement policies and system chaincode, to ensure transaction integrity and prevent double-spending. The system's modular architecture supports various consensus mechanisms, catering to different network requirements and enhancing overall flexibility and scalability. These features collectively establish Hyperledger Fabric as a robust platform for deploying secure, scalable Blockchain solutions tailored to enterprise needs.

Hyperledger Fabric networks are structured to provide ledger and smart contract services to applications, enabling the generation and distribution of transactions across peer nodes where they are immutably recorded. These networks often involve multiple organizations forming a channel with transactions governed by agreed-upon policies. In Fabric, “network” and “channel” refer to the organizations, components, policies, and processes within a defined structure.

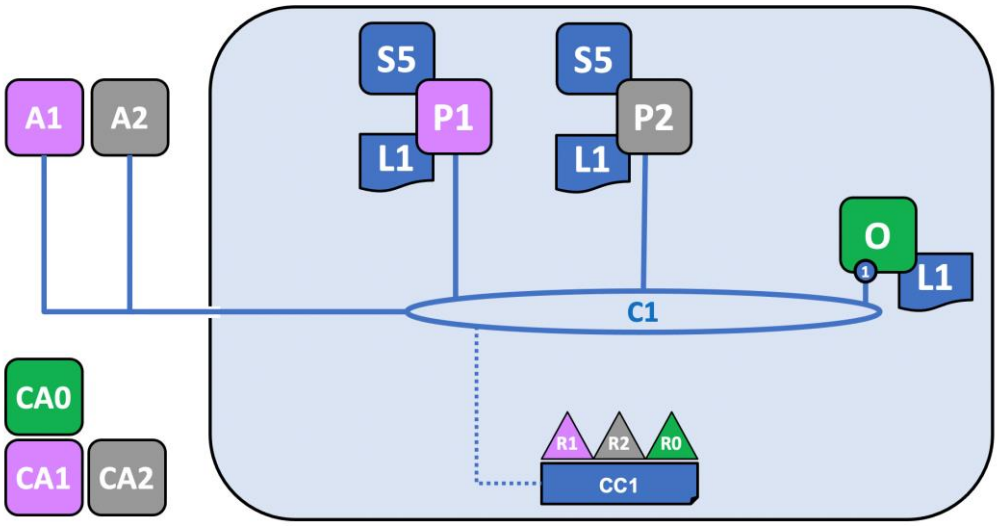

Figure 3.1: Hyperledger Fabric Framework

As seen in Figure 3.1, organizations R1, R2, and R0 collaboratively establish a network with configuration CC1, defining organizational roles and policies. R1 and R2 join peers P1 and P2 to channel C1, while R0 operates the ordering service O. All nodes maintain a copy of the ledger L1, where transactions are recorded. Applications A1 and A2 owned by R1 and R2 interact with the channel.

Creating a network or channel begins with agreeing on a configuration, resulting in a configuration block created by the configtxgen tool from a configtx.yaml file. This block records organizational details and policies governing the network. Certificates issued by Certificate Authorities (CAs) identify components and sign transactions, ensuring secure interactions within the network. Peers and ordering nodes join the channel, hosting ledgers and chaincode that define business logic for transactions.

Chaincode are installed on peers, approved by relevant organizations, and committed on the channel, with endorsement policies specifying which organizations must endorse

transactions. Client applications invoke transactions via the Fabric Gateway service, managing transaction proposals and endorsements, ensuring transactions are securely recorded on the ledger.

Certificate Authorities play a crucial role in the network by dispensing X.509 certificates used to identify components as part of an organization and sign transactions to indicate endorsement, necessary for ledger acceptance. Different components use these certificates to identify themselves within the network, with each organization typically having its own CA. Hyperledger Fabric includes a built-in Fabric-CA to simplify the setup process, though organizations often use their own CA. The Membership Services Provider (MSP) structure maps certificates to member organizations, enabling policies to assign rights and permissions within the channel. Certificates are fundamental to the transaction generation and validation process, as X.509 certificates are used to digitally sign transactions, which are then verified by network nodes before being accepted onto the ledger.

Joining nodes to the channel involves peers, which host ledgers and chaincode, and ordering services, which gather endorsed transactions and order them into blocks for distribution. R1, R2, and R0, listed in the channel configuration, join their peers (P1 and P2) and ordering service (O) to the channel, storing copies of the ledger L1 updated with each new block. It's essential for R1 and R2 to make P1 and P2 anchor peers to facilitate network communication. Once the ordering service joins the channel, updates to the channel configuration can be proposed and committed.

Installing, approving, and committing a chaincode involves deploying the business logic that defines peer interactions with the ledger. Chaincode is installed on peers, approved by relevant organizations, and committed on the channel, with endorsement policies specifying which organizations must endorse transactions. This process, known as the chaincode lifecycle, ensures that only endorsed transactions are accepted onto the ledger. Client applications invoke transactions via the Fabric Gateway service, completing the network structure by accessing the ledger L1 through smart contract S5, ensuring that endorsed transactions are written to the ledger.

Identity

In a Blockchain network, various actors including peers, orderers, client applications, and administrators are identified through digital identities encapsulated in X.509 digital certificates. These identities are crucial as they define permissions over resources and access to information within the network. Digital identities in Hyperledger Fabric are associated with additional attributes to form a principal, which determines the permissions of an actor based on properties like organization, role, and specific identity.

To ensure an identity is verifiable, it must be issued by a trusted authority known as a Membership Service Provider (MSP). MSPs use X.509 certificates as identities, adopting a traditional Public Key Infrastructure (PKI) model. PKI serves as a card provider, dispensing verifiable identities, while an MSP acts like a list of accepted card providers, turning these identities into trusted network members.

A Public Key Infrastructure (PKI), as represented in Figure 3.2, comprises technologies ensuring secure communications in a network, such as HTTPS. PKIs include Certificate

Authorities (CAs) that issue digital certificates, which authenticate parties in a network. A CA's Certificate Revocation List (CRL) lists invalid certificates, which is crucial for maintaining network security. A Blockchain network relies on PKI standards for secure communication and message authentication among participants. Key PKI elements include digital certificates, public and private keys, CAs, and CRLs. Digital certificates, compliant with the X.509 standard, authenticate identities and ensure message integrity through digital signatures, which use cryptographically connected public and private keys.

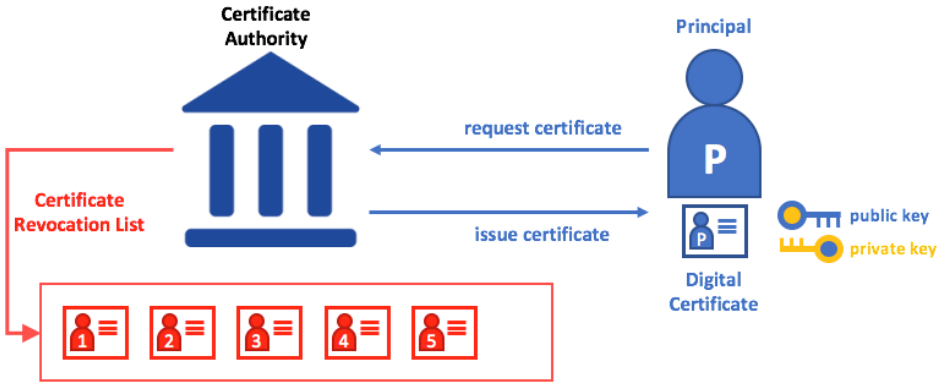

Figure 3.2: What is a PKI?

Certificate Authorities, as illustrated in Figure 3.3, issue digital certificates, binding an actor's identity to their public key. Trusted CAs validate these identities, ensuring secure network interactions. Certificates are disseminated without private keys, serving as trust anchors for message authentication. In a Blockchain setting, CAs define an organization's members digitally, providing the basis for verifiable digital identities. CAs are categorized into Root CAs and Intermediate CAs. Root CAs distribute certificates via Intermediate CAs, establishing a chain of trust. This structure scales certificate issuance securely and limits Root CA exposure. Intermediate CAs enable flexible certificate issuance across multiple organizations, essential for a permissioned Blockchain system like Fabric.

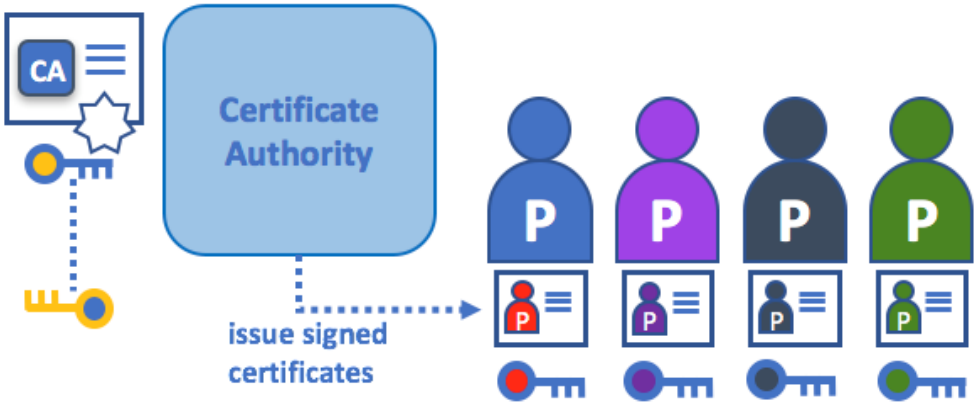

Figure 3.3: Certificate Authorities

Hyperledger Fabric includes a built-in CA component, Fabric CA, which manages digital identities in the form of X.509 certificates. Although Fabric CA is primarily for managing Fabric participants' identities, it is not suitable for providing SSL (Secure Sockets Layer) certificates for browsers. Fabric CA can be used in test environments or alongside public/commercial CAs for identity management. A Certificate Revocation List (CRL) references certificates that are no longer valid. When verifying an identity, checking the issuing CA's CRL ensures the certificate has not been revoked, maintaining network security against compromised identities.

Membership Service Provider (MSP)

Hyperledger Fabric is a permissioned network, requiring participants to verify their identity to interact on the Blockchain. The Membership Service Provider (MSP) plays a critical role in this process. Leveraging Public Key Infrastructure (PKI), MSPs provide verifiable identities through a chain of trust. Certificate Authorities (CAs) issue key-pairs used for identity verification. The MSP validates these identities, ensuring the network recognizes and trusts them. To transact on a Fabric network, a participant must have an identity issued by a trusted CA, be recognized by the organization's MSP, and ensure the MSP is included in network policies. An MSP not only lists permissioned identities but also assigns roles and privileges, turning identities into specific roles like admin, peer, client, orderer, or member.

MSPs operate in two domains: locally on nodes (local MSP) and within channel configurations (channel MSP). Local MSPs set node-level permissions and authenticate member messages, while channel MSPs define administrative and participatory rights at the channel level, ensuring transactions from authorized identities are accepted. Local MSPs are defined for clients and nodes, setting permissions for node operations and user interactions. Each node must have a local MSP to authenticate members and define administrative rights, ensuring a consistent root of trust across nodes, organization admins, and the nodes themselves. Channel MSPs manage rights at the channel level, defining relationships between channel member identities and policy enforcement. Each organization participating in a channel must have a channel MSP, ensuring only authorized members perform administrative tasks. Channel MSPs also facilitate privacy by segregating ledger data among channel members and can be further divided into organizational units (OUs) for more granular access control. Channel MSPs are instantiated on every node's file system in the channel and synchronized via consensus, ensuring consistent policy enforcement and member authentication across the network.

Policies

A policy in Hyperledger Fabric is a set of rules that define decision-making structures and specific outcomes. Policies typically describe the who and what, such as the access or rights an individual has over an asset. For example, an insurance policy defines the conditions under which an insurance payout will be made, agreed upon by the policyholder and the insurance company. In Hyperledger Fabric, policies are essential for infrastructure management. They determine how members agree on changes to the network, a channel, or a smart contract. Policies are established when the channel is initially configured but can be modified as the channel evolves. They describe criteria for actions like adding or removing members, forming blocks, or specifying the number of organizations required to endorse a smart contract. Essentially, everything on a Fabric network is controlled by a policy.

Policies are crucial in Hyperledger Fabric because it is a permissioned Blockchain, unlike Ethereum or Bitcoin. In public Blockchains, transactions can be generated and validated by any node. In Fabric, recognized users decide on network governance before launch and can change governance on a running network. Policies allow members to decide which organizations can access or update the network and enforce those decisions. They contain lists of organizations with access to a resource and specify how many need to agree on proposals to update resources like channels or smart contracts.

Policies are defined within the administrative domain of the action they control. For instance, adding a peer organization to a channel is controlled by a policy in the Application group. Adding ordering nodes is controlled by a policy in the Orderer group. Actions that span both domains fall under the Channel group. Typically, these policies default to a majority of admins but can be customized. Policies evaluate the signatures attached to transactions and proposals, validating them against the agreed governance. Each smart contract has an endorsement policy specifying how many peers from different channel members must execute and validate a transaction for it to be considered valid. Endorsement policies define which organizations' peers must approve the execution of a proposal.

Modification policies specify which identities must sign off on configuration updates. They govern how policies themselves are updated, ensuring any changes have the required approvals. To change anything in Fabric, the policy associated with the resource describes who needs to approve it, either explicitly by individuals or implicitly by a group. Explicit sign-offs use the Signature syntax, while implicit sign-offs use the ImplicitMeta syntax.

Signature policies specify types of users who must sign for a policy to be satisfied, such as OR('Org1.peer', 'Org2.peer'). These policies are versatile, allowing for specific rules like requiring signatures from multiple organization members. ImplicitMeta policies are used in channel configuration, aggregating results from policies deeper in the configuration tree defined by Signature policies. They are implicit because they are constructed based on the current organizations in the channel configuration and are meta because they evaluate sub-policies below them in the configuration tree.

Peers

In a Hyperledger Fabric Blockchain network, peer nodes, or peers, are fundamental as they manage ledgers and smart contracts. Starting with Hyperledger Fabric v2.4, peers also handle transaction proposals and endorsements through the Fabric Gateway service. A ledger records all transactions generated by smart contracts, which are contained in chaincode. These contracts and ledgers encapsulate the processes and information shared by channel peers, making peers a critical element in understanding a Fabric network.

A Fabric Blockchain network is comprised of peers, each storing and managing copies of ledgers and smart contracts. For example in Figure 3.4, a network N may consist of peers P1, P2, and P3, each maintaining its own instance of the distributed ledger L1. Peers are flexible and can be created, started, stopped, reconfigured, and deleted. They expose APIs that enable client applications to interact with their services, particularly the Fabric Gateway service.

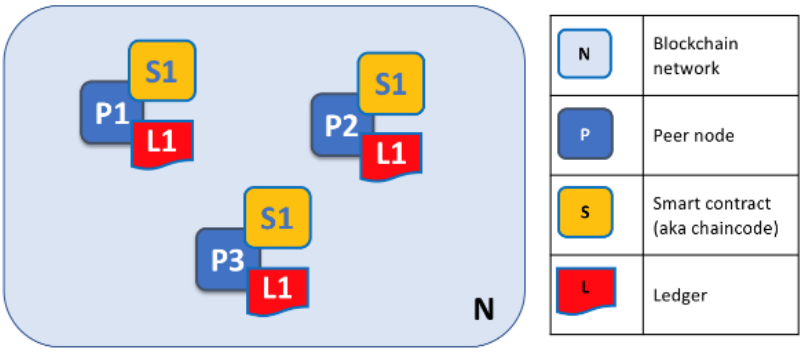

Figure 3.4: Peers

Fabric implements smart contracts through chaincode, which accesses the ledger using chaincode APIs. A peer hosts instances of ledgers and chaincode, ensuring consistent replicas across peers in a channel. This redundancy avoids single points of failure and maintains current ledgers. Client applications and administrators must interact with a peer to access these resources, making peers the building blocks of a Fabric network.

Starting in Hyperledger Fabric v2.4, the Fabric Gateway service is installed and enabled on each peer by default. This service manages transaction proposals and endorsements, simplifying application development. Client applications connect to the Fabric Gateway service on a peer to access ledgers and chaincode. Through this connection, applications can run chaincode to query or update the ledger.

Peers, alongside orderers, ensure ledger consistency across a channel. The process, as shown in Figure 3.5, involves three phases: transaction proposal and endorsement, transaction submission and ordering, and transaction validation and commitment. In the first phase, a client application submits a transaction proposal, which the peer executes and endorses. The second phase involves submitting the transaction to an ordering node, which packages it into blocks. In the final phase, peers validate and commit the transaction to the ledger, updating the channel's world state and sending a commit status event to the client.

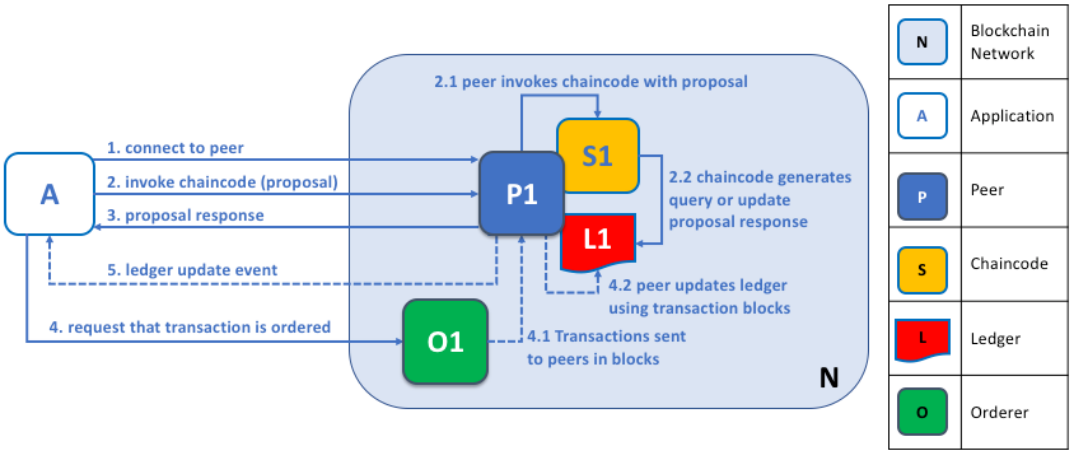

Figure 3.5:Transaction process

The Ledger

In Hyperledger Fabric, a ledger, represented in Figure 3.6, consists of two main parts: the world state and the Blockchain. The world state is a database that holds the current values of ledger states, making it easy to access current values directly. Ledger states are expressed as key-value pairs and can be frequently updated. The Blockchain, on the other hand, is a transaction log recording all changes that have led to the current world state. Transactions are collected in blocks appended to the Blockchain, providing an immutable history of changes. The ledger in a Hyperledger Fabric network is logically singular but has multiple consistent copies distributed throughout the network. This is achieved through a process called consensus, making it a Distributed Ledger Technology (DLT).

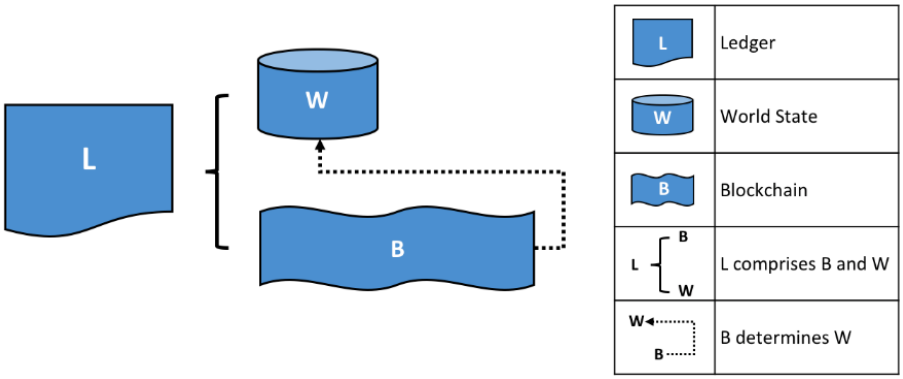

Figure 3.6: Hyperledger Fabric Ledger

The world state stores the current values of business object attributes as unique ledger states. Programs can directly access these values without traversing the entire Blockchain. Ledger states can be simple or complex, and the world state is implemented as a database to facilitate efficient storage and retrieval. Transactions that capture changes to the world state are committed to the ledger Blockchain, and only transactions signed by the required endorsing organizations result in an update to the world state. A ledger state also has a version number, incremented with each change to ensure consistency. Initially, the world state is empty and can be regenerated from the Blockchain at any time, which is useful for peer creation and recovery from failures.

The Blockchain provides a historical record of how ledger states have changed over time. It is structured as a sequential log of interlinked blocks, each containing a sequence of transactions. Block sequencing and transaction ordering are established by the ordering service, and each block's header includes hashes for security. The Blockchain is implemented as a file, optimized for appending new blocks and infrequent queries.

Together, the world state and Blockchain ensure that the ledger is both current and historically accurate, providing a robust foundation for a Hyperledger Fabric network.

The Ordering Service

Many distributed Blockchains like Ethereum and Bitcoin rely on probabilistic consensus algorithms, which can lead to divergent ledgers. Hyperledger Fabric, however, uses deterministic consensus algorithms facilitated by ordering nodes, forming an ordering service. This design ensures finality and prevents ledger forks. Orderers separate chaincode

execution from transaction ordering, enhancing performance and scalability. Orderers also enforce basic access control for channels and handle configuration transactions. They validate requests against existing policies, generate new configuration transactions, and package them into blocks for peer distribution. Each component in the network, including orderers, gets its identity from a Certificate Authority (CA) and Membership Service Provider (MSP).

In phase one of the transaction flow, a client application sends a transaction proposal to the Fabric Gateway service via a peer. The gateway forwards the transaction to peers required by the endorsement policy. In phase two, the endorsed transaction is sent to the ordering service, which orders and packages it into a block. In phase three, the block is distributed to all peers for validation and commitment to the ledger.

Hyperledger Fabric's ordering service ensures the strict sequencing of transactions, eliminating the possibility of ledger forks. Orderers do not execute smart contracts but focus on collecting, ordering, and packaging transactions into blocks. There are different implementations of the ordering service, including Raft, which is recommended for its crash fault tolerance and ease of setup, and Kafka, which is being deprecated in favor of Raft. Raft ordering service operates using a leader-follower model. A leader node is elected among the ordering nodes in a channel and replicates messages to follower nodes. Raft's crash fault tolerance ensures the service continues operating despite node failures, making it suitable for high availability strategies. Each channel runs on a separate instance of the Raft protocol, allowing decentralization and the dynamic addition or removal of ordering nodes. Leader election in Raft involves nodes starting as followers. If no log entries or heartbeats are received within a set time, nodes become candidates and request votes. A candidate receiving a quorum of votes becomes the leader, responsible for accepting and replicating log entries to followers. This process ensures a robust and fault-tolerant ordering service for Hyperledger Fabric networks.

Smart Contracts and Chaincode

From an application developer’s perspective, a smart contract, together with the ledger, forms the heart of a Hyperledger Fabric Blockchain system. Whereas a ledger holds facts about the current and historical state of a set of business objects, a smart contract defines the executable logic that generates new facts that are added to the ledger. A chaincode is typically used by administrators to group related smart contracts for deployment but can also be used for low-level system programming of Fabric. Before businesses can transact with each other, they must define a common set of contracts covering terms, data, rules, and processes. These contracts lay out the business model governing interactions between transacting parties. Using a Blockchain network, these contracts can be turned into executable programs called smart contracts. Smart contracts implement governance rules for any business object, automatically enforcing them when executed, making business processes more efficient. For example, a smart contract might ensure a car delivery within a specified timeframe or release funds according to prearranged terms.

Hyperledger Fabric users often use the terms smart contract and chaincode interchangeably. Generally, a smart contract defines the transaction logic controlling a business object's lifecycle in the world state. It is packaged into a chaincode for deployment to the network. Smart contracts govern transactions, while chaincode governs how smart contracts are packaged. A chaincode can contain multiple smart contracts. When deployed, all smart contracts within it become available to applications.

At its simplest level, a Blockchain immutably records transactions that update states in a ledger. A smart contract programmatically accesses two ledger components: a Blockchain recording transaction history and a world state holding the current value of these states. Smart contracts can put, get, and delete states in the world state and query the Blockchain record. The focus of application development in Fabric is on smart contracts, which are defined within a chaincode. Deploying a chaincode to a network makes all its smart contracts available to the organizations in that network. Associated with every chaincode is an endorsement policy indicating which organizations in a Blockchain network must sign a transaction for it to be valid. Endorsement policies ensure that transactions are validated by trusted organizations. For example, a government organization must sign a valid issueIdentity transaction, or both the buyer and seller of a car must sign a car transfer transaction. Valid transactions update the world state, while invalid ones do not, although both are recorded on the Blockchain.

Hyperledger Fabric allows organizations to participate in multiple, separate Blockchain networks via channels. Channels provide separate communication mechanisms, maintaining data and communications privacy while allowing coordination of independent activities when necessary. When a chaincode definition is committed to a channel, all smart contracts within it become available to applications on that channel. A chaincode definition includes parameters like name, version, and endorsement policy, agreed upon by channel members. Smart contracts can call other smart contracts within the same channel or across different channels, allowing them to read and write world state data they would not otherwise access. System chaincode defines low-level program code corresponding to domain-independent system interactions. Different types of system chaincode include lifecycle management, channel configuration changes, ledger queries, transaction endorsement, and validation processes.

3.3. Encryption

For encryption, which is not the primary focus of our work, we will use Rivest-Shamir-Adleman (RSA) encryption already mentioned in Section 2.6. The RSA cryptosystem is a widely used encryption technique that involves a pair of keys: a public key for encryption and a private key, which is secret, for decrypting the data encrypted with the public key.

Key Components

Public Key (e, n):

- **e**: A large prime number (commonly 65537).
- **n**: The product of two secret large prime numbers, p and q, each with hundreds of digits.

Private Key (d, n):

- **d**: Computed such that $d \times e \equiv 1 \bmod (p-1)(q-1)$.

Encryption and Decryption Process

- **Message (M)**: The plaintext message to be encrypted, represented as an integer.
- **Ciphertext (C)**: The encrypted message, calculated as C = M^e mod n.
- **Decryption**: The original message is retrieved using M = C^d mod n.

Example:

- Given M=2, e=7, p=3, q=5:
    - n = p × q = 3×5 =15
    - Find d such that d×7 ≡ 1 mod (2×4), which simplifies to d=7.

Encryption:

- C = 2^7 mod 15=8 (encrypted message)

Decryption:

- M = 8^7 mod 15=2

Problem: Identical messages yield the same output, which can be problematic given the limited number of possible outputs, complicating security especially when there are a limited number of distinct parties.

RSA with Optimal Asymmetric Encryption Padding (OAEP)

To enhance security, RSA can be combined with Optimal Asymmetric Encryption Padding (OAEP), which involves adding random bits to the message before encryption.

Process:

1. **Padding**: Add random bits to the message, known as a "seed" or "mask."
2. **Hashing**: Combine the padded message with a cryptographic hash function (such as SHA-1 or SHA-256) to add a layer of security.
3. **Encryption**: Encrypt the combined result using RSA.

This method ensures that even identical messages will produce different ciphertexts, thus enhancing security by mitigating the problem of identical outputs for the same input (Bellare & Rogaway, 1995).

3.4. System Architecture

Our system will operate per voting Assembly or Section, and it will be deployed at traditional on-site voting locations, replacing the usual paper-based methods. Each Assembly or Section will have its own dedicated Blockchain network, with physical machines as illustrated in the Figure 3.7.

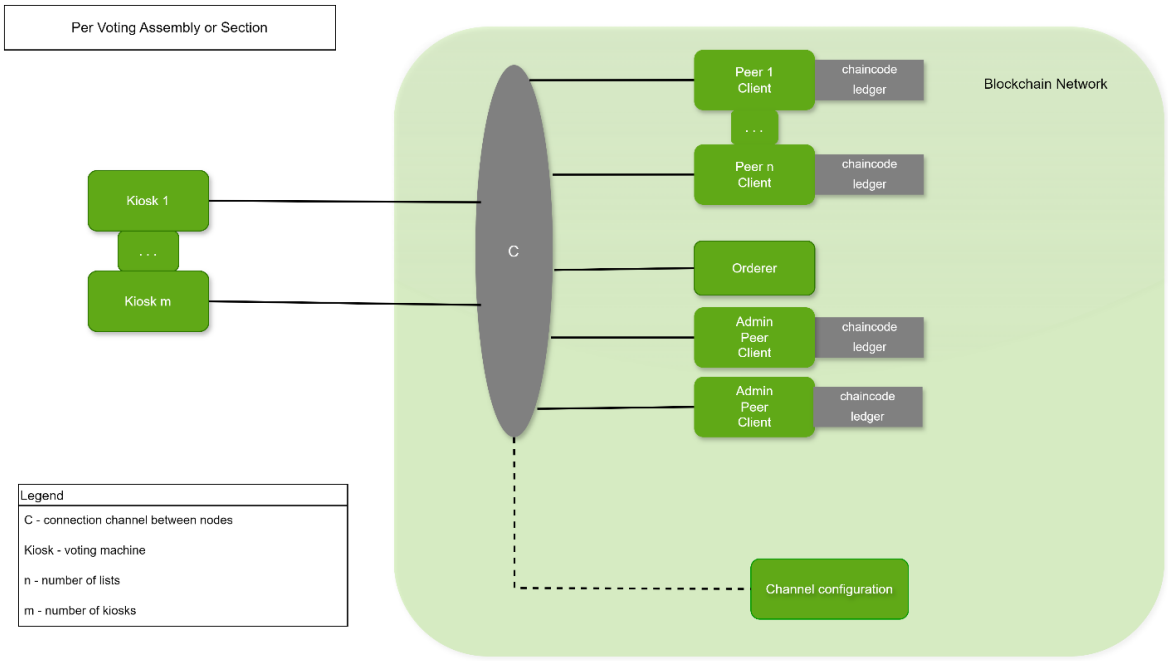

Figure 3.7: Concept System Architecture

The kiosks, whose number will depend on voter turnout, will allow voters to cast their votes and send them to the network. These machines will be capable of scanning identity cards and verifying the voter's face, adding the similarity percentage to the vote information. They will connect to the Administrator nodes to send transactions. Similar to the normal paper-based voting, there will still be electoral rolls in each kiosk with the list of eligible voters for initial recognition of the voter.

The Administrator Nodes will consist of three machines: two identical Admin nodes and one Orderer node. The two Admin nodes will act as both Peer and Client nodes. Connected to the Kiosks, they will be the first to receive votes for authentication. They will have the capability to send votes to the network for endorsement, to query the network, and to store a copy of the ledger and chaincode. The Orderer node will collect the endorsed votes and distribute them across the network.

The channel configuration will be public and defined during the network's creation, specifying the roles and permissions of each organization in the network. When adding a new party node, this configuration will be updated to accommodate the new organization.

Party nodes will function as both Peer and Client nodes. They will store a copy of the ledger and chaincode and will have the capacity to query the database, but they will not have access to the chaincode for adding new votes or querying vote intentions.

3.5. Working Process

Our voting process will be composed of three main parts, the beginning of the voting process, the voting process, and the count of the votes.

Voting Initialization

Before the voting process begins, the channel configuration is established, including the addition of peers, kiosks, and administrators. The administrator generates a public key using RSA cryptography with the OAEP algorithm, ensuring the private key remains isolated. This public key is shared with the kiosks for encryption, initiating the voting process. Each kiosk has the corresponding electoral roll for initial voter recognition by the staff, similar to traditional paper-based voting.

Voting Process

The voting process consists of three main parts: transaction proposal, execution and endorsement, transaction submission and ordering into blocks, and transaction validation and commitment.

First Part: Ledger Update (Write)

1. **Voters Recognition**:
    - The voter presents themselves at the table for initial recognition by the staff on the normal paper-based electoral rolls using their citizen card. Once successfully recognized, the voter proceeds to the kiosk, where they place their card on the machine. The machine checks the similarity of the photo with the voter and saves this information to add to the transaction proposal.
2. **Transaction Proposal**:
    - The voter selects their candidate, and the kiosk submits a voting proposal. This proposal includes the voter's information, vote similarity percentage, and the vote encrypted with RSA (OAEP). The proposal is signed and sent to the gateway service of the connected Admin Peer.
3. **Transaction Execution**:
    - The Peer executes the specific chaincode (S1) mentioned in the proposal, generates a response, signs this proposal, and returns it to the gateway.
4. **Transaction Endorsement**:
    - The gateway repeats the transaction execution process for the other peers (lists) according to the chaincode (smart contracts) endorsement policies. The gateway service collects the signatures of the proposals and creates a transaction envelope, which is then returned to the kiosk for signature.

Second Part: Transaction Submission and Ordering into Blocks

1. **Transaction Submission**:
    - The kiosk sends the signed transaction envelope to the gateway service. The gateway then forwards it to the ordering node and returns a success message to the client.

2. **Transaction Ordering**:
    - The ordering node verifies the signature, orders the transaction, and groups it with other transactions into blocks. The Orderer then distributes the block to all peers in the channel for validation and commitment to the ledger.

Third Part: Transaction Validation, Ledger Commitment, and Commit Event

1. **Transaction Validation**:
    - Each peer verifies that the client's signature on the transaction envelope matches the signature on the original transaction proposal. Each peer also checks that all read-write responses are consistent (endorsements from all peers) and that the endorsements satisfy the endorsement policies. Each peer then marks each transaction as valid or invalid for addition to the ledger.
2. **Transaction Commitment**:
    - Each peer commits the ordered block of transactions to its ledger. The commit is an immutable ledger update (write) for the channel's ledger. The world state of the channel is updated with the results of the valid transaction.
3. **Commit Event**:
    - Each peer that commits to the ledger sends the client a commit status event, providing proof that the ledger has been updated.

Voting Finalization

At the end of the voting process, an administrator program, which has access only to the encrypted votes, uses the private key to decrypt the votes and tally the results. Once the results are presented, the private key is eliminated to ensure security.

3.6. Smart Contracts and Chaincode

Our chaincode will follow the requirements outlined in Table 3.1 to create functions that closely follow the voting laws in Portugal. This approach ensures that the Electronic Voting process aligns with existing legal frameworks. To facilitate the development and testing of our system, this chaincode was created using the Hyperledger Fabric samples and is written in JavaScript. This enabled us to simulate the system and ensure its functionality. The chaincode will have various capabilities (functions) that will help the voting process run smoothly without any organization having the capability to compromise others. Each vote (transaction) will be composed of four parameters, as shown in Figure 3.8:

```
@Object()
export class Vote {
    @Property()
    public ID: string;

    @Property()
    public Name: string;

    @Property()
    public Biometric_confidence_level: string;

    @Property()
    public Intention: string;
}
```

Figure 3.8: Vote transaction properties. Source: elaborated by the author.

ID with the personal ID number of each voter, Name with the voter's name, biometric_confidence_level with the similarity percentage taken in the act of voting, and Intention with the encrypted vote intention.

The following chaincode functions that will add and manage the votes in the network:

Function "startVoting": can only be called once every election by the Administrator nodes, the first transaction that will be necessary for all the election to work will register the time it started.

```
@Transaction()
public async startVoting(ctx: Context, startTime: string): Promise<void> {
    const clientMSPID = ctx.clientIdentity.getMSPID();
    if (clientMSPID !== 'Org1MSP') {
        throw new Error('Unauthorized access. This function can only be used by Org1');
    }

    let state;
    try {
        state = await this.getState(ctx);
    } catch (error) {
        state = { votingActive: false, startTime: 'Not started', endTime: 'Not ended' };
    }

    state.votingActive = true;
    state.startTime = startTime;
    await ctx.stub.putState('votingState', Buffer.from(stringify(state)));
    console.info(`Voting started at ${startTime}`);
}
```

Figure 3.9: Function startVoting. Source: elaborated by the author.

Function "endVoting": can only be called once at the end of the voting process by the Administrator nodes, it will end the election, not letting any other transactions be processed, registering the time it ended.

```
@Transaction()
public async endVoting(ctx: Context, endTime: string): Promise<void> {
    const clientMSPID = ctx.clientIdentity.getMSPID();
    if (clientMSPID !== 'Org1MSP') {
        throw new Error('Unauthorized access. This function can only be used by Org1');
    }

    const state = await this.getState(ctx);
    state.votingActive = false;
    state.endTime = endTime;
    await ctx.stub.putState('votingState', Buffer.from(stringify(state)));
    console.info(`Voting ended at ${endTime}`);
}
```

Figure 3.10: Function endVoting. Source: elaborated by the author.

Function "CastVote": This is the only function that will serve as a transaction. It has four main characteristics. It will verify that the client submitting the vote is one of the Administrators, preventing other parties from submitting votes to the network, it will check if the vote is already on the network, ensuring that no one can vote twice, it will check if the biometric_confidence_level of the voter is greater than 80%, and can only be used during the designated voting period. This function follows many requirements of the Portuguese Electoral system, votes can only be cast when the voting period has started, with administrators consent, preventing fraudulent votes from being cast.

```
// CastVote issues a vote to the world state with given details.
@Transaction()
public async CastVote(ctx: Context, id: string, name: string, biometric_confidence_level: string, intention: string): Promise<void> {
    // Check If the voting period is active
    const state = await this.getState(ctx);
    if (!state.votingActive) {
        throw new Error('Voting is not active.');
    }

    // Retrieve the client identity from the context
    const clientMSPID = ctx.clientIdentity.getMSPID();

    // Check if the client belongs to Org1MSP
    if (clientMSPID !== 'Org1MSP') {
        throw new Error('Unauthorized access. This function can only be used by Org1');
    }

    const exists = await this.AssetExists(ctx, id);
    if (exists) {
        throw new Error(`The Vote ${id} already exists`);
    }

    // Check if the biometric_confidence_level is greater than 80      You, last week • Uncommitted changes
    if (parseInt(biometric_confidence_level) <= 80) {
        throw new Error('The biometric_confidence_level is bellow 80, the person is a fake');
    }

    const Vote = {
        ID: id,
        Name: name,
        Biometric_confidence_level: biometric_confidence_level,
        Intention: intention,
    };
    // we insert data in alphabetic order using 'json-stringify-deterministic' and 'sort-keys-recursive'
    await ctx.stub.putState(id, Buffer.from(stringify(sortKeysRecursive(Vote))));
    console.info(`Vote ${id} casted`);
}
```

Figure 3.11: Function CastVote. Source: elaborated by the author.

Function "ReadVote": This function will take an ID and check the ledger to see if the vote already exists. It will respond with either a message indicating that the vote does not exist, or the information associated with the request, excluding the voter's intention for security reasons . This allows any election worker to verify if a specific person's vote has already been cast.

```
@Transaction(false)
public async ReadVote(ctx: Context, id: string): Promise<string> {
    const assetJSON = await ctx.stub.getState(id); // get the asset from chaincode state
    if (!assetJSON || assetJSON.length === 0) {
        throw new Error(`The Vote ${id} does not exist`);
    }
    let asset = JSON.parse(assetJSON.toString());
    delete asset.Intention; // Remove the Intention property
    return JSON.stringify(asset);
}
```

Figure 3.12: Function ReadVote. Source: elaborated by the author.

Function "GetAllVotes": This function will return all the votes recorded in the ledger by alphabetical order, excluding the voters' intentions for security reasons. It provides only the list of votes present in the network, allowing for a worker to check all the existing votes in the network.

```
@Transaction(false)
@Returns('string')
public async GetAllVotes(ctx: Context): Promise<string> {
    const allResults = [];
    // range query with empty string for startKey and endKey does an open-ended query of all assets in the chaincode namespace.
    const iterator = await ctx.stub.getStateByRange('', '');
    let result = await iterator.next();
    while (!result.done) {
        const strValue = Buffer.from(result.value.value.toString()).toString('utf8');
        let record;
        try {
            record = JSON.parse(strValue);
            // Skip the voting state entry
            if (record.votingActive !== undefined) {
                result = await iterator.next();
                continue;
            }
            // Remove the Intention property before adding to the results
            delete record.Intention;
        } catch (err) {
            console.log(err);
            record = strValue;
        }
        allResults.push(record);
        result = await iterator.next();
    }

    // Sort results alphabetically by Name property
    allResults.sort((a, b) => {
        if (a.Name < b.Name) {
            return -1;
        }
        if (a.Name > b.Name) {
            return 1;
        }
        return 0;
    });

    return JSON.stringify(allResults);
}
```

Figure 3.13: Function GetAllVotes. Source: elaborated by the author.

Function "shuffleArray": A private function used solely for randomizing an array. It will be used to randomize the array of intentions for the counting process.

```
// Helper function to shuffle an array
private shuffleArray(array: any[]): any[] {
    for (let i = array.length - 1; i > 0; i--) {
        const j = Math.floor(Math.random() * (i + 1));
        [array[i], array[j]] = [array[j], array[i]];
    }
    return array;
}
```

Figure 3.14: Function shuffleArray. Source: elaborated by the author.

Function "GetAllIntentions": This function can only be called by Administrator nodes after the voting period ends. It will return only the vote intentions, using the "**shuffleArray"** function to randomize the votes. It will not return any other information corresponding to each vote.

```
// Function to return all intentions in a randomized order, callable only by Org1 if voting has ended
@Transaction(false)
@Returns('string')
public async GetAllIntentions(ctx: Context): Promise<string> {
    const clientMSPID = ctx.clientIdentity.getMSPID();
    if (clientMSPID !== 'Org1MSP') {
        throw new Error('Unauthorized: this function can only be called by Org1');
    }

    const state = await this.getState(ctx);
    if (state.votingActive) {
        throw new Error('Voting is still active. Intentions can only be retrieved after voting has ended.');
    }

    const allIntentions = [];
    const iterator = await ctx.stub.getStateByRange('', '');
    let result = await iterator.next();
    while (!result.done) {
        const strValue = Buffer.from(result.value.value.toString()).toString('utf8');
        let record;
        try {
            record = JSON.parse(strValue);
            if (record.Intention) {
                allIntentions.push(record.Intention);
            }
        } catch (err) {
            console.log(err);
        }
        result = await iterator.next();
    }
    return JSON.stringify(this.shuffleArray(allIntentions));
}
```

Figure 3.15: Function GetAllIntentions. Source: elaborated by the author.

Function "getVotingState" and "getState": These functions work together to return if the voting state is active, and the start and end times of the voting period, they also help other functions in validation.

```
@Transaction(false)
@Returns('string')
public async getVotingState(ctx: Context): Promise<string> {
    let state;
    try {
        state = await this.getState(ctx);
    } catch (error) {
        // Handle the case where the state does not exist
        state = {
            votingActive: false,
            startTime: 'Not started',
            endTime: 'Not ended',
        };
    }

    const votingState = {
        votingActive: state.votingActive,
        startTime: state.startTime || 'Not started',
        endTime: state.endTime || 'Not ended',
    };
    return JSON.stringify(votingState);
}

private async getState(ctx: Context): Promise<any> {
    const stateAsBytes = await ctx.stub.getState('votingState');
    if (!stateAsBytes || stateAsBytes.length === 0) {
        throw new Error('Voting state not found');
    }
    return JSON.parse(stateAsBytes.toString());
}
```

Figure 3.16: Function getVotingState and getState. Source: elaborated by the author.

To provide a comprehensive understanding of how these functions interact to facilitate a secure and transparent voting process, let's walk through the steps involved.

Before the voting begins, the Administrator nodes call the "startVoting" function to initialize the election and record the start time. This crucial step ensures that no votes can be cast before the official start time, setting the stage for a structured and time-bound voting period.

During the voting period, voters approach the voting machines to cast their votes. The "CastVote" function is invoked to submit their votes, where it performs several important checks: verifying the voter's identity, checking for duplicate votes, ensuring that the biometric confidence level is sufficient, and confirming that the voting period is active. These steps collectively ensure that only valid votes are recorded, preventing any unauthorized or fraudulent voting activities.

To maintain transparency and prevent multiple votes by the same individual, election workers can use the "ReadVote" function. This function allows them to verify if a voter's vote has

already been cast, providing a reliable mechanism for monitoring voter participation. Throughout the voting period, the "GetAllVotes" function plays a critical role in ensuring that all votes are accounted for. It allows election workers to monitor all votes recorded in the ledger, providing a comprehensive overview of the voting activity.

At the end of the voting period, the Administrator nodes call the "endVoting" function to officially close the election and record the end time. This function is essential for preventing any further votes from being cast, thereby securing the integrity of the election process once the designated voting time has elapsed. Once the voting period ends, the focus shifts to counting the votes. The "GetAllIntentions" function is used to retrieve and randomize the vote intentions for tallying. To ensure voter privacy and maintain the integrity of the count, the "shuffleArray" function is employed to randomize the order of the votes, making the tallying process fair and transparent.

By integrating these steps, our system demonstrates a robust approach to electronic voting, leveraging blockchain technology to enhance security, transparency, and reliability. This final section highlights how the system adheres to legal requirements and ensures a seamless voting experience in Portugal.

Conclusions and Future Work

Blockchain is a groundbreaking technology that offers decentralization, immutability, and transparency, making it an ideal solution for secure and reliable Electronic Voting systems. Its potential to revolutionize transaction management across various applications is underscored by its core features and the benefits they bring. Blockchain's decentralized nature eliminates the need for third-party intermediaries, enhancing security and reducing the potential for fraud. Immutability ensures that once data is recorded, it cannot be altered or deleted, providing a reliable and tamper-proof record of transactions. Transparency allows all participants to verify the integrity of the data, fostering trust in the system.

The research delved deeply into these fundamental aspects of Blockchain, exploring different types of Blockchains—public, private, and consortium—and consensus protocols, Proof of Work (PoW) and Proof of Stake (PoS). It also addressed the inherent challenges and issues such as scalability, energy consumption, and security vulnerabilities, providing a comprehensive understanding of the technology and its implications for Electronic Voting.

Electronic Voting has been a subject of research and implementation for several decades, offering numerous advantages over traditional paper-based voting. These benefits include environmental sustainability, as it reduces the need for paper ballots, and real-time counting and processing, which speeds up the tallying process and reduces human error. Additionally, Electronic Voting can enhance overall voter turnout by making the process more accessible and convenient. However, Electronic Voting systems face significant challenges, particularly in ensuring security, privacy, and resistance to fraud and coercion. Centralized systems are susceptible to various cyber-attacks, and ensuring the trustworthiness of these systems remains a critical concern.

In this work, I developed a Blockchain-based Electronic Voting system tailored specifically for the Portuguese electoral process. By leveraging Hyperledger Fabric, a permissioned Blockchain framework, the system addresses critical requirements such as voter privacy, vote integrity, and system transparency. The smart contracts and chaincode developed for managing the voting process automate and secure election procedures. Functions like startVoting, endVoting, CastVote, and ReadVote ensure that the election is conducted securely and transparently, with verifiable and immutable records of each vote. This not only enhances the security of the voting process but also increases transparency, allowing all parties to verify the integrity of the election.

The integration of biometric data, particularly biometric confidence levels, into the voting process further enhances security and authenticity in voter identification, significantly reducing the risk of fraudulent voting activities. Biometric authentication ensures that only eligible voters can cast their votes by comparing the photo on the voter's ID to the biometric data captured at the voting machine. This real-time verification process, combined with Blockchain technology, guarantees that each vote is recorded immutably and transparently. This dual layer of security is critical in maintaining the integrity of the electoral process.

The proposed system architecture offers a practical approach to implementing a Blockchain-based Electronic Voting system within the existing legal and procedural framework of Portuguese elections. The workflow ensures compliance with current electoral laws while

incorporating advanced technological features. Voters are verified using normal paper lists before being directed to the voting machine to cast their votes. This approach allows for a seamless integration of new technology with existing systems, facilitating a gradual transition to more secure and efficient voting processes.

Despite these promising results, several challenges remain, including the need for secure and reliable network systems, the scalability of the Blockchain network, and balancing transparency with voter privacy. Ensuring that the entire voting system is secure is crucial for the integrity of the voting process. Given the complexity of these systems, even a small mistake can compromise the entire process, highlighting the need for meticulous attention to detail and rigorous testing. Scalability issues need to be addressed to handle larger volumes of transactions, especially in national elections. Moreover, maintaining voter privacy while ensuring transparency requires careful consideration and the implementation of advanced cryptographic techniques.

These challenges highlight areas for further research and development. Future research should focus on addressing scalability issues inherent in Blockchain technology. Enhancing the user experience, particularly for non-technical voters, is critical for broader adoption. Future studies should develop more intuitive interfaces and ensure accessibility for voters with disabilities. Another important area for future research is improving the encryption methods used to ensure that the vote intention is not directly connected to the voter's identity. This could involve developing better encryption techniques or creating systems that disconnect vote intention from personal identifiers.

The recent European elections in Portugal provide an interesting case for consideration. In these elections, each polling station was equipped with two PCs connected to a network for the electoral rolls, allowing voters to vote at any location while maintaining the use of traditional paper ballots (CNE, 2023). This hybrid system could be a valuable model for future implementations. By replacing paper-based electoral rolls with a secure digital system, voters can cast their votes anywhere. This system would register a token from the location or voter, enhancing flexibility and convenience without compromising security.

By addressing these areas, future research can build on the foundation laid by this thesis, contributing to the development of more secure, reliable, and accessible Electronic Voting systems. The potential for Blockchain technology to transform Electronic Voting is immense, and continued research and innovation in this field will be crucial for realizing its full benefits.

Bibliographical References

Agbesi, S., & Asante, G. (2019). Electronic Voting Recording System Based on Blockchain Technology. *2019 12th CMI Conference on Cybersecurity and Privacy (CMI)*, 1–8. https://doi.org/10.1109/CMI48017.2019.8962142

Ahmad, T., Hu, J., & Han, S. (2009). An Efficient Mobile Voting System Security Scheme Based on Elliptic Curve Cryptography. *2009 Third International Conference on Network and System Security*, 474–479. https://doi.org/10.1109/NSS.2009.57

Aranha, D. F., & van de Graaf, J. (2018). The Good, the Bad, and the Ugly: Two Decades of E-Voting in Brazil. *IEEE Security & Privacy*, *16*(6), 22–30. IEEE Security & Privacy. https://doi.org/10.1109/MSEC.2018.2875318

Avalanche. (2024). *Avalanche: Create Without Limits | dApp Platform*. https://www.avax.network/

Bellare, M., & Rogaway, P. (1995). *Optimal Asymmetric Encryption How to Encrypt with RSA*.

Ben Ayed, A. (2017). *A CONCEPTUAL SECURE BLOCKCHAIN-BASED ELECTRONIC VOTING SYSTEM*. https://doi.org/10.5121/ijnsa.2017.9301

Bhutta, M. N. M., Khwaja, A. A., Nadeem, A., Ahmad, H. F., Khan, M. K., Hanif, M. A., Song, H., Alshamari, M., & Cao, Y. (2021). A Survey on Blockchain Technology: Evolution, Architecture and Security. *IEEE Access*, *9*, 61048–61073. https://doi.org/10.1109/ACCESS.2021.3072849

Bond, S. (2022, October 22). *Why false claims about Brazil's election are spreading in far-right US circles*. Opb. https://www.opb.org/article/2022/10/22/why-false-claims-about-brazil-s-election-are-spreading-in-far-right-u-s-circles/

Carcia, J. C. P., Benslimane, A., & Boutalbi, S. (2021). Blockchain-based system for e-voting using Blind Signature Protocol. *2021 IEEE Global Communications Conference (GLOBECOM)*, 01–06. https://doi.org/10.1109/GLOBECOM46510.2021.9685189

Chatterjee, R., & Chatterjee, R. (2017). An Overview of the Emerging Technology: Blockchain. *2017 3rd International Conference on Computational Intelligence and Networks (CINE)*, 126–127. https://doi.org/10.1109/CINE.2017.33

Chu, S., & Wang, S. (2018). *The Curses of Blockchain Decentralization* (arXiv:1810.02937). arXiv. http://arxiv.org/abs/1810.02937

CNE. (2023, December 28). *Lei n.º 80/2023 – Regimes excecionais de exercício do direito de voto em mobilidade e do direito de voto antecipado para a eleição para o Parlamento Europeu a realizar em 2024 | Comissão Nacional de Eleições*. https://www.cne.pt/news/lei-no-802023-regimes-excecionais-de-exercicio-do-direito-de-voto-em-mobilidade-e-do-direito-de-voto_7953

Corda. (2024). *Home Page*. Corda. https://corda.net/

Di Pierro, M. (2017). What Is the Blockchain? *Computing in Science & Engineering*, *19*(5), 92–95. https://doi.org/10.1109/MCSE.2017.3421554

Díaz-Santiso, J., & Fraga-Lamas, P. (2021). E-Voting System Using Hyperledger Fabric Blockchain and Smart Contracts. *Engineering Proceedings*, *7*(1), Article 1. https://doi.org/10.3390/engproc2021007011

Ehin, P., Solvak, M., Willemson, J., & Vinkel, P. (2022). Internet voting in Estonia 2005–2019: Evidence from eleven elections. *Government Information Quarterly*, *39*(4), 101718. https://doi.org/10.1016/j.giq.2022.101718

Ethereum. (2024). *Início*. ethereum.org. https://ethereum.org/pt/

Ferdous, M. S., Chowdhury, M. J. M., Hoque, M. A., & Colman, A. (2020). *Blockchain Consensus Algorithms: A Survey* (arXiv:2001.07091). arXiv. http://arxiv.org/abs/2001.07091

Gao, W., Hatcher, W. G., & Yu, W. (2018). A Survey of Blockchain: Techniques, Applications, and Challenges. *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, 1–11. https://doi.org/10.1109/ICCCN.2018.8487348

Gerlach, J., & Gasser, U. (2009). *Three Case Studies from Switzerland: E-Voting*.

Ghiro, L., Restuccia, F., D'Oro, S., Basagni, S., Melodia, T., Maccari, L., & Cigno, R. L. (2021). *What is a Blockchain? A Definition to Clarify the Role of the Blockchain in the Internet of Things* (arXiv:2102.03750). arXiv. http://arxiv.org/abs/2102.03750

Hjalmarsson, F. P., Hreioarsson, G. K., Hamdaqa, M., & Hjalmtysson, G. (2018). Blockchain-Based E-Voting System. *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, 983–986. https://doi.org/10.1109/CLOUD.2018.00151

Hyperledger Fabric. (2024). https://www.hyperledger.org/projects/fabric

Hyperledger Fabric Docs. (2023). *A Blockchain Platform for the Enterprise—Hyperledger Fabric Docs main documentation*. https://hyperledger-fabric.readthedocs.io/en/release-2.5/index.html

Jafar, U., Aziz, M. J. A., & Shukur, Z. (2021). Blockchain for Electronic Voting System—Review and Open Research Challenges. *Sensors*, *21*(17), Article 17. https://doi.org/10.3390/s21175874

Jornal de Notícias. (2023, March 11). *Novos estatutos do F. C. Porto incluem voto eletrónico e eleições até junho*. https://www.jn.pt/1683691466/novos-estatutos-do-f-c-porto-incluem-voto-eletronico-e-eleicoes-ate-junho/

Kshetri, N., & Voas, J. (2018). Blockchain-Enabled E-Voting. *IEEE Software*, *35*(4), 95–99. https://doi.org/10.1109/MS.2018.2801546

Lashkari, B., & Musilek, P. (2021). A Comprehensive Review of Blockchain Consensus Mechanisms. *IEEE Access*, *9*, 43620–43652. https://doi.org/10.1109/ACCESS.2021.3065880

Legislação Eleitoral | Comissão Nacional de Eleições. (2024). https://www.cne.pt/content/legislacao-eleitoral

Liu, Y., & Wang, Q. (2017). *An E-voting Protocol Based on Blockchain* (2017/1043). https://eprint.iacr.org/2017/1043

Montargil, F. (2004). *Voto Electrónico em Portugal e Democracia*.

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*.

Ordem dos Advogados. (2022, November 3). *Eleições 2023-2025 | Mensagem sobre Votação Electrónica*. Ordem Dos Advogados. https://portal.oa.pt/ordem/historia/eleicoes/eleicoes-2023-2025/eleicoes-2023-2025-mensagem-sobre-votacao-electronica/

Oyinloye, D. P., Teh, J. S., Jamil, N., & Alawida, M. (2021). Blockchain Consensus: An Overview of Alternative Protocols. *Symmetry*, *13*(8), Article 8. https://doi.org/10.3390/sym13081363

Panwar, A., & Bhatnagar, V. (2020). Distributed Ledger Technology (DLT): The Beginning of a Technological Revolution for Blockchain. *2nd International Conference on Data, Engineering and Applications (IDEA)*, 1–5. https://doi.org/10.1109/IDEA49133.2020.9170699

Polge, J., Robert, J., & Le Traon, Y. (2021). Permissioned blockchain frameworks in the industry: A comparison. *ICT Express*, *7*(2), 229–233. https://doi.org/10.1016/j.icte.2020.09.002

Rajasekaran, A. S., Azees, M., & Al-Turjman, F. (2022). A comprehensive survey on blockchain technology. *Sustainable Energy Technologies and Assessments*, *52*, 102039. https://doi.org/10.1016/j.seta.2022.102039

Regulamento Eleitoral do Sport Lisboa e Benfica. (2021, September 24). https://www.slbenfica.pt/pt-pt/agora/noticias/2021/09/24/clube-benfica-regulamento-eleitoral-para-as-eleicoes-dos-orgaos-sociais-dia-9-de-outubro

Renascença. (2023, October 9). *Sporting. Voto eletrónico não passa, mas Varandas promete voltar ao assunto—Renascença*. Rádio Renascença. https://rr.sapo.pt/bola-branca/noticia/sporting/2023/10/09/sporting-voto-eletronico-nao-passa-mas-varandas-promete-voltar-ao-assunto/349966/

Rivest, R. L., Shamir, A., & Adleman, L. (1978). *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*.

Sankar, L. S., Sindhu, M., & Sethumadhavan, M. (2017). Survey of consensus protocols on blockchain applications. *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1–5. https://doi.org/10.1109/ICACCS.2017.8014672

Seftyanto, D., Amiruddin, A., & Hakim, A. R. (2019). Design of Blockchain-Based Electronic Election System Using Hyperledger: Case of Indonesia. *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 228–233. https://doi.org/10.1109/ICITISEE48480.2019.9003768

SGMAI. (2019, May 20). *Secretaria Geral do MAI*. https://www.sg.mai.gov.pt/administracaoeleitoral/publicacoes/relatoriosatoseleitorais/Paginas/default.aspx

Singh, A., Ganesh, A., Patil, R. R., Kumar, S., Rani, R., & Pippal, S. K. (2023). Secure Voting Website Using Ethereum and Smart Contracts. *Applied System Innovation*, *6*(4), Article 4. https://doi.org/10.3390/asi6040070

STAPE. (2001). *Ensaio Piloto de Voto Electrónico—Preservada pelo Arquivo.pt*. https://arquivo.pt/wayback/20100807103955/http://www.stape.pt/eleiref/ensaio.htm

Stenerud, I. S. G., & Bull, C. (2011). *When Reality Comes Knocking Norwegian Experiences with Verifiable Electronic Voting*.

Taş, R., & Tanrıöver, Ö. Ö. (2020). A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting. *Symmetry*, *12*(8), Article 8. https://doi.org/10.3390/sym12081328

Tezos. (2024). *Home*. Tezos. https://tezos.com

Tornos, J. L., Salazar, J. L., & Piles, J. J. (2015). Optimizing ring signature keys for e-voting. *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 817–821. https://doi.org/10.1109/IWCMC.2015.7289188

Tu, S.-F., Hsu, C.-S., & You, B.-L. (2023). An On-Site Electronic Voting System Using Blockchain and Biometrics. *The International Arab Journal of Information Technology*, *20*(5). https://doi.org/10.34028/iajit/20/5/13

Twesige, R. (2015). *Bitcoin A simple explanation of Bitcoin and Block Chain technology JANUARY 2015 RICHARD LEE TWESIGE*. https://doi.org/10.13140/2.1.1385.2486

Xinyi, Y., Yi, Z., & He, Y. (2018). Technical Characteristics and Model of Blockchain. *2018 10th International Conference on Communication Software and Networks (ICCSN)*, 562–566. https://doi.org/10.1109/ICCSN.2018.8488289

Yang, R., Yu, F. R., Si, P., Yang, Z., & Zhang, Y. (2019). Integrated Blockchain and Edge Computing Systems: A Survey, Some Research Issues and Challenges. *IEEE*

Communications Surveys & Tutorials, *21*(2), 1508–1532. https://doi.org/10.1109/COMST.2019.2894727

Zarrin, J., Wen Phang, H., Babu Saheer, L., & Zarrin, B. (2021). Blockchain for decentralization of internet: Prospects, trends, and challenges. *Cluster Computing*, *24*(4), 2841–2866. https://doi.org/10.1007/s10586-021-03301-8

Zhong, Y. (2022). An Overview of RSA and OAEP Padding. *Highlights in Science, Engineering and Technology*, *1*, 82–86. https://doi.org/10.54097/hset.v1i.431

Zīle, K., & Strazdiņa, R. (2018). Blockchain Use Cases and Their Feasibility. *Applied Computer Systems*, *23*(1), 12–20. https://doi.org/10.2478/acss-2018-0002

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa