

# iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA

---

## **Electronic Voting Through Blockchain: A proposed Framework**

Antonio Lorenzo Rezende de Castro

Master in Computer Science and Business Management

Supervisor:

PhD Carlos Eduardo Dias Coutinho, Assistant Professor, Iscte

October, 2023

(This page was intentionally left blank)

## **Acknowledgments**

I would like to express my gratitude to those who made this work possible. I am deeply thankful for the guidance and support of my professor and advisor, Carlos Coutinho, who provided guidance and support in the writing of this dissertation also appreciate the support of my friends and family. This dissertation would not be complete without the participation of those who shared their insights.

Thank you all for your essential contributions.

(This page was intentionally left blank)

## Resumo

Eleições são um assunto predominante na história moderna, uma vez que a democracia tornou-se o sistema preferencial da maioria dos atuais governos, o interesse de atores maliciosos em corromper uma eleição tem-se tornado crescente. Vários países têm explorado o voto eletrônico, como o Brasil, Austrália e Paquistão. Com o desenvolvimento da tecnologia blockchain e sua natureza imutável e transparente, houve múltiplos estudos explorando o uso da tecnologia em sistemas de votação e examinando a sua flexibilidade em manter o direito de privacidade dos eleitores.

Este trabalho é composto por duas partes principais. Em primeiro lugar, uma Revisão Sistemática da Literatura sobre a pesquisa atual do Estado da Arte em sistemas de voto baseados em blockchain, explorando soluções, categorizando e identificando pontos na taxonomia dessas soluções. Em segundo lugar, a Metodologia de Design Science Research (DSR) foi escolhida para construir o artefacto, neste caso, uma solução de votação baseada em blockchain. Foram desenvolvidas versões da solução com complexidade e tecnologias crescentes, explorando paradigmas atuais e também novos desenvolvimentos criptográficos. Em cada iteração, uma avaliação de aspetos como privacidade, escalabilidade e segurança foi realizada para identificar melhorias necessárias.

Foi identificado que as tecnologias blockchain atualmente usadas, como Ethereum e outras Soluções de Layer 2, ainda carecem escalabilidade e privacidade. No entanto, a exploração de novos paradigmas criptográficos como ZK-Starks e o desenvolvimento de "roll ups" Ethereum que utilizam máquinas virtuais demonstraram avanço significativo na privacidade dos eleitores, escalabilidade e descentralização, embora essa tecnologia ainda não esteja pronta para ser usada em produção.

**Palavras-Chave:** Blockchain, Framework, E-Voting, Sistemas Distribuídos



## Abstract

Voting has been a prevalent matter in modern history, and since democracy has been the go-to system for governing nations the interest from malicious actors to disrupt an election has been notorious. Multiple countries have been exploring electronic voting as of Brazil, Australia, and Pakistan. With the development of blockchain technology and its immutable and transparent nature there has been multiple studies exploring the use of blockchain technology for voting systems and exploring its flexibility and challenges with maintaining voters' privacy.

This work is composed of two main parts. Firstly, a Systematic Literature Review on the current State of the Art research for blockchain voting systems, exploring solutions and reviews, and categorizing and identifying multiple points in the taxonomy of these solutions. Secondly the Design Science Research (DSR) was chosen to build the artefact, in this case a blockchain voting solution. Multiple versions of the solution were developed with increased complexity and technologies. Exploring the paradigm of the currently available blockchain solutions and new and upcoming cryptographic developments. With each iteration, an evaluation of aspects such as Privacy, Scalability and security were done to identify needed improvements.

It was identified that currently used blockchain technology such as Ethereum and other Layer 2 Solutions still lacks the scalability or privacy. That said the exploration of Zero-Knowledge Proof Cryptography and developing roll ups for Ethereum that use Zero-Knowledge Virtual machined Showed a significant development in voters' privacy, scalability, and decentralization, although this technology is not yet ready to be used in production.

**Keyword:** Blockchain, Framework, E-Voting, Distributed Systems





## Index

Acknowledgments.....	iii
Resumo.....	v
Abstract.....	vii
Chapter 1.....	1
Introduction.....	2
Motivation.....	2
Research Methodology.....	4
Research Objectives.....	2
Chapter 2.....	7
Conducting the Review.....	9
Discussion and Conclusion of Review.....	22
Outlining our Literature review.....	8
Pre-Solution State.....	19
Chapter 3.....	25
Evaluation.....	29
Proposal.....	27
Technology Stack.....	25
Chapter 4.....	31
Evaluation.....	36
Problem Identification.....	31
Proposal of Layer 2 Solution.....	34
Research and Analysis.....	32
Chapter 5.....	38
Development of Proposal.....	41
Evaluation.....	46
Out EVM Voting Solution.....	43
Research and Analysis.....	38
Chapter 6.....	49
Contributions.....	49
Future Work.....	50
Limitations.....	50

## Figures Index

Figure 1 – Design Science Research Methodology .....	4
Figure 2 - SLR Structure.....	7
Figure 3 - Distribution of Articles by Publication Type.....	10
Figure 4 - Distribution of Articles by Year .....	11
Figure 5 - Overview of Ethereum Voting Solution .....	26
Figure 6 - Smart Contract Data Flow .....	27
Figure 7 - Application Data Flow .....	28
Figure 8 - Overview of Layer2 Voting Solution .....	34
Figure 9 - Polygon Proof of Stake Architecture .....	35
Figure 10 - Comparison between Block time Ethereum and Polygon .....	36
Figure 11 - Comparison between Transaction Count in 24 hour period .....	36
Figure 12 - Miden Transaction Model.....	42
Figure 13 - Miden Account Model .....	42
Figure 14 – zk-Evm Solution dataflow.....	43
Figure 15 - Debug Inputs .....	44
Figure 16 – Definition of Field Elements from input of unsigned Integers .....	44
Figure 17 – Load of Advice Provider .....	44
Figure 18 – MASM Contract.....	45
Figure 19 – Prove function .....	45
Figure 20 – Redefinition of Field Elements from Stack Elements .....	46
Figure 21 – Debug Logs .....	46

## Tables Index

Table 1 - Design Science Research Guidelines .....	5
Table 2 – Key Evaluation Questions .....	6
Table 3- Number of extracted Surveys.....	6
Table 4 - SLR Filters .....	9
Table 5 - Conference Ranking.....	10
Table 6 - Journal Ranking.....	10
Table 7 - Relative Comparison of “Blockchain” and “E-voting” Proposals IEEE .....	12
Table 8 - Relative Comparison of “Blockchain” and “E-voting” Proposals ACM.....	16
Table 9 – Relative Comparison of “Blockchain” and “E-voting” Proposals SpringerLink.....	18
Table 10 - Number of extracted Surveys.....	19
Table 11 - Relative Comparison of Extracted Surveys .....	20
Table 12 - Distribution of Tools of Stake.....	22
Table 13 – Single Core Benchmarks .....	47

# List of abbreviations and acronyms

**API** - Application Programming Interface

**CLI** - Command Line Interface

**DAO** - Decentralized Autonomous Application

**Dapp** - Decentralised Application

**DSR** - Design Science Research

**DSRM** – Design Science Research Methodology

**ECC** - Elliptic Curve Cryptography

**ETH** - Ether

**EVM** – Electronic Voting Machines

**E-voting** - Electronic Voting

**FOO** - Fujioka, Okamoto and Ohta

**GETH** - Go Ethereum

**GUI** - Graphical User Interface

**IT** - Information Technology

**L1** - Layer 1

**L2** - Layer 2

**MASM** - Miden Assembly

**PoA** - Proof of Authority

**PoS** - Proof of Stake

**PoW** - Proof of Work

**RSA** - Rivest-Shamir-Adleman

**RQ** - Research Question

**SGX** - Software Guard Extension

**SLR** - Systematic Literature Review

**TPS** - Transactions Per

**UTXO** - Unspent Transaction Output

**VM** - Virtual Machine

**ZK** - Zero Knowledge

**Zk-SNARK** - Zero Knowledge Non-Interactive Argument Knowledge

**Zk-STARK** - Zero Knowledge Scalable Transparent Argument of Knowledge

**Zk-VM** - Zero Knowledge Virtual Machine

**Zk-EVM** - Zero Knowledge Ethereum Virtual Machine



## CHAPTER 1

### **Introduction**

As technology advances, we find ourselves increasingly immersed in a digitally driven world, where a significant portion of our daily activities occurs in digital spaces. This digital shift not only leads to the generation of more data but also exposes our data to heightened vulnerabilities from malicious actors. The exposing of data has grown substantially in tandem with the expanding digital presence of corporations and the soaring value of the vast datasets they accumulate (Brooks, 2023). The consequence of this is a notable surge in cyber-attacks, particularly ransomware incidents orchestrated by groups like Revil (Accenture, 2021).

Nonetheless voting has held a significant place in modern history, and since democracy has been the go-to system for governing nations the eagerness of malicious actors to interfere with elections has become notorious. Many of the electronic voting systems have yet to gain the full trust of the public (Khanpara et al., 2022). Elections, as pivotal pillars of democracy, represent a particularly vulnerable target for cyber-attacks, ranging from groups seeking financial gain to political parties striving to gain an edge in fiercely competitive elections. The stability of democracies is closely intertwined with the integrity of their election systems. Notably, several nations, including Estonia (Alvarez et al., 2009), Brazil, Switzerland, and Pakistan (Kamran et al., 2021), have embraced electronic voting in their electoral processes. It's worth noting that Estonia and Australia support Internet Voting, while Brazil and Pakistan employ Electronic Voting Machines (EVMs).

Despite the growing adoption of electronic voting by various nations and states, such as Switzerland (Gerlach & Gasser, 2009), Hawaii, Idaho, and Louisiana (Mia Logan, 2020), these implementations are not without their challenges. For example, in Australia, specifically in the District of New South Wales, the "iVote" system experienced compromises in more than 60 thousand votes (Safi, 2016). Germany abandoned its E-voting project due to concerns about transparency (The Constitutionality of Electronic Voting in Germany, 2013), and the Lithuanian president expressed doubts about the secrecy and security of online voting (Baltic Times, 2017). These issues all revolve around the critical factors of trust, transparency, and safety (Avgerou, 2013; Wang et al., 2017).

## 1.1 Motivation

Growing up in Brazil, a very recent democracy, we have seen a growing popularity with dangerously authoritarian figures that have been attacking the Brazilian election system and creating scepticism regarding the veracity of the latest election (Matheus Teixeira, n.d.). This unfortunately is not an event exclusive to Brazil, as we see significant political figures attacking the election system in multiple countries such as USA, Pakistan, and Others (Ingraham, 2020). Some of these countries such as Brazil utilises an electronic voting system that has started with a countrywide network of voter registration, solving the problem of double voting in the 1980s then implemented country wide in the early 2000s (Tavares, 2011). These electronic systems have been the target of the media and other actors, questioning how much we can trust our democratic future in these machines that we have no way of auditing.

Brazil has not been the only one exploring electronic voting. We can see numerous attempts of creating and implementing electronic voting systems have been done since the 1980s but very few have remained. Transparency has been one of the key factors for the disparagement of these systems. As technology has been lacking the means to provide privacy and anonymity and at the same time to an auditable system, characteristic such as privacy and anonymity are the pillars for a reliable and democratic voting system.

## 1.2. Research Objectives

This present work has the intention of creating a proposed framework for developing a Decentralized Voting Application on the Ethereum ecosystem, to facilitate the building of electronic voting elections through a blockchain system. Providing Open-Source software tools for creators to build upon and connect this technology to other established products in the Ethereum ecosystem such as MetaMask (MetaMask, n.d.) Wallet and Layer 2 Solutions such as Polygon (Poligon, n.d.). The integration with these popular technologies means a seamless adoption, easier testing, and faster development by the Open-Source community.

With these objectives this dissertation aims to answer the following Research Question:

**RQ1: “How can a blockchain-based solution implement an e-voting system that fulfils a successful electoral process?”**

With this question a secondary question come to mind:

**RQ2: “What are the technical, organizational, and political needs and requirements for implementing blockchain e-voting?”**

As discussed in the introduction e-voting systems have been around since the 1980s and have been heavily criticized and eventually lost their trust in many modern democracies, including Germany, that stated electronic voting goes “Against the principles of democracy”. This understandable criticism loses its ground when emerging technology such as blockchain that offers transparency, trust, and immutability come into play.

And to access this research question this work will enumerate the following hypotheses answering RQ1 and RQ2 respectively:

**H1. The development of a Decentralized Voting Application that includes key technical components such as a blockchain architecture, cryptography, and smart contracts, can play a crucial role in implementing a successful blockchain e-voting system.**

**H2. The issues related to political bias, legislation and regulation, public trust and perception, and government support commonly associated with an election can be addressed through the development of a Decentralized Voting application that incorporates these factors into its design, contributing to the successful implementation of a blockchain e-voting system.**

The structure of this dissertation is as follows: Chapter 1 presents the motivation, research questions, research objectives and chosen methodologies for the development of this work. Chapter 2 shows a literature review over the topics being researched. Chapter 3 does the first DSR iteration. Chapter 4 and Chapter 5 presents subsequent DSR iterations. And finally, Chapter 6 presents the conclusions and future work.

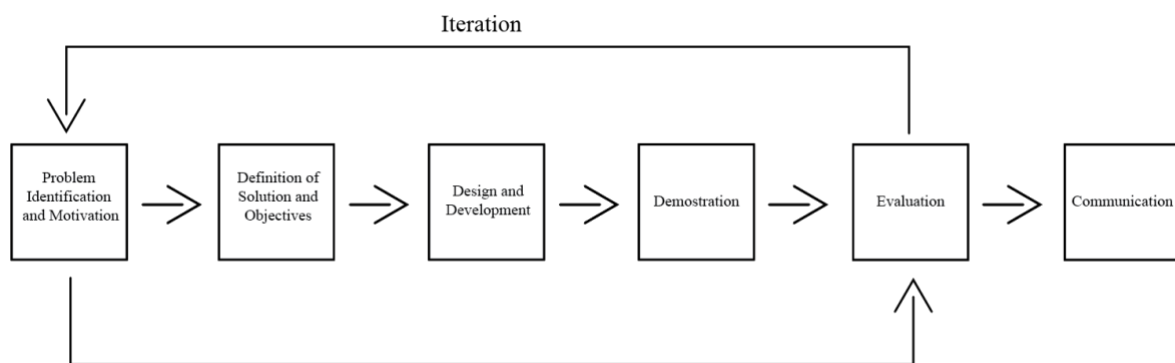
The findings of this dissertation were selected for publication and presentation at the 5th IEEE-International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA 2023) conference. This recognition highlights the significance and interest of this research for the scientific community (de Castro & Coutinho, 2023).



### 1.3 Research Methodology

The chosen methodology for this dissertation is Design Science Research (DSR). The DSR research approach is commonly implemented in the field of IT and Software development and viewed as an “Problem Solving Paradigm” (vom Brocke et al., 2020). With the objective of achieving the desired results of a Blockchain based Electronic Voting System. Design Science Research Methodology (DSRM) refers to the process of designing and researching a functional artifact or instructional material with the aim of addressing a specific issue (March & Storey, 2008). It is a comprehensive and structured approach to developing artifacts, that seeks to provide effective solutions and innovation, especially suited to software and IT related that can be also utilized in several areas, enabling the production of academic and organizational knowledge (Hevner et al., 2004).

The DSR methodology consists of six main stages: problem identification, objectives of the solution, design and development, demonstration, evaluation, and communication. Figure 1 demonstrates how the DSR Approach was conducted.



**Figure 1 – Design Science Research Methodology**

Based in figure 1, the respective DSR stages are detailed below:

1) Problem Identification and Motivation:

Current e-voting systems presents significant concerns regarding security, privacy, and trust. Traditional Elections often rely on trusted third parties to collect and tally votes, which can lead to potential collusion attacks, vote tampering, and a lack of transparency in the voting process.

2) Definition of Solution and Objectives

The proposed solution for this dissertation is to design and develop a blockchain-based electronic voting system that addresses the key challenges in security, privacy, and trust associated with traditional voting systems. By incorporating the decentralized and immutable nature of blockchain technology into the voting system.

3) Design and Development.

Develop a Decentralized Voting Solution using the latest best practices and technologies regarding the web3 and Ethereum ecosystem.

4) Demonstration:

Create a mock-up app and test the system with mock up elections in multiple scenarios.

5) Evaluation:

Review System features, functionality, and scalability. Compare system to past proposals and check improvements through DSR Iteration

6) Communication:

Submission of scientific articles.

Table 1 shows the proposed guidelines of Design Science Research, proposed by Hevner's in Design Science in Information Systems Research (vom Brocke et al., 2020). This Guidelines ensure the research is rigorous and contributes to a meaningful proposal to the field. Table 2 demonstrates the evaluation questions used during the DSR evaluation.

**Table 1 - Design Science Research Guidelines**

<b>Guideline 1: Design as an Artifact</b>
The artifact of this Dissertation is a Decentralized Voting System for electronic voting.
<b>Guideline 2: Problem Definition</b>
A need for trustable and reliable online voting solutions
<b>Guideline 3: Design Evaluation</b>
The development of a Mock-up APP for testing the developed Features
<b>Guideline 4: Research Contributions</b>
A open source system that can be expanded upon developers and organizations needs
<b>Guideline 5: Research Rigor</b>
The fundamental practices of DSR were used to develop the artifact, also applying the latest research and developed technologies in the blockchain field on the voting system.
<b>Guideline 6: Design as a Search Process</b>
The result of this research comes from the aggregation of researched proposals in the Literature Review section and application of new technologies from the Pre-Solution State section.
<b>Guideline 7: Communication of Research</b>
This research intends to be submitted to a journal/conference with a high reputation in the scientific community.

Key Evaluation questions:

Table 2 – Key Evaluation Questions

<b>What are the positive aspects of the proposed voting System?</b>
<b>What are the negative aspects of the proposed voting System?</b>
<b>Based on the Negative aspects what improvements can be made for the next iteration?</b>

However, to gather data for this dissertation a Systematic Literature Review (SLR) is utilised and available in Chapter 2. The SLR research method aims to identify, evaluate, and synthesize all available evidence related to a specific research question (Kitchenham, 2004).

To acquire this data, a well-defined protocol is established for the selection of relevant articles or other research artifacts from scientific databases. This protocol includes the use of a search string and the application of filters, with a focus on keywords within the Abstract and Title of the articles. The process of article filtering is presented in Table 3.

Table 3- Number of extracted Surveys

<b>Data Base</b>	<b>Search String</b>	<b>Filter</b>	<b>Total</b>
<b>IEEE</b>	<b>10</b>	<b>5</b>	5
<b>ACM</b>	<b>12</b>	<b>2</b>	1
<b>Scopus</b>	<b>67</b>	<b>8</b>	2
<b>TOTAL</b>			5

Subsequently, the selected articles undergo a comprehensive review. In this case, various information points are extracted, including details concerning tools and security failures. Additionally, a concise summary of each article is generated, highlighting its primary insights and findings. A comprehensive explanation of the SLR approach utilized in this dissertation, along with the specifics of data collection, keyword selection, and the steps involved in the reporting process, is elaborated upon in Chapter 2 Introduction and 2.1 Outlying the Literature Review.

## CHAPTER 2

### Literature Review

A Systematic literature review is an approach to conduct a clear and rigorous literature review, according to (Kitchenham, n.d.) “A systematic literature review is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest.”

To address the purpose of this dissertation, a Systematic literature review was conducted based on the guidelines of the authors Webster & Watson,2022. This review aims to analyse different state of the art approaches for building a blockchain voting system and bring to light the most common tools used for building these systems and current issues and challenges faced in the process. Figure 2 demonstrates the structure for this review.

1. Outlining Systematic Literature Review		2. Conducting Systematic Literature Review		3. Reporting the Review
<b>Identification of the why of this review:</b> Gather information about current blockchain e-voting applications and proposals.	>	<b>Selecting our articles:</b> Filtering, Ranking and Accessing content.	>	<b>Reporting:</b> Discussion about the findings and Conclusion
<b>Objective of the review:</b> Organize information about current state of the art proposals and e-voting applications.		Analysing selected studies and Extracting Data		
Research Protocol				

**Figure 2 - SLR Structure**

## 2.1. Outlining the Literature review

In initial research with no defined keywords, a defined line in articles that explored blockchain technologies in e-voting applications in technical aspects and other articles that explored e-voting and its implications on society. In this Review we will focus on the first alternative, research the current applications of e-voting utilising blockchain technologies and their proposed technical solutions and Surveys.

To obtain information for this review we selected these electronic repositories:

- IEEE Xplore Digital Library (<https://ieeexplore.ieee.org/Xplore/home.jsp>);
- SpringerLink (<https://link.springer.com>);
- ACM (<https://dl.acm.org>);
- Wiley (<https://onlinelibrary.wiley.com>);

This review included only English articles and surveys that were published on Journals or Scientific Magazines or Conferences Proceedings, no date filter was used.

This review was based on 3 Main Stages: Getting to know current literature as a whole and getting a sense of the best keywords, then, asserting our keywords and removing duplicate articles and filter articles by quality. As multiple repositories have different search approaches, we built slightly different search strings and methods to get to the same result on each database utilising operators such as “AND” and “OR” our main keywords were “Blockchain” AND “E-voting” this system resulted in applying 4 filters to all electronic repositories.

To achieve a broader sample of articles regarding blockchain and Electronic Voting, the keywords were expanded into “Blockchain” AND (“E-Voting” OR “Electronic Voting”). This provided more articles regarding the applications of blockchain in voting systems as “E-Voting” can be referred as mobile voting or remote voting and the goal of this review is to access all papers regarding any form of Blockchain and Electronic voting,

The First filter was applying our keywords to the Title, Abstract and Author Keywords, the second filter was removing our duplicated articles and proceedings. Our Third Filter was removing Articles that did not belong to Higher Ranking conferences/Journals. For this process we utilised 2 websites: (Scimago, n.d.) and (Conference Ranks, n.d.) which provided rankings for Journals and Conferences. For Conferences, only Articles belonging to A, B, A1, A2, 2 ranks of ERA and Qualis rankings were accepted and only journals from Q1 and Q2 rankings were accepted. Our final filter and last stage of this article selection was reading the Abstract and Introduction of these articles and evaluating if the article proposed a blockchain based solution to an e-voting election process. Articles and proceedings that addressed registration of voters but did not address the voting process were not included in this review.

## 2.2. Conducting the review

As discussed previously our SLR is made of four filters of which all the articles must go through to be selected. Table 4 Demonstrates how many articles were filtered through each phase of the review.

Table 4 - SLR Filters

	No Filter	1st Filter	2nd Filter	3rd Filter	4th Filter
<b>IEEE</b>	510	288	128	33	19
<b>ACM</b>	3103	76	23	8	4
<b>SpringerLink</b>	1538	98	62	34	5
<b>Wiley</b>	676	28	14	12	3
<b>Total</b>	6306	672	227	51	30

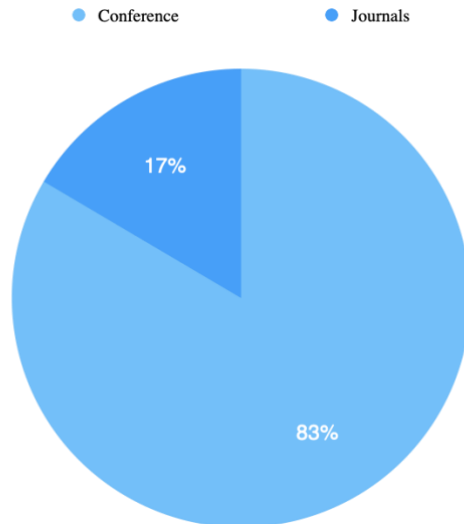
The 1st Filter had the goal of excluding articles that would be related to only one area such as only blockchain or only e-voting, without this filter there was multiple articles that would explore e-voting or blockchain entirely separately and would not contribute to this review. The 2nd Filter just had the purpose of eliminating duplicated articles, the same filter was applied when selection the final round of articles from all Databases together. On the 3rd filter, articles were ranked by Qualis and ERA rankings according to our threshold and only 51 were left. On the 4th and final filter we were accessing the Abstract and Introduction of these papers Only 30 Articles were chosen to participate in this review. These articles proposed some sort of framework or solution to blockchain based e-voting.

### 2.2.1 Information Extraction Process:

After the final selection of articles and filtering we carried out an analysis of each article. For each article we extracted the subsequent information: year the article was published, article domain, conference and Journal Ranking, tools utilised in the development process if available, type of blockchain applied in the proposal, and most importantly what key technical, organizational, and other considerations these proposals had for an effective a blockchain e-voting system.

### 2.2.2. Sample characteristics:

The sample made from 30 articles, that are proposals and frameworks for blockchain based e-voting, reviews and other forms of contributions were not accessed on this review. Only Journal and Conference proceedings were accepted, and we can see the distribution in figure 3



**Figure 3 - Distribution of Articles by Publication Type**

The main source of articles for this review came from Journals, Table 5 and Table 6 demonstrates the distribution of the final article selection by their conference Ranking and Journal publication rank respectively.

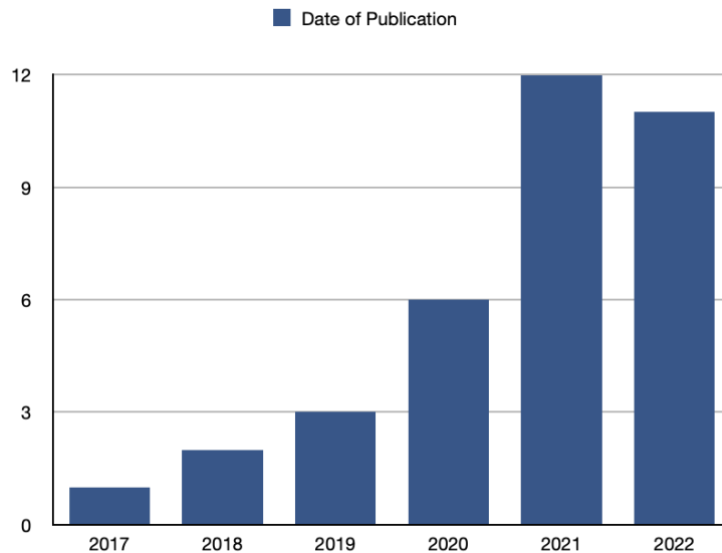
**Table 5 - Conference Ranking**

	Conference Rank	Total
<b>ERA</b>	A	2
	B	7
<b>QUALIS</b>	A1	3
	A2	3
	B1	2
	B2	1
<b>Total</b>		17

**Table 6 - Journal Ranking**

Journal Rank	Total
<b>Q1</b>	9
<b>Q2</b>	4
<b>Total</b>	13

As stated previously, no date filters were used in this Systematic Literature review, to discuss the Evolution of Blockchain E-voting protocols Figure 4 demonstrates the distribution by year of publication. We can see that the interval goes from 2017 to 2022, as no date filter has been included in this systematic literature review, the distribution show how there is an increase in interest into the topic of “Blockchain” and “E-voting”.



**Figure 4 - Distribution of Articles by Year**

Table 7 demonstrates a comparison of the evaluation, tools, and limitations of each proposal of an e-voting system found in IEEE database. To build a trustworthy and reliable framework we will analyse the technical aspects and tools of each proposal and check if they accomplished the research objective, thereafter, extract the limitations and future work of each proposal for further information.

**Table 7 - Relative Comparison of “Blockchain” and “E-voting” Proposals IEEE**

References	Objectives	Evaluation	Tools	Limitation/Future Work
(Lyu et al., 2019)	Trustable e-voting in a Ethereum smart contract	Time analysis in a local server, gas prices in Ethereum test net	Ethereum, Ethereum Private Net, Solidity	Destruction by malicious voters in key generation stage, too expensive to scale
(Zhu et al., 2022)	Two-layer Blockchain Architecture for a multi-district election	Eligibility, Correctness, Time analyses on Hyperledger Fabric	Hyperledger Fabric,	Does not go over security
(Huang, He, Obaidat, et al., 2022)	Solve the abstention problem through e blockchain e-voting	Computational Cost, Time Analysis,	NA	Election Authority, Decentralization
(X. Yang et al., 2020)	High level security and privacy,	Computational Cost, Time Analysis,	SGX, Ethereum	Side channel Attacks
(Panja et al., 2020)	Implement a Borda count blockchain e-voting system	Computational Cost, Time Analysis, Gas cost,	Ethereum, Solidity	Expand into other form of counting votes
(Banawane et al., 2022)	Novel Approach on blockchain e-voting	NA	Python, Django, SQL	Security
(Poniszewska-Maranda et al., 2022)	Private Blockchain e-voting application	NA	Hyperledger Fabric, Typescript.	NA



			NodeJs, Vue.JS	
(Zaghloul et al., 2021)	Remote E-voting	Computational Cost, Time Analysis,	Solidity, Swift, MetaMask	Scalability issues
(Kamran et al., 2021)	Meet anonymity and universal verifiability	NA	NA	Englobe miners in the system, improve scalability
(Nguyen & Thai, 2022)	Provide lightweight cryptography for a secure protocol	Computational Cost, Time Analysis,	NA	Can improve parallel processing
(Carcia et al., 2021)	Implement Ring Signature	Computational Cost, Time Analysis,	MatLab, Hyperledger Fabric, Docker, Hyperledger Caliper	Cannot unlink ballot from voter
(Shahzad & Crowcroft, 2019)	What activities formulate electronic Voting	Evaluates Hash Algorithms	NA	Scalability
(M. Li et al., 2022)	Security and performance Blockchain platform	Computational Cost, Time Analysis, Gas and ETH Cost	Solidity, Ethereum	Scalability,
(Hjalmarsson et al., 2018)	Blockchain As a Service for e-voting	Transactions per Second	Rust, Quorum, GO, C, JavaScript,	NA
(Farooq et al., 2022)	DApp for E-Voting with flexible consensus	Latency, Time Analysis, Response Time.	Ethereum, Solidity	Prevents 51% attack
(Khandelwal, 2019)	A safe e-voting blockchain System	NA	Ethereum, Solidity,	Security and traceability issues
(Kumar, 2021)	Secure and reliable E-voting System with PoW Consensus	Security Evaluation	Python, Flask,	Integrate the Ethereum network to minimize power consumption
(Fernandes et al., 2021)	Voter anonymity and vote verifiability	NA	Ethereum, Remix, MetaMask, Rust,	NA
(Y. Yang et al., 2021)	Robust Distributed E-Voting platform.	Performance Analysis, Security Analysis, Cost Analysis	Charm, Python, Ethereum,	Include coercion resistant electronic voting.

In this section of our Literature review we will cover other work done by other researchers in regards of blockchain e-voting applications and frameworks. This section does not overview Surveys, only implementations and proposed Solutions.

(Lyu et al., 2019) write about an E-voting system based on Smart Contracts on the Ethereum blockchain, the objective is to assure privacy and that no actor can tally the votes before the election is finished, uses threshold encryption for that purpose and for ensuring the voter anonymity it uses a linkable ring signature, threshold encryption works in this case by a pair of public/secret keys, where the public key is known by all parties and the secret key is “given” only when a certain percentage of the parties involved complete their vote. The linkable ring signature allows to identify the anonymous voter. the system provides the voter to generate a signature from a list of public keys. This system allows the voter to be anonymous during the voting process (he is the only one who knows who generated the signature) but its drawbacks are that it really is only efficient with more people using the voting system. however, the system has security failure that allows destruction of voters in the key generation phase. The takeaways for this paper are the linkable ring signature that provides some anonymity in the Ethereum environment.

(L. Zhu et al., 2019) propose a blockchain-based voting system for Multi-District Elections using a two-layer blockchain architecture to improve efficiency. The author recreates the voting system of multiple districts system as on a Multi district election is not counting each vote and yes counting the results of each district election. The 2-layer solution tries to balance the cost of an elections providing the count and voting of each district election on a separate layer, and then sending these results back to be processed on layer 1. This system increases efficiency and provides a way of achieving more scalable elections given an appropriate number of nodes. The 2-layer solution is a great way to make blockchain elections in the Ethereum ecosystem more affordable and scalable this solution is going to be added in further iterations of our DSR process.

(Huang, He, Chen, et al., 2022) proposes a Blockchain-based Self-Talling voting Protocol that aims to solve the issues of Privacy, duplicated voting, and abstention, three common problems regarding self-talling protocols in voting. Utilises a Homomorphic encryption on the ballot content and proposes a proof algorithm to ensure that the ballot has valid content, that will not be disclosed during validation. The Autor suggests two main cryptographic techniques, the homomorphic encryption and the Shamir threshold secret sharing. Where the homomorphic encryption allows mathematical operations to be performed on the encrypted data without the need of depiction first. Then the ballot content is separated in different secrets to apply Shamir Secret Sharing. This blinds the result of the homomorphic encryption. The system was tested on Hyperledger fabric and shows that it could be applied to different voting scenarios, that said the performance linearly degrades with the addition of candidates.

(X. Yang et al., 2021) propose a distributed e-voting system that counter the adaptative and aborting issue, utilising blockchain and Intel SGX for an elevated level of security and privacy, Intel SGX is a hardware-based security technology that isolates the execution of code. Not letting malicious actors execute code outside of that enclave, this solves the Adaptative issue of a voter starting the tallying procedure before the end of the election by holding all submission within the SGX enclave until the deadline has passed. In case of the abortive issue, that means, the voter did not submit their vote. The SGX can cast a neutral vote for that voter, Scaling is still an issue with this system as the time complexity scales quadratically with the number of submissions. The idea of publishing the voting fingerprints for the verification of votes will be one of our takeaways. That said is not stated in the paper if the absences are being kept track of, as it is an important metric to have.

(Panja & Roy, 2021) propose a Blockchain Borda count voting system, Borda count is a system where voters are asked to rank the candidates or options in order of preference. Each candidate or option is then assigned a certain number of points based on its rank, with the highest-ranked candidate receiving the most points and the lowest-ranked candidate receiving the least points. This article proposes a smart contract system for implementing decentralized Borda count voting on the Ethereum blockchain. The authors describe the features and design of the smart contract system, including the use of a scheme to ensure voter anonymity and a dispute resolution mechanism to handle challenges to the voting results. They also present the results of an experimental evaluation of the smart contract system, which showed that it can effectively and securely conduct Borda count voting with many voters and candidates. The authors conclude by discussing the potential applications and future work for their smart contract system in the context of decentralized decision-making.

(Banawane et al., 2022) has the objective of proposing a Novel approach for blockchain e-voting system, the author has into account security and cost, and does not test performance or evaluates the scalability of the solution, the paper does go into more detail into the technologies used such as Python Django, SQLite, and MongoDB, hashing and blockchain code will also be written in Python. Banawane Sets as requirements for his system: Authentication, the voters have to be registered to vote and the registration will not be supported by the system. Anonymity. The voters identity should stay hidden and should not be any links between voters and its identity after the election results. Accuracy and verifiability, the system should count votes accordingly and the algorithm should support the verification of votes after the end of the election.

(Poniszewska-Maranda et al., 2022) propose a decentralised e-voting system on the Hyperledger fabric, the solution approaches not only the blockchain challenge but also the middleware and user interface, the solution is composed by two smart contracts that provide the logic for voting. The complete stack is made of the Hyperledger fabric, a rest backend made in typescript and NodeJS, and front end made in Vue.js. Upon voting the voter receives a transaction number that can be verified against the application. There are no evaluations of time complexity, scalability, or computational costs on this paper.

(Zaghloul et al., 2021) propose a blockchain-based e-voting platform for mobile devices where voter verifiability is based on randomly generated values and does not provide any information on how a voter may have voted even if the number is shared. All computation is performed outside of the blockchain, that acts only as a public information. The article also provides proof on how the application is secure and protected against double voting and that it preserves anonymity. The application is deployed over Ropsten Ethereum TestNet and all Smart contracts are developed on Solidity language, the communication between the Platform and the Ethereum balance was done with MetaMask Wallet

(Nguyen & Thai, 2022) demonstrate with a proof of concept a blockchain-based platform for remote e-voting, the Platform aims to maintain privacy and anonymity for voters with a zero-knowledge proof protocol for the authentication and validation process. The article does not specify tools or languages used in the process. The authors utilize Zero-knowledge proof protocols to maintain privacy in membership and input validation, verifying the eligibility of users and validate their votes while remaining anonymous. Zero-knowledge proof protocols allow a party to prove to another actor that they know a piece of information without revealing the information itself. This approach where a user can verify their own vote is a feature this work tends to propose.

(Carcia et al., 2021) propose a self-counting blockchain system to ensure voter privacy and anonymity when storing ballots in the blockchain. The paper aims only in this mechanism of the blockchain and does not access the other parts of a voting system such as registration. A security Analysis is made alongside the evaluation of performance and scalability. The solution average voting time has a linear increase linearly with voting count. To ensure privacy and security store voters information the author uses a blind signature protocol, where the ballot is split into two parts so it's not possible to access the ballot with only one of the parts. The blind signature protocol is also used so it's possible to verify your votes without the need for breaching voters privacy.

(Shahzad & Crowcroft, 2019) address the issues that most common countries face when dealing with an electronic voting election, highlighting recent democracies such as Brazil, India, Pakistan, and Bangladesh. The article exposes the nature of current mistrust in these e-voting systems, current mistrust issues are: Rigging, Duplicate Vote Casting, Influence over pooling Staff, Unsupervised Vote counting, Lack of audit and lack of interest by the people. The article also goes over current state of the art blockchain concepts such as Proofs and Hashing Algorithms and provides an overview on which would be the better suited solution for an electronic voting system. It does not provide an insight on voter registration and assumes that the voter will register with its identity card or any form of identification accepted by the government.

(Y. Li et al., 2022) Proposes an E-voting protocol utilizes a homomorphic time-lock puzzle to take care not only of the self-tallying but as also the privacy of the voting ballots. The article also provides a mathematical explanation of the implementation of the prototype as also an evaluation of the time consumption of its operations and their respective gas costs. The author defines one of the privacy features of the system as a homomorphic time lock puzzle, a feature that allows operations to be done on the encrypted data without revealing information. Another introduction by the author is Time-Bounded Privacy, defined by a basic security requirement where the content of ballot is kept secret against all adversaries. The project also uses Ethereum Virtual machine where a smart contract was written to run necessary functionalities. This smart contract approach will be essential in this project as our goal is to create a DAO based on the Ethereum ecosystem.

(Hjalmarsson et al., 2018) propose a E-Voting-As-A-Service with a permissioned blockchain system, the paper reviews current proposals for blockchain based e-voting and makes the design considerations to base its system. The paper utilises a Ethereum based permissioned PoA (Proof of Authority) blockchain, a smart contract as the election logic. The system allows the voter to verify his vote providing verifiability but does not allow traceability of votes. PoA consensus delivers faster transactions based on its identity-based mechanism and the author states the use of Go-Ethereum or Geth, the original Ethereum implementation as the best choice as it does not allow downtime.

Similarly to Table 7, Table 8 accesses the articles and proceedings found in the ACM repository. We conducted the same analysis as in the IEEE database to retrieve information regarding Research Objective, how was made the evaluation of the solution, the tolls used in the process and lastly what limitations that proposal encountered.

**Table 8 - Relative Comparison of “Blockchain” and “E-voting” Proposals ACM**

<b>References</b>	<b>Objectives</b>	<b>Evaluation</b>	<b>Tools</b>	<b>Limitation/Future Work</b>
(Killer et al., 2020)	Cast-As-Intended Blockchain Voting System	Time and Cost analysis.	Ethereum, JSON, Rust	Security Limitations
(Pawlak & Poniszewska-Marańda, 2019)	Intelligent agents and multi-agent system concept for Auditable Blockchain Voting System.	NA	Ethereum	NA

Similarly, the IEEE section, in this section of our Literature review will cover other work done by other researchers in regards of blockchain e-voting applications and frameworks. This Section does not overview Surveys, only implementations and proposed Solutions.

(Killer et al., 2020) Implements a Cast-as-Intended verifiable Blockchain Voting System, the author designs the system with the context of Elections in Switzerland, based on a few assumptions such as that the system will operate only on secured infrastructure controlled by the election authorities, the voters would receive the execution binary to run the voting. The vote then would be encrypted with zero-knowledge proof before the submission, Thanks to the proof the voter could always see if his vote was tampered with or not inserted by a malicious binary, the computational cost is linear in relation to the scale of elections, but the system is more centralized than most proposals.

(Pawlak & Poniszewska-Marańda, 2019) Proposes an Intelligent and Multi Agent Approach for an Blockchain e -voting System, the intelligent agent would be implemented with Smart Contracts, where the agent would oversee and check for tampering in the election. The intelligent agent approach made by the author consist of a computer program located in a certain environment, The goal is for these agents to take autonomous decisions and talk to other agents The author states that these agents should operate by these three interactions: Cooperation, Coordination, Negotiation. The agents should operate in synchrony and resolve most issues regarding the election. The author defines five types of actors (Agents) that will be involved in the system. Each actor has functions that provide its role in the system. There is also a difference of authorization between agents, some agents have more permissions than others, although this solution seems extremely complex on the outside, these intelligent agents function as smart contracts deployed on the blockchain, reacting to the behaviours and inputs from third parties and other agents. This is the approach this work intends to take for its system, that said the development of multiple smart contracts could be too expensive to justify the implementation.

**Table 9 – Relative Comparison of “Blockchain” and “E-voting” Proposals SpringerLink**

<b>Reference</b>	<b>Objective</b>	<b>Evaluation</b>	<b>Tools</b>	<b>Limitation/Future Work</b>
(Chouhan & Arora, 2022)	Blockchain Counting Mechanism using Secret Sharing	Performance Analysis, Security Testing.	Hyperledger Fabric,	Separate Application for Each entity.
(Y. Zhou et al., 2020)	Improves FOO Voting System	Time analyses and cost analyses, Security testing	Hyperledger Fabric,	NA
(Yi, 2019)	Synchronized voting model, User Credential model and Withdraw Model.	Security and Anonymity	Python	NA
(S. Zhang et al., 2020)	Scalable and Robust Distributed Voting System	Computational Cost, Security Testing	NA	Client-side Tampering

(Y. Zhou et al., 2020) Proposes an improved FOO voting protocol, where traditional FOO voting adopted centralized counting, an improved FOO Blockchain system Where the centralized authority is changed for smart contracts. FOO Voting (Fujioka, Okamoto, and Ohta) its proposal for e-voting system that uses a centralized third party and blind signature to ensure voter privacy. By having a centralized third party responsible for vote counting, this system did not have a lot of trust between voters and had multiple weaknesses, with the author proposal of substituting the Trusted Third Party by Smart Contracts give this voting a protocol a much more robust approach to voting. This is also the approach this work intends to propose in the developments of a DAO. The author utilises Hyperledger fabric for blockchain development.

(Chouhan & Arora, 2022) proposal is a chain-code implementation of a secure and verifiable voting system utilising a partitioning scheme, chain-code is the equivalent implementation of a smart contract in the Hyperledger framework. Providing a way to run code on top of the blockchain implementation. The partitioning scheme highlighted by the author is based on the splitting of the vote data, making it impossible for the vote to be reconstructed without all the data being pieced together. This implementation is similar to other reviewed by this work. The system highest throughput is 200-400 transactions per second and optimal latency of 18 seconds for vote distribution. And 4.5s for the vote count.

(Yi, 2019; S. Zhang et al., 2020) Proposes a synchronized model of voting records based on a distributed ledger paired with a user credential model based on Elliptic Curve Cryptography (ECC). ECC is a public key encryption that provides the same level of security as RSA algorithms with significant smaller key sizes. The author also proposes a withdraw model that allows the voter to change their vote before the preset deadline. This work does not intend to implement this feature as having the option to change your secret vote means a window for coercion and manipulation to ensure voters are voting for a certain party. The author developed the solution on Linux Systems and Python. The evaluation considers the system secure with the implementation of the ECC Encryption. That said the vote is not anonymous currently.

(S. Zhang et al., 2020) Analyses previous work and identifies a series of requirements for a voting system, the author develops a system to achieve the outcome based on his identified needs: Scalability, Privacy Enhancement, Universal verifiability, End-To-End verifiability, Vote and Go, Affordability, Fairness, Robustness. After the author evaluated a series of related projects and classified them based on these requirements. Not finding a single proposal that achieved all the requirements. The author achieved this result with a Blind signature authentication and a multi-platform proposal, and more cost-effective with the combination of the counting Bloom filter and the Merkle hash tree.

## 2.3 – Pre-Solution State

This section of this dissertation aims to access the current solution state of blockchain e-voting. For this purpose, we will access a selection of surveys regarding the blockchain e-voting opportunities, challenges, and overviews current solutions available in the market for building a blockchain e-voting system.

### 2.3.1 Surveying Literature:

To gather information around current Opportunities and challenges this section accesses a selection of surveys around blockchain e-voting. This selection contains 5 surveys related to blockchain e-voting.

To gather this selection of surveys this section outlines a similar protocol to the Systematic Literature Review, where a selection of scientific databases was searched with the search query:

“Blockchain” AND “E-voting” AND “Survey”

To get to these results, the string was applied to the Databases, to arrive at a more focused group of articles, a filter for Title and Abstract was applied. After this filter all articles had their introduction and abstract read and evaluated. If the articles did not offer a survey regarding blockchain voting the survey was excluded from the results.

The Databases and Results of this search are the ones stated in table 10:

**Table 10 - Number of extracted Surveys**

Data Base	Search String	Filter	Total
IEEE	10	5	5
ACM	12	2	1
Scopus	67	8	2
<b>TOTAL</b>			5

**Table 11 - Relative Comparison of Extracted Surveys**

Survey	Objective	Contributions	Limitations and Open Issues
(Vivek et al., 2020)	Evaluate proposals utilising the Hyperledger Sawtooth Framework	Analyses key design and encryption method for multiple proposals utilising Hyperledger Sawtooth framework	Found scalability issues in multiple proposals
(Ohammah et al., 2022)	Compares various proposals for blockchain Voting	Explores current deployed solutions for blockchain voting	None of the proposals have used a public Blockchain. Performance Issues in most reviewed applications.
(Al-Maaitah et al., 2021)	Evaluates different blockchain voting applications	Explores implemented and “Draft paper”	N/A



		Proposals, explores security deeply	
(Heinl et al., 2023)	Provide an overview of the historical and recent developments in the area	Evaluates each proposal carefully with a clear set of rules	Missing tabular data
(Shanthinii et al., 2023)	Compare Solution to Current Blockchain E-voting Frameworks	Recognized standards for voting and most recognized settings for blockchain based voting	N/A

After the selection of articles was complete the extraction of information was made with the objective to extract the following information from the surveys: The Survey objective, the survey main contribution and limitations and Open issues highlighted during the survey. Table 11 shows as comprehensive view on the surveys reviewed.

The author (Vivek et al., 2020) analyses multiple paper proposals and existing e-voting system, such Estonia I-Voting, and its translation to a Blockchain Architecture. Goes over requirements for a blockchain voting system and goes over frameworks in Ethereum and Hyperledger Sawtooth.

(Al-Maaitah et al., 2021) Reviews draft paper blockchain solutions, categorizing then according to Architecture and Design, Security and Their Limitations. States how none of the reviewed proposals have motivated the management or maintenance of a blockchain and found mostly performance issues in the reviewed papers.

(Ohammah et al., 2022) Survey highlights the impacts of low voter turnouts in multiple occasions and presents current e-voting solutions in the market, the review goes from the system capabilities to tools used. The survey concludes that scalability is still an issue for most blockchain voting systems.

(Heinl et al., 2023) Not only goes over technical requirements or overview of current proposals, but also provides an overview of historical and recent events on blockchain electronic voting. Analysed both the legal and the technical requirements for internet voting.

(Shanthinii et al., 2023) Conducts a survey and analysis of voting system and the Ethereum ecosystem, classifying the general necessities for a protected voting system.

### **2.3.2 Current market solutions:**

(Voatz, 2023) A US company that provides a blockchain-based mobile voting platform. The platform is designed to provide a secure, convenient, and accessible way for voters to cast their votes remotely. The Voatz platform uses a combination of biometric authentication and blockchain technology to ensure the integrity and privacy of each vote. The system has been used in US state elections such as Oregon, Utah, and Arizona, and has been used primarily by military personnel stationed abroad.

(Agora, 2023) Created in coordination with the Swiss Federal Institute of Technology Agora is a company that provides a secure and transparent platform for digital voting. Agora's e-voting solution uses an inhouse developed blockchain protocol and application. The system is designed to provide a high degree of transparency and security, as each vote is recorded immutably on the blockchain and can be audited by anyone with access to the network. Agora's e-voting platform was used in the 2018 presidential election in Sierra Leone, marking the first time blockchain technology was used in a national election. The system was designed to provide transparency and security in a country where traditional voting methods have been subject to fraud and manipulation.

(Kaspersky, n.d.) Kaspersky Polys project creates not only a blockchain based online voting platform but also provides enterprise solutions and voting machines. Polys supports different use cases such as university election for students, digital voting for political parties and digital voting for trade unions.

## 2.4 – Discussion and Conclusion of Review

### 2.4.1 – Tools and Frameworks.

Authors and researchers have chosen different tools to build their proposals for a blockchain based electronic voting system, it is possible to see in table 12 the distribution of tools based on proposals:

Tool	Use
Ethereum	45%
Hyperledger Fabric	18%
MetaMask	15%
Pycharm	3%
Django	3%
Geth	2%
MATLAB	2%
AmazonEC2	2%
Truffle	2%
Flask	2%
Remix	2%
Quorum	2%
Hyperledger Composer	2%

Table 12 - Distribution of Tools of Stake.

There is a clear preference from the authors to use the Ethereum Environment in their proposals, not only the Actual Ethereum environment but also other forms of implementation such as Geth (Go Ethereum). Ethereum is an open-Source decentralized blockchain that offers the smart-contract Functionality, as Ether (ETH) being the native cryptocurrency of its ecosystem, it is design to work with complete transparency and offers a satisfactory level of safety and robustness as it is one of the most widely used Blockchains today. There are drawbacks to this ecosystem, Ethereum has been subject to hard Forks and due to its fast-changing development it can be unstable. Ethereum moved to PoS (Proof of Stake) consensus with the goal to improve its scalability and reduce its energy

consumption by 99.95% (Ethereum Foundation, 2023).

Geth or Go-Ethereum is a Command Line Interface (CLI) form of implementing a node of the Ethereum ecosystem, where is possible to interact with the Ethereum network and test smart contracts before deploying them to the Ethereum blockchain. Ethereum offers the Smart contract functionality, a feature used in multiple proposals such as (Lyu et al., 2019; Panja et al., 2020), Smart contracts offer the possibility of coding the rules and automatically execute, verify, and enforce or perform the rules of a contract. In the context of Ethereum a smart contract is a program that runs on the EVM (Ethereum virtual Machine), Smart contracts can be written in a variety of languages, but most common implementations are made in Solidity.

Hyperledger Fabric is an opensource blockchain project by the Linux Foundation, it is an component based system that allow consensus, and other services to be easily integrated in a blockchain environment, it has an open Smart Contract model and supports languages such as Go, Java and JavaScript It is currently used in multiple sectors such as IoT, Healthcare, Finance etc, but it doesn't come without disadvantages, it has a fairly complex architecture and it's not a Fault Tolerant Network (Poniszewska-Maranda et al., 2022).

### **2.4.2 Conclusion of Review.**

Today Blockchain Technology faces a new set of challenges, not only relating to e-voting systems, but for blockchain to assume a primary role in our society it needs to achieve safety, reliability, and scalability, and to gain the trust of the people to be applied in our election systems. The challenges facing Blockchain voting system are also Anonymity, Privacy and Safety.

**Scalability:** Currently Ethereum stands in a maximum of 15 transactions per second in its ecosystem, even though this number does not seem low, it does not come close to the scalability of current payment networks and other centralized systems, and it is not viable to run a country-wide election for a country with a big population such as USA or Brazil with such limitations. This issue has its workarounds with other types of consensus, but it is not currently supported by the Ethereum ecosystem.

**Privacy and Safety:** Balancing Voter privacy and Transparency in an election process is no easy task, especially talking about a decentralized system such as blockchain. This issue has been circumvented by the authors (Nguyen & Thai, 2022) and (T. Zhang et al., 2022). But it does not come without its drawbacks in scalability and efficiency.

**Security:** Security is one of the main challenges in implementing an electronic voting system, in our review this was one of the main points that the authors focused when proposing a system, and although blockchain may have transparency there were plenty of proposals where the security analysis showed vulnerabilities, such as Transaction Malleability or other exploits to tamper votes.

Blockchain has achieved considerable progress in the E-Voting Sector, and this review demonstrates a trend in the increasing number of papers submitted with these keywords. For the development of this dissertation this work will focus on the highlighted issues found during this review such as scalability, security and privacy and have in mind the most common tools used by authors and developers in the ecosystem.

### **2.4.3 Conclusion And Research Proposal**

This section has provided an overview of several blockchain based Electronic voting platforms and companies that are currently active, including Voatz, Agora, and Polys<sup>1</sup>. These platforms use various combinations of voter authentication and a variety of blockchain infrastructure to provide secure and transparent voting systems.

---

<sup>1</sup> Voatz: <https://voatz.com/>, Agora: <https://www.agora.vote/>, Polys: <https://polys.me/>

Furthermore, the pre solution state has examined several research surveys that have analysed multiple blockchain-based e-voting systems. (Vivek et al. 2020) analysed multiple paper proposals and existing e-voting systems, such as Estonia I-Voting, and translated them to a blockchain architecture. (Al-Maaitah et al. 2021) reviewed draft paper blockchain solutions, categorizing them according to architecture and design, security, and their limitations. (Ohammah et al. 2022) surveyed current e-voting solutions in the market, highlighting the impacts of low voter turnouts. Finally, (Heinl et al. 2023) provided an overview of both legal and technical requirements for internet voting. (Shanthinii et al. 2023) conducted a survey and analysis of voting systems on the Ethereum ecosystem.

This section has gathered the most current research of blockchain voting technology, evaluating transparency, security, accessibility and respective tools and technologies used in these solutions. There are clearly promising platforms currently available in the market, however further research is needed to address issues such as scalability and security.

As stated on table 12 on the Literature review, 46% of E-voting proposals have been done in the Ethereum ecosystem, and many others utilize technologies such as Geth or other variants of the Ethereum implementation. Tools such as MetaMask wallet are also fully compatible with the Ethereum ecosystem and other frameworks that are based on the Ethereum technology.

Therefore, one of the contributions this work tends to propose is a Decentralized Voting Application for building voting systems in the Ethereum ecosystem.

## 1<sup>st</sup> DSR Iteration Proposal and Evaluation

The proposed blockchain-based e-voting system was the focus of an iterative DSR process that included the Development of a mock-up App to test the application. Each iteration produced a variety of benchmarks and tests with validating, consolidating, and improving the e-voting system. The iteration method used in this dissertation, similar to an Agile method with frequent feedback, this methodology served as a mean to build an increasingly complex solution and overcome problems and challenges shown in the evaluation phase.

This section will go over the development of the first iteration of the voting system. In this first iteration this section will go over the Tech Stack, blockchain architecture and other detailed concepts. This information will not be repeated in other iterations if there is not a change or addition in these sections.

### 3.1 Technology Stack

For the development of this Decentralized Application there is a few tools that have been used, the implementation was done using a MacBook with apple silicon, NPM, Truffle Framework, MetaMask, Solidity Toolkit and Ganache.

NPM is package manager that manages, installs, updates or uninstalls the node.js packages in an application. It is a command line-based tool. It operates in two modes: local mode and global mode. In global mode all node.js application is affected and in local mode only particular directory of an application gets affected (NPM, n.d.).

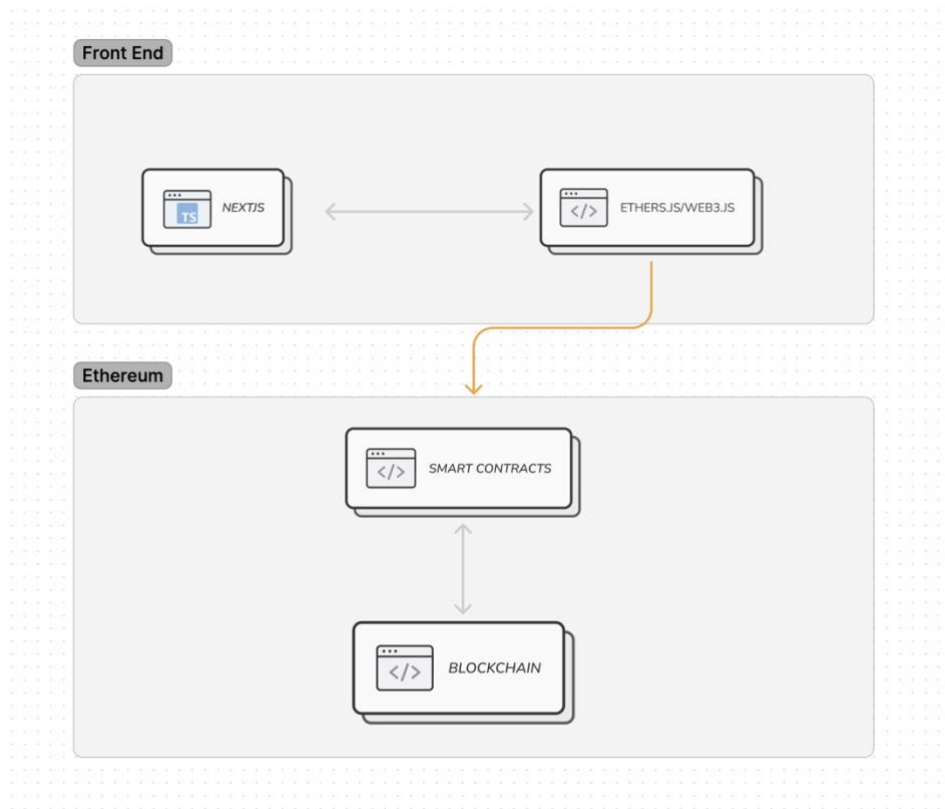
Truffle, part of truffle Suite is a powerful and widely used development framework for working with Ethereum smart contracts. It streamlines various tasks associated with smart contract development and deployment, such as compilation, linking, deployment, and testing. Truffle provides a robust testing platform for automated contract testing, simplifying the process of validating smart contract functionality (TruffleSuite, n.d.).

Ganache is one of the tools in the Truffle Suit, providing a local blockchain for development and testing. It simulates an Ethereum network with customizable settings and is available as a command-line tool and a GUI. Ganache enables rapid prototyping, development, and testing of smart contracts in a controlled environment before deployment to a live network (TruffleSuite, n.d.).

MetaMask as talked about before in our work is a popular browser extension and mobile app that serves as a cryptocurrency wallet and gateway to Ethereum-based decentralized applications. It allows users to manage multiple Ethereum accounts, securely store Ether and ERC-20 tokens, and interact with smart contracts and decentralized apps directly from their browser or mobile device. By acting as a bridge between the traditional web and the decentralized Ethereum network.

Solidity is a high-level, statically typed programming language specifically designed for writing smart contracts on Ethereum-based platforms. It is influenced by C++, Python, and JavaScript (Gavin Wood, 2014).

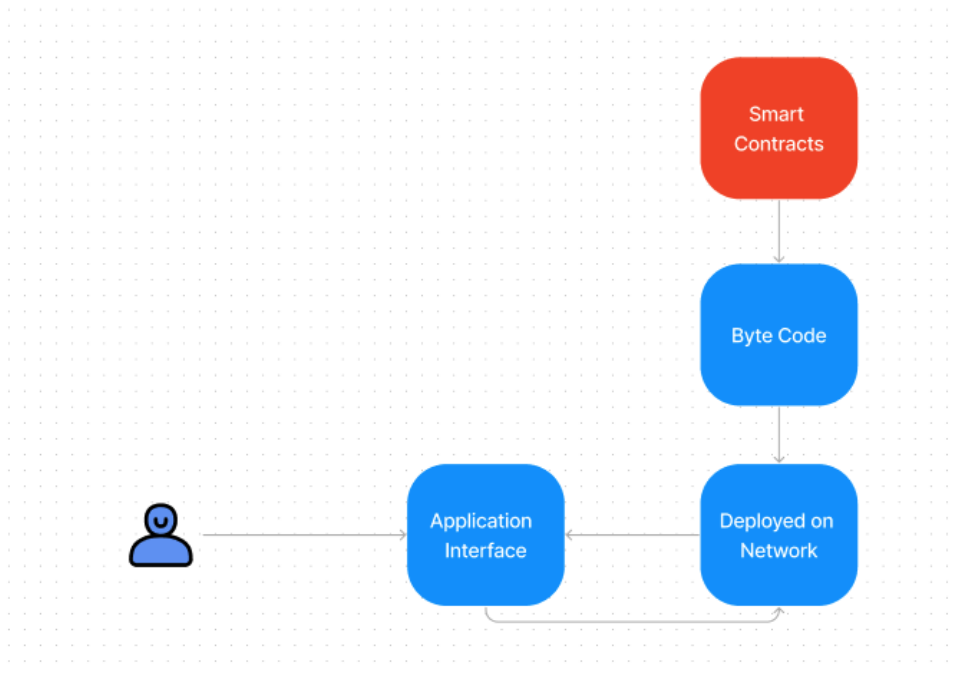
The tools chosen for the development are the most common technologies regarding Ethereum development. This aligns with the objective of developing an open source and approachable platform for further development. Figure 5 shows an overview of the architecture of the solution:



**Figure 5 - Overview of Ethereum Voting Solution**

In the front end this chosen framework is Next.js A framework built on top of React, it has become a popular choice for creating modern web applications, including single-page applications.

In addition to React framework the chosen libraries to communicate between MetaMask and the user are Ethers.JS and WEB3.JS. The open-source library provides the tools for managing Ethereum accounts, signing transactions, working with smart contracts, and handling Ethereum-based tokens. it can be easily integrated into JavaScript applications for seamless interaction with the Ethereum ecosystem.



**Figure 6 - Smart Contract Data Flow**

### 3.2. Proposal

This section describes our system working and functionality. This e-voting system works as follows: In our proposed e-voting system, voters can participate in the election process through a user-friendly voting website. To interact with the local blockchain, voters must use the MetaMask Extension to connect their Ethereum account/Wallet. Once connected, the website displays the list of candidates and their current vote counts, allowing the voter to select their preferred candidate and cast their vote. As seen on works of (Fernandes et al., 2021).

Our e-voting testing is done upon a local blockchain deployed using Ganache, which is connected to Meta mask for managing Ethereum transactions. The Truffle framework enables the deployment of Solidity-based smart contracts onto the local blockchain, which facilitates the development process and testing.

In this case the voter still must authorize the transaction and “Pay” For the vote, this payment is necessary for the approval of the Ethereum transaction.

The Smart Contract:

In the contract, two structs are created Candidate and Voter that are employed to store information about candidates and voters, respectively. The Candidate struct captures details such as the candidate's name, age, image, ID, vote count, and address, while the Voter struct records the voter's name, image, address, ID, voting allowance, voting status, and vote cast.



The contract assigns the address that deploys it as the voting organizer. This organizer has the authority to create candidates and authorize voters. To create a candidate, the organizer invokes the setCandidate function, which records the candidate's details, assigns a unique ID, and emits a CandidateCreate event. Similarly, the organizer can grant voting rights to individuals using the voterRight function, which sets the voter's details, including their voting allowance and status.

Voters can cast their votes using the vote function, which takes the candidate's address and vote ID as arguments. The function ensures that the voter is authorized and hasn't voted yet, then records the voter's choice and updates the candidate's vote count accordingly.

The contract also provides several view functions to access information about candidates, voters, and voting results. Functions like getCandidate, getCandidateData, getVoterData, getVotedVoterList, and getVoterList enable users to retrieve relevant data about the voting process, ensuring transparency and easy access to the voting records.

Figure 7 represents the application data flow for the voting system.

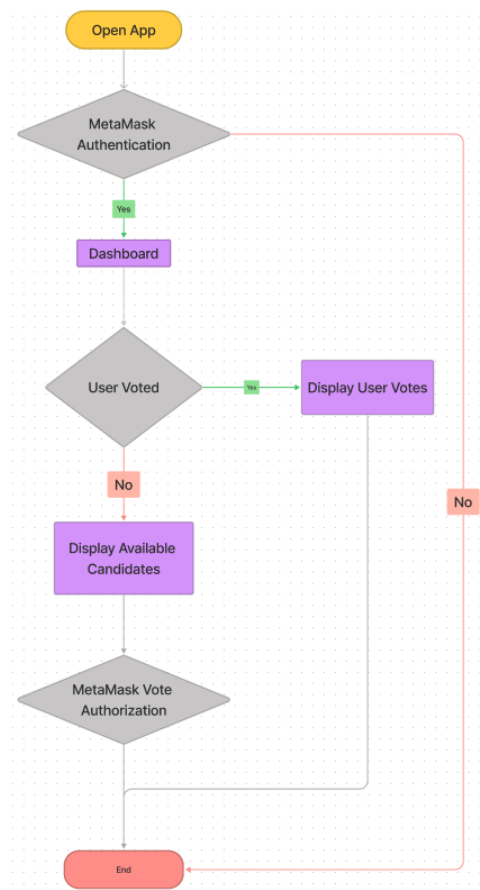


Figure 7 - Application Data Flow

### 3.3. Evaluation

In this section, we will examine and assess the performance of our Voting System during its first DSR Interaction. The evaluation process will help us identify potential enhancements, pinpoint areas that require corrections, and determine essential improvements to optimize the system's overall performance.

While our first DSR iteration has successfully demonstrated the potential of a decentralized voting system but there are several areas where improvements can be made. To address potential trust issues, the deployment process could be further decentralized, allowing for a more community-driven approach to election management. Currently the user authenticates and votes in the election without any need for any type of account verification. Currently it is responsibility of the election owner to create the voters for the election. With the function voterRight. The addition of multiple vote types, such as Weighted Voting and Token Based Voting is also one the additions for the next iterations, it's possible to see this feature in (Panja et al., 2020).

Enhancing voter privacy while maintaining the transparency and security of the system is a critical challenge that should be addressed in future iterations. Adding encryption technologies stated in works of (Farooq et al., 2022; Nguyen & Thai, 2022) and secret sharing techniques as seen on the works of (Chouhan & Arora, 2022).

As the number of voters and candidates increases, the system may face scalability challenges. Optimizing the smart contract and exploring layer 2 solutions could help improve the system's performance and accommodate larger elections. A feature seen on the work of (H. Zhu et al., 2022) This iteration demonstrates the potential of a layer 1 decentralized voting solution for the Ethereum environment. The application's core features and functionalities provide a solid foundation for a transparent voting process. But lack in features regarding Privacy and scalability.

However, addressing the challenges of decentralizing the deployment process, enhancing voter privacy, improving scalability, and refining the user experience in future iterations will contribute to the development of our blockchain-based voting system.

Ethereum has emerged as a leading platform for decentralized applications (dApps) and smart contracts. However, it faces a significant challenge in terms of scalability. Scalability, in this context, refers to the network's capacity to handle a substantial number of transactions simultaneously and efficiently. As of now, Ethereum can process approximately 15 transactions per second (TPS), a rate that is insufficient for global-scale applications. This limitation stems from Ethereum's inherent design, which prioritizes decentralization and security over high throughput.

Ethereum's scalability problem is primarily due to its consensus mechanism, Proof of Work. In PoW, every transaction must be validated by all nodes in the network, which is a time-consuming and resource-intensive process. Furthermore, each block in the Ethereum blockchain has a gas limit, restricting the number of transactions that can be included in a block. As the network grows and the number of transactions increases, these constraints result in slower transaction times and higher fees, thereby limiting the platform's scalability. This scalability issue is a significant hurdle for Ethereum, as it impedes the network's ability to support large-scale, high-speed applications, which are crucial for the widespread adoption of blockchain technology.

## 2<sup>nd</sup> DSR Iteration Proposal and Evaluation

This chapter will cover the evolution of the second iteration of the voting system. In its first section, we will delve into Problem Identification, where we pinpoint issues from the previous iteration. Subsequently, the Research and Analysis section will explore potential solutions in depth. We will also discuss the proposed architecture and provide detailed implementation concepts in the third section and evaluate the solution in the fourth section.

### 4.1 - Problem identification

The first iteration conducted as part of the DSR methodology, provides an assessment of the system's performance and highlights areas for future enhancements. This section aims to examine these areas of concern.

**User Authentication and Verification:** The system currently allows users to authenticate and cast their votes without any stringent verification processes. While this facilitates user engagement, it poses a significant security risk. The lack of a robust authentication system makes the platform susceptible to harmful activities, such as vote manipulation. Therefore, stricter user verification protocols must be instituted to enhance the integrity of the voting process.

**Privacy Concerns:** The matter of privacy remains a precarious balancing act. While transparency and security are of extreme importance, voter privacy cannot be compromised. Adding layers of encryption technologies as indicated in works by (Farooq et al., 2022; Nguyen & Thai, 2022), along with secret sharing techniques from (Chouhan & Arora, 2022) could be instrumental in augmenting the privacy aspect while maintaining system transparency.

**Scalability Issues:** As delineated in the evaluation, scalability is a known concern in the Ethereum mainnet. While Ethereum offers the benefits of robust decentralization, its throughput of approximately 15 transactions per second (TPS) is grossly inadequate for large-scale applications (Zhu et al., 2022). Consequently, the system faces challenges in scalability, which could impede its mass adoption. Smart contract optimization and layer-2 solutions may provide avenues for improving the system's scalability.

**Limitations of Ethereum's Architecture:** Ethereum's inherent design choices prioritize decentralization and security but do so at the expense of scalability and efficiency. Its consensus mechanism, Proof of Work (PoW), entails a comprehensive validation of each transaction by all network nodes, a process that is resource-intensive and restricts the number of transactions that can be processed in each block (Zhu et al., 2022). This problem is exacerbated when the network experiences increased load, leading to slower transaction times and higher transaction fees.

In conclusion the main problems we find in our previous DSR iteration is our scalability and privacy concerns. This Iteration will address these problems with a layer2 solution based on Ethereum. These layer2 Solutions allow the use of different consensus and cryptographic mechanisms that can bring benefits to our system.

## **4.2. Research and Analysis: Introduction to layer 2 And Ethereum Scaling.**

A layer 2 solution is a separate blockchain that extends Ethereum and inherits the security guarantees of Ethereum. (Ethereum, 2023) These Ethereum extensions have the goal of increasing scalability (transaction throughput) without sacrificing decentralization or security. There are different layer2 solutions with different trade-offs and this section will explore the most popular ones and their pros and cons. This section will also explore Sidechain and Validiums, these approaches scale similarly to Layer 2 solutions but are not considered as layer 2 solutions because of different trust assumptions. (Thibault et al., 2022) This matter will be discussed in the next section.

**RollUps:** Rollups bundle multiple transactions to lighten the load on the main network, distributing and executing the transactions off the main network but submitting the data to the main network. For this reason, Rollups benefit from the security of Ethereum. As this means that reverting data from these transactions would mean reverting the Ethereum main network, providing the safety of the main network to these scaling solutions. (Thibault et al., 2022) There are two main types of rollups right now.

**Optimistic Rollups:** Optimistic Rollups is an interactive approach since it involves the use of fraud proofs. In this kind of solution, new batches are published by operators, without being proved to be right by the Rollup smart contract. Unlike other scaling solutions, such as sidechains, optimistic rollups derive security from the Main net by publishing transaction results on-chain. (Barati & Rana, 2021)

**ZK-Rollups:** ZK-rollup operators submit a summary of the changes required to represent all the transactions in a batch rather than sending each transaction individually. They also produce validity proofs to prove the correctness of their changes. The validity proof demonstrates with cryptographic certainty that the proposed changes to Ethereum's state are truly the end-result of executing all the transactions in the batch (Ben Sasson et al., 2014).

Sidechains: Plasma Introduced in 2017 by Joseph Poon and Vitalik Buterin (Poon & Buterin, 2017) Layer 2 scalability solution for the Ethereum blockchain, enhancing its transaction throughput, latency, and cost-efficiency. Plasma seeks to extend the principles underlying sidechains to improve Ethereum's scalability. In traditional sidechain architectures, assets are deposited into a smart contract located on the primary or Layer 1 (L1) blockchain, such as Ethereum. These assets are then monitored by sidechain operators who credit the assets to users within the sidechain ecosystem. Sidechains often employ alternate consensus mechanisms like proof-of-authority to achieve faster block times, thereby significantly elevating transaction throughput and minimizing transaction costs.

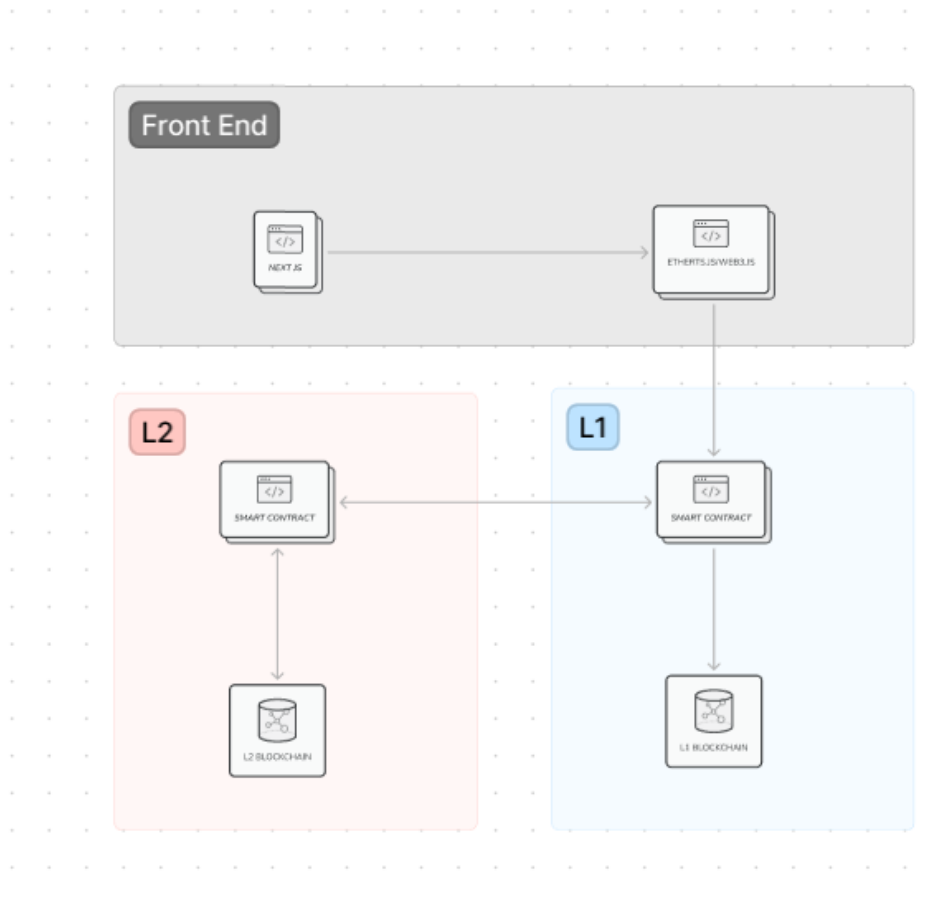
However, the gains in scalability and efficiency often come at the expense of decentralization and security. For instance, in proof-of-authority sidechains, a limited set of validators could arbitrarily halt the production of new blocks or stop processing withdrawal requests. (Jourenko et al., n.d.) Thus, these systems necessitate a degree of trust in the sidechain operators, compromising on the ideal of trustless interaction. Plasma introduces the concept of periodic commitments on the root or L1 chain. Specifically, each sidechain block header is published to the L1 chain. This strategy minimizes the trust required while enabling verifiable fraud proofs and enforceable state transitions, assuming the root chain remains secure and consistently accessible. By offloading a considerable volume of transactions from the Layer 1 to the Layer 2 Plasma chains, only periodic commitments are needed on the root chain. This not only enhances transaction throughput but also reduces storage requirements on the root chain.

Validiums: Validium is an off-chain scalability solution that resembles zk-Rollups and Optimistic Rollups but differentiates itself in the treatment of data availability. Unlike traditional Layer 2 solutions where data is posted on-chain for availability, Validium opts to keep most of the data off-chain, with only the proofs and a subset of data committed on-chain. A core tenet in the quest for scalability is data availability. Traditional Layer 2 solutions like Plasma struggle to achieve optimal data availability while maintaining security and decentralization. Validium addresses this by using zero-knowledge proofs to validate transactions off-chain and then submitting these proofs on-chain, thus optimizing data storage and computational resources.

In Validium, data is primarily stored off-chain in specialized data-availability layers, often maintained by a consortium of operators. The transaction data remains off-chain, but transaction proofs are computed off-chain and submitted to the root chain for verification. By doing this, Validium significantly reduces the data footprint on the main chain while still benefiting from its security (Creimer, 2023).

### 4.3 Proposal of Layer2 Solution

This Iteration of the DSR reutilized the Front end and most code from our first iteration, this section will cover the changes on the code to implement our voting system with a Layer 2 Solution. Figure 8 shows the updated architecture of our system:



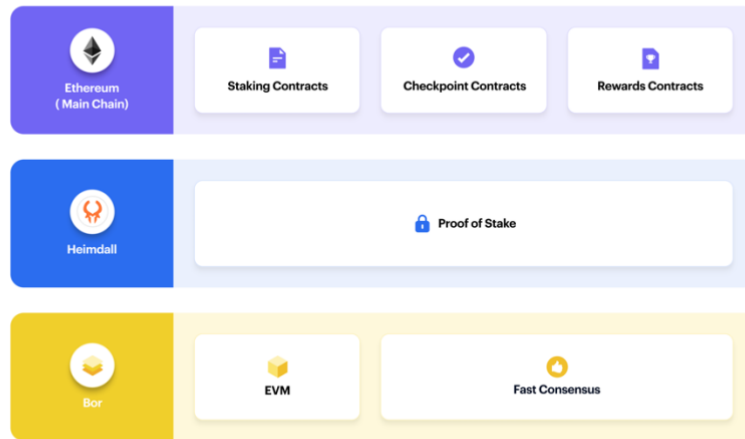
**Figure 8 - Overview of Layer2 Voting Solution**

In this iteration the front-end code remains basically the same, and Ethereum still is the foundation of the voting application. In this case this application was developed with Polygon. A Proof of Stake Layer2 solution. Polygon is fully compatible with Ethereum Virtual Machine. Meaning all the tools and processes from the previous iteration remain the same.

A Polygon PoS dApp consists of 3 layers:

- 1 Ethereum layer: a set of contracts on the Ethereum mainnet.
- 2 Heimdall layer; a set of proof-of-stake Heimdall nodes running parallel to the Ethereum mainnet, monitoring the set of staking contracts deployed on the Ethereum mainnet and committing the Polygon Network checkpoints to the Ethereum mainnet. Heimdall is based on Tendermint.

- 3 Bor layer: a set of block-producing Bor nodes shuffled by Heimdall nodes. Bor is based on Go Ethereum. (Polygon Documentation, 2023) Figure 9 demonstrates the Architecture of Polygon PoS:



**Figure 9 - Polygon Proof of Stake Architecture**

To enable the Proof of Stake mechanism, polygon employs a set of staking management contracts on the Ethereum mainnet. The staking contracts provide that anyone can stake MATIC tokens on the staking contracts on the Ethereum mainnet and join the system as a validator, the person who join as a validator earns staking rewards for validating state transitions on the Polygon Network. The contracts also act as save checkpoints on the Ethereum main net (Polygon Documentation, 2023).

Heimdall is the proof of stake validation layer that handles the aggregation of blocks produced by Bor and publishes the root chain. The periodic publishing of snapshots of Bor is called checkpoints. These checkpoints validate all the blocks since the last checkpoint. These Checkpoints provide proof of burn in withdrawal of assets.

Bor is Polygon's block producer layer responsible for aggregating transactions into blocks. Currently, it is a basic Geth implementation with custom changes done to the consensus algorithm. Block producers are a subnet of the validators and are periodically shuffled via committee selection on Heimdall in durations termed as a span in Polygon. Blocks are produced at the Bor node, and the VM is EVM-compatible. Blocks produced on Bor are also validated periodically by Heimdall nodes.

Polygon native cryptocurrency is Matic, meaning the system will use this cryptocurrency in the form of paying gas fees.



## 4.4 - Evaluation

Proof-of-Stake is a consensus mechanism where block creators are selected and rewarded based on the amount of tokens they hold into the system (Q. Zhou et al., 2020). This has a number of advantages, as the system being much more energy efficient. By disassociating the system from such an energy dependent mechanism such as PoW prevents some unexpected behaviours, such as when electricity cost rise miners reduce the mining effort to decrease mining difficulty (Thibault et al., 2022).

One of the most prevalent scalability features of polygon is block time, according to Etherscan (Etherscan, 2023) is possible to see a difference of 10x in block time from Ethereum to polygon.

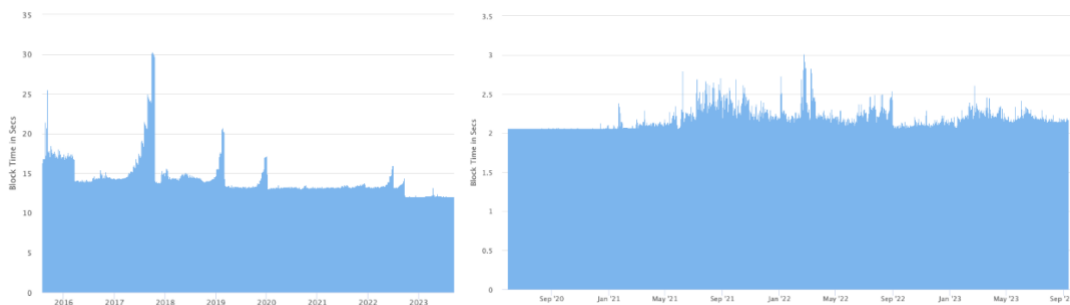


Figure 10 - Comparison between Block time Ethereum and Polygon

It is also possible to notice in figure 11 the difference in volume possible to be executed by the network. Where Ethereum had the peak transactions per day at 1,932,711 Polygon boasted 9,177,310 transactions in the 24h period.

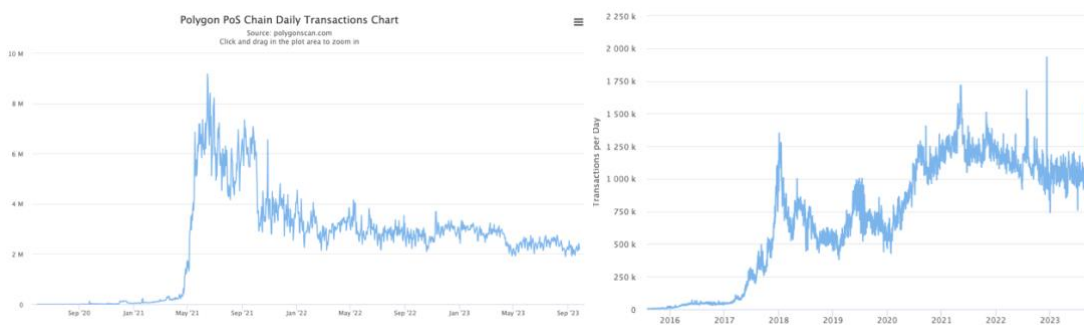


Figure 11 - Comparison between Transaction Count in 24 hour period

Sidechain, the case for Polygon PoS are connected to the main chain by a two way bridge, this means the possibility of running different consensus and technologies, but it also means the security is not guaranteed by the layer one protocol (Sguanci et al., 2021). In this case Ethereum.

PoS also can suffer for other security exploits such as the “Nothing as Stake” attack, where block creators work on several chain forks at the same time, and when these forks are incoherent cause conflicting transactions (W. Li et al., 2017). There are other multiple Proof of stake attack models known. and polygon has multiple security mechanisms to prevent these attacks. Such as Checkpointing. Where the networks snapshot the state of the blockchain and enforces these checkpoints on attackers, making it extremely difficult for an attacker to rewrite history and perform a long-range attack.

## 3<sup>rd</sup> DSR Iteration

After the evaluation of the previous DSR Iteration. This iteration explores a zero-knowledge virtual machine implementation Ethereum Virtual Machine, every time a program is executed a STARK Proof is generated, this allows anyone (that holds the proof) That a program has executed correctly without knowing the contents of the program itself. This ZK-Stark Virtual machine also allows multiple cryptographic operations in its program execution, allowing for a more secure execution (Polygon Miden VM, 2023).

Zero-Knowledge Proofs the fundamental piece of cryptography behind the blockchain technology explored in this iteration. This section will briefly introduce this concept and the subsequent sections will explore this concept and its nuances deeply.

A ZKP (zero-knowledge proof) allows one party to verify a claim that a transaction is valid or correct without the need to carry additional information about the transaction. Through them, cryptography found a way to prove the authenticity of a transaction without revealing sensitive information. For a zero-knowledge proof to work, it needs to fulfil three conditions – completeness, soundness, and zero-knowledge. ZKPs can be of one of two types – interactive and non-interactive. Interactive zero-knowledge proofs required constant, back-and-forth interactions between the prover and the verifier until the verifier confirmed the truth in the prover's claims. Non-interactive ZKPs require no back-and-forth interactions, with a single exchange of information sufficing to satisfy both parties (StarkWare, 2023).

### 5.1 Research and Analysis

#### 5.1.1 ZK-STARKs vs ZK-SNARKs

When comparing zk-SNARKs vs zk-STARKs, their fundamental differences are in transparency, zk-SNARKs are based on a trusted setup between the involved parties that generate the required randomness to produce the zero-knowledge proofs. These parameters need to be held in a safe environment, If the parameters fall into the wrong hands, dishonest actors could use them to create false proofs (Ben-Sasson et al., 2018). Creating a single point of failure in the system (W. Li et al., 2017) zk-STARKs work in a fundamentally different way. STARKs divide the computation where the prover recomputes the program and outputs the proof. The verifier runs the proof and checks if the steps taken in the verification where the same as in the original computation Without trusted setups, the parameters for generating randomness are public, limiting centralisation and empowering transparency. STARKs also allow for secret inputs, allowing for no leakage of data from the original computation (Szepeieniec, 2023).

Zero-knowledge SNARKs use an initial trusted setup to generate the random parameters. The computation of these parameter assumes that provers have limited computing power. However, when a prover uses an unlimited amount of computing power, they will be able to, for example, make use of algorithms that can execute extremely quick parallel integer factorization computations that can extract a private key from a public key. Breaching proof systems. Making SNARKs vulnerable to quantum computing attacks. zk-STARKs do not need an initial trusted setup and using a collision-resistant approach they do not require high computation costs, eliminating the threat of being compromised by quantum computing (Team, 2022).

### **5.1.2 Why a ZK-VM**

As of now, few ZK-rollups could do more than simple transactions (token transfers, atomic swaps, etc.) (Jourenko et al., n.d.). In the past, those that can support smart contract deployment usually required their developers to depart from the Ethereum framework, as existing ZK-rollups were not EVM (Ethereum Virtual Machine) compatible, this meant they could not execute smart contracts. Zero-knowledge Ethereum Virtual Machine They aim to replicate the Ethereum environment as a rollup, allowing developers to build on them like they would on Ethereum. They can also execute smart contracts in a manner that supports zero-knowledge technology.

Zk-VMs follow the general workings of zero-knowledge rollups. However, it is essential to note that several ideas surround the ideal structure of a Zk-VM and its operations. The truest form of a zero-knowledge EVM would be fully Ethereum-equivalent, allowing no changes even if they could make proof generation any easier. Zero-knowledge rollups take and complete transactions off-chain, in batches, and submit a cryptographic report that proves the correctness of these interactions to Ethereum (Vilà Brualla, 2022). The zero-knowledge proof does not reveal the details of all the transactions in the batch, but only confirms that they are accurate enough to trigger a transition to the Ethereum state. After that, it provides validity proofs to a smart contract set up on the L1 chain. Once received and confirmed, it verifies the inputs. To understand how Zk-VMs work, we must acknowledge that they are of different types, as shown by the projects currently in the works. Although they all share common goals, they differ in approach.

### 5.1.3 Current Zk-VM and Zk-EVM Solutions

Polygon announced they were building a zkEVM, rebranding the Polygon Hermez project to Polygon zkEVM. Polygon zkEVM is open-source, and adopted the Type-2 approach, aiming to be EVM-equivalent but falling short of Ethereum-equivalence. Polygon zkEVM's will be fully adaptable to the EVM tools. Polygon expects it to reach a higher transactions per second throughput and cut transaction costs, being far less expensive than the Ethereum Mainnet without compromising security or efficiency, making it a good choice for scaling decentralized applications (dApps) that demand large throughput. In Oct.10, Polygon launched its zkEVM Public Testnet. Polygon Hermex became a payment focused product with stated 2000 TPS (Polygon Hermez, 2020).

Created by Matter Labs, zkSync is a layer-2 scaling solution that adopts the Type-4 approach, supporting compatibility with Solidity and Vyper, Ethereum's coding languages. zkSync 1.0 is already live, having processed millions of transactions (zkSync Era Block Explorer, 2023). zkSync's new product, zkSync 2.0 is a EVM-compatible ZK rollup powered by a zkEVM, although it falls under EVM compatibility rather than EVM equivalence. Within the zkSync ecosystem, developers are able to write Solidity smart contracts, which the protocol will transpire into Yul, and recompile the Yul bytecode to a custom bytecode set specially designed for zkSync's EVM. As a Type-4, zkSync Era experienced quicker proving times but suffers from less application compatibility than its competitors. zkSync is live on Ethereum Testnet.

StarkNET: While most zkEVM projects use ZK-SNARKs, Starkware's StarkNET adopts ZK-STARKs, which are more secure than ZK-SNARKs in theory but require more gas, take longer to verify, and occupy more block space. StarkNET has already launched its Alpha version and as of today is on version 0.12.2 (StarkNet, 2023a), although it remains limited. StarkNET, like zkSync, follows the Type-4 zkEVM approach, which categorizes it as compatible with Solidity or high-level languages. StarkNet STARK proofs are written in CAIRO a Turing complete programming language designed for STARK proofs (StarkNet, 2023b).

Scroll: In collaboration with the Privacy and Scaling Explorations group, Scroll is a zkEVM solution that angles towards the Type-2 class, like Polygon zkEVM. A completely compatible with Ethereum interfaces. The project has a live main net and aims to build the first genuinely EVM-equivalent zkEVM and prioritize security and transparency (Scroll, 2023).

Consensus zkEVM: Developed by ConsenSys R&D and run by ConsenSys, the ConsenSys zkEVM network is a new Type-2 zkEVM. The rollup offers full Ethereum Virtual Machine (EVM) compatibility, allowing developers to deploy and maintain applications using well-known tools like MetaMask, Truffle, and Infura as if they were using Ethereum directly. Consensus deployed a live testNet on March 3<sup>rd</sup> 2023 (Liochon, 2023).

## 5.2. Proposed Development

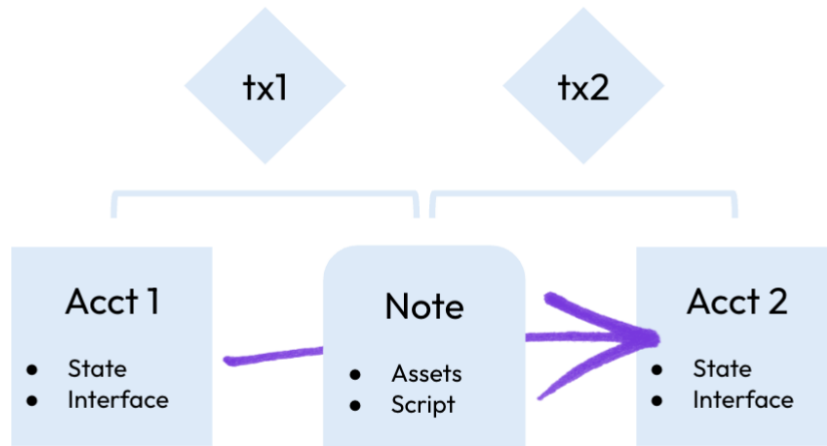
### 5.2.1 What is Miden

Polygon Miden is a zero-knowledge rollup running on the Miden Virtual Machine. "Polygon Miden prioritizes ZK-friendliness over EVM compatibility, that way it can offer features and benefits that are not available on Ethereum. It aims at builders that want to create high-throughput and private dApps. Miden is a general-purpose rollup and builders can write and deploy arbitrary smart contracts". Polygon Miden is consistent of multiple Parts:

AirScript	Domain-specific language for writing AIR constraints for Miden VM.
Miden Base	Core components of the Polygon Miden rollup.
Miden Client	A reference Miden Client to be used by users.
Miden Crypto	Cryptographic primitives used in Polygon Miden rollup.
Miden IR	MidenIR for compiling to Miden Assembly from higher-level languages.
Miden Node	Miden Node to be used by the Miden operators.
Miden VM	STARK-based virtual machine for Polygon Miden rollup.

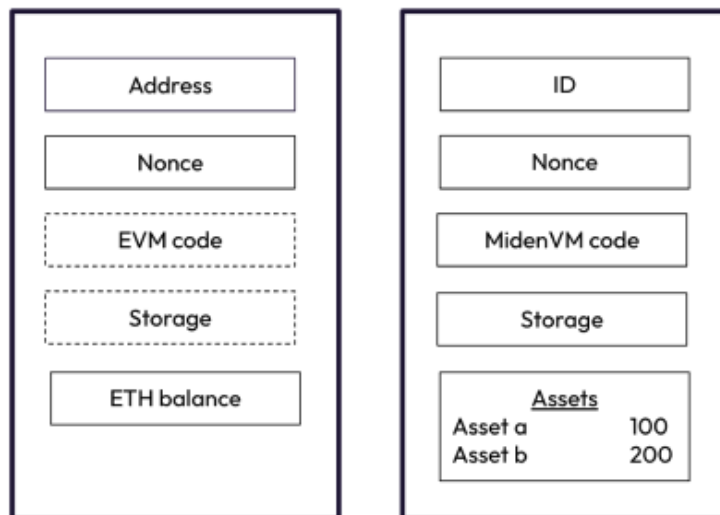
This DSR iteration will focus on the Miden VM and Miden Assembly language for the development of the Voting use case. Some of features the next sections are going to explore are:

**Transaction Model:** Miden offers accounts and notes, both of which can hold assets. Transactions are facilitation on state changes, where one account and notes act as input and output with new notes and new state. Notes are an interesting parallel with the actor Model, commonly used in distributed system, where each 'actor' behaves like a state machine. In this model actors communicate by exchanging messages which can be processed asynchronously and change internal state. In the case of the network accounts communicate with each other by consuming notes. Notes act as messages that carry assets and scripts that can define how an asset can be consumed. This model distances itself from Ethereum account model as this model combines the elements of (UTXO) Unspent Transaction Output model and the account-based model. This approach provides more flexibility and functionality.



**Figure 12 - Miden Transaction Model**

Account and Asset Mode: On Miden, it is possible for a user to create and trade fungible and non-fungible assets. Similar to Ether in the Ethereum network. Miden offers native assets that are data structures that follow the asset model. Differently than Ethereum, accounts in Miden hold assets locally. Where in Ethereum accounts only hold Ether Balance. This provides more privacy and scalability.



**Figure 13 - Miden Account Model**

### 5.2.2 – Explaining the Miden VM:

Miden VM is a zero-knowledge virtual machine, for any program executed on Miden VM, a STARK-based proof of execution is automatically generated. This proof can then be used by anyone to verify that the program was executed correctly without the need for re-executing the program or even knowing the contents of the program (Polygon Miden, 2023). As of the time of writing, Miden is on alpha release 0.6, given most core features have been stabilized and STARK proof has been implemented. As a virtual machine Miden is Turing complete and provides all features necessary in a VM.

To write a program to run in Miden is necessary to write the program in Miden Assembly, a low-level language that is compiled in raw instruction sets. It is possible to work with input with the operand stack and Advice Stack. The program outputs a single stack after a successful execution.

### 5.3 – Our EVM Voting Solution.

This section goes over the development of an equivalent voting system to the previous two iterations utilizing the Miden virtual machine. To develop the system accordingly it is necessary to write the equivalent of the voting contract in Miden assembly and recreate the virtual machine environment for the testing. In this case the chosen method is to connect a RUST web server that was deployed running the Miden-vm crate. In this case when the server receives a request the virtual machine will process the request, create the proof and run the proof through the verifier.

To execute the contract and test the Proof of Concept the virtual machine was deployed together with Rust Rocket, which provided the necessary Application Programming Interface (API) tooling to connect the Front End to the Miden-vm server. This section will go over implementation details and explanations of the Virtual machine workings.

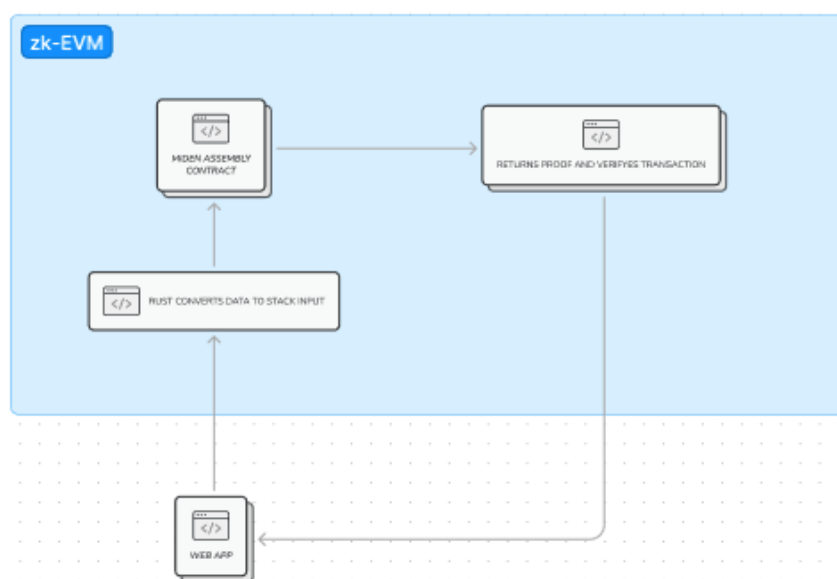


Figure 14 – zk-Evm Solution dataflow



### 5.3.1 - Overview

The Voting Contract is written in MASM (Miden assembly) and receives the stack input from the advice provider. The advice provider is used for non-deterministic inputs. Figure 15 shows example of the operand and advice stack, in this example this formatting is used for debugging purposes.

```
{
  |
  | "operand_stack": ["0"],
  | "advice_stack": [
  | | "2", "3"
  | ]
}
```

Figure 15 - Debug Inputs

Although in this example the program pushes the inputs to the advice stack accordingly and loads the values to the current memory. During execution the values are loaded from a vector of Field Element Values. In the case of this implementation Field Values were implemented based on (Pornin, 2022) cGFp5: a Specialized Elliptic Curve. This is achieved through winter-math crate. Figure 16 demonstrates a representation loading the values from a hardcoded variable, creating the values from an unsigned integer.

```
let mut candidate_1: BaseElement = Felt::from(0_u32);
let mut candidate_2: BaseElement = Felt::from(0_u32);
let mut vote_index: BaseElement = Felt::from(0_u32);

let mut populated_stack: Vec<Felt> = vec![Felt::from(candidate_1), Felt::from(candidate_2), Felt::from(vote_index)];
```

Figure 16 – Definition of Field Elements from input of unsigned Integers

```
let advice_provider: MemAdviceProvider = MemAdviceProvider::from(advice_inputs);
let host: DefaultHost<MemAdviceProvider> = DefaultHost::new(adv_provider: advice_provider);
```

Figure 17 – Load of Advice Provider

After creating the vector of values, it is necessary to load the advice inputs in memory with the MemAdviceProvider, then create a host with the loaded memory. The functionality of the Host is to communicate with the virtual machine.

For the program to execute in the RUST server is necessary to load the VM, figure 18 demonstrates the load of MASM contract with an immutable variable called assembler from the standard library. In this example the assembler is loaded as default and compile the program to be used in the execute function:

```

let source: String = format!(
    "
    begin
        adv_push.3
        push.0
        eq
        if.true
            push.1
            add
        else
            swap
            push.1
            add
            swap
        end
    end
    ",
);
let program: Program = Assembler::default().compile(&source).unwrap();

```

**Figure 18 – MASM Contract**

As the Virtual Machine works with a single stack of public and secretive inputs, and with the objective of keeping voters privacy and not allow a third party observer to get information during the process the contract was limited to use only one stack.

The contract works taking  $N + 1$  variables as input,  $N$  being the number of candidates on the election. The candidates inserted in a sorted order on the stack and the last parameter is the index of the candidate in which the voter desires to vote. The contract then manipulates the stack to move the index accordingly and adds one to that candidates index. Then the contract reorders the stack to output the result of that vote.

To execute the contract and create a proof of the execution it necessary to use the Prove function.

```

let (outputs: StackOutputs, proof: ExecutionProof) = prove(
    &program,
    StackInputs::default(),
    host: DefaultHost::from(host),
    options: ProvingOptions::default(),
) Result<(StackOutputs, ExecutionProof), ...>
.unwrap();

```

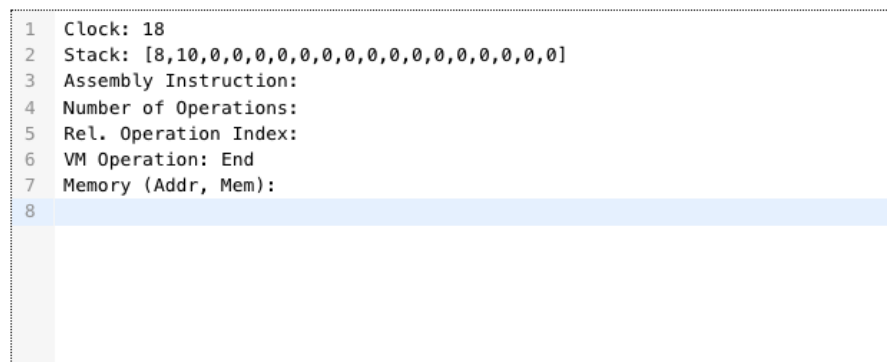
**Figure 19 – Prove function.**

After running the prove function in which the program inputs a reference of the MASM contract, inputs the Host that communicates with the advice provider and in this case, it's running with default options. The program loads the stack elements into the candidate variables. This provides the VM to keep updating the contract.

```
candidate_1 = Felt::from(stack_element[1]);  
candidate_2 = Felt::from(stack_element[0]);
```

**Figure 20 – Redefinition of Field Elements from Stack Elements**

The simple voting contract developed consists of multiple simple stack operations on the Virtual machine. Each of these operations has a “price” and some operations consume more cycles than others. When running the program in a debugging tool it's possible to check that a single voting operation in the virtual machine costs 18 clocks, and when tested in Miden Web Assembly environment took an average of 5ms of runtime and 100ms of proving time. Proof sizes are an average of 33kb. Proof generation is a fairly adjustable and is possible to trade off execution time, proof size and security level. Figure 21 shows the logs from a debug execution.



```
1 Clock: 18  
2 Stack: [8,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]  
3 Assembly Instruction:  
4 Number of Operations:  
5 Rel. Operation Index:  
6 VM Operation: End  
7 Memory (Addr, Mem):  
8
```

**Figure 21 – Debug Logs**

## 5.4 – Evaluation

Instead of relying on the traditional method of verifying transactions by re-executing them, Polygon Miden leverages zero-knowledge proofs to eliminate the need for re-execution. Zero-knowledge proofs enable verification without requiring transparency or significant processing power. Users can perform smart contract operations on their own devices and subsequently transmit zero-knowledge proofs to the network. Operators can then verify these proofs significantly faster than executing the original transactions and update the blockchain state accordingly. This approach has its limitations regarding privacy but the gains on scalability are significant with this approach.

Polygon Miden benchmarks demonstrates a execution time of 1ms for  $2^{10}$  virtual machine cycles, but it was not possible to achieve this level of performance for as is unclear which hash function was being used to create the stark proofs. Table 13 demonstrates Miden benchmarks for the single-core prover execution.

**Table 13 – Single Core Benchmarks**

<b>VM cycles</b>	<b>Execution time</b>	<b>Proving time</b>	<b>RAM consumed</b>	<b>Proof size</b>
$2^{10}$	1 ms	120 ms	30 MB	61 KB
$2^{12}$	2 ms	460 ms	106 MB	77 KB
$2^{14}$	8 ms	1.4 sec	500 MB	90 KB
$2^{16}$	27 ms	4.9 sec	2.0 GB	103 KB
$2^{18}$	81 ms	20.1 sec	8.0 GB	121 KB
$2^{20}$	310 ms	90.3 sec	20.0 GB	138 KB

During the local testing of the solution it was possible to test the speed for execution and proof creating for our program. The testing was done in a M1 Pro Chip with 16GB of memory. In this testing it was possible to achieve an execution and proof creation time of 60ms meaning the local execution proved to be faster then the Miden WebAssembly environment.

Privacy:

Traditionally there is no privacy in blockchain, with its immutable and transparent data structure. Miden aims to provide privacy with client side proving where our system doesn't disclose transaction details, individuals only send the zk-Proof of their transactions. By the nature of the ZK Virtual machine, privacy becomes a natural option with Miden. With Miden users can store data publicly, encrypted and off-chain. When storing data off chain and committing proofs to the network the network gains a significant boost in privacy and scalability.

(PolygonBlog, n.d.) Raises a question about how experienced observers could gain information by monitoring the network state and its changes. This question is due to Miden Accounts and Notes architecture:

*“Every note in Polygon Miden has a unique nullifier that points to either 0 or 1, depending on whether the note is already consumed. Now, the observer could watch the nullifier database and observe with which transaction a certain nullifier change. For public notes, it is possible to watch the nullifier database and gain information about who consumes a note. In this case, the note data is publicly visible, and the network knows the sender, the recipient, and the assets involved. However, for notes that are stored off-chain, the network only knows the sender and the note hash, with even the sender being mask-able. Outsiders cannot compute and know the nullifier with this information.”*

## Discussion and Conclusion

### 6.1. Discussion

The process and results of this dissertation is intertwined with the developments of new technologies such as ZK-Stark proofs and the advance in multiple layer scaling solutions, demonstrating the applications of these technologies in a voting solution and providing key insights from the extensive systematic literature review completed.

This work findings extend previous literature regarding blockchain voting with the application of ZK-Stark Proofs providing a framework for other authors to extend this work and test it's application in a real world scenario. While this experiments in these new technologies demonstrated positive outcomes with experimentation the lack of developer tools and documentation proved to be difficult entry barriers for developers, currently in miden it is necessary to create contratcs with miden Assembly, an assembly like language that allows full control of the Miden Virtual machine .

The use of Polygon Miden, a project in which is still in its beta development phase also difficult the comparison with other solutions as it is not feasible to run the solution in a test network at the moment, with that in mind this work still tested the solution locally and benchmarked it's results regarding computational stress, speed and size and its performance regarding virtual machine cycles.

### 6.2. Conclusion

The proposed Voting Solutions were developed with stablished blockchain technologies and with up-and-coming technologies still in development with the purpose of identifying the most resilient and well-suited technology for a voting solution and answering the hypotheses proposed in this dissertation. Regarding the first hypotheses is possible to conclude that current blockchain technology is mature enough to develop an electoral process, and the current fast paced evolution of technologies such as ZK-STARKs play a crucial role in the implementation of future voting systems. During the Systematic literature review it was also possible to see current developed by companies such as Voatz and Agora, both using blockchain technologies for their voting solution.

Regarding the second hypotheses, the systematic literature review (SLR) presented in this dissertation revealed a challenge in obtaining sufficient data to comprehensively address issues like political bias, legislation and regulation, public trust, and the public's perception in the context of blockchain e-voting. While these aspects are recognized as critical for successful implementation, the lack of available data highlights a gap in existing research and emphasizes the need for further investigation. Nonetheless the research pointed how some countries successfully integrated voter identification and validation in their systems with the goal of a future Electronic Voting Process and highlighted current trust issues and treats regarding electronic voting. However it was not possible to validate the second hypotheses with this current research, it is apparent that other aspects could also have been targeted in the developed solutions such as Voter Identification and Registration. The positives aspects of proposed Voting Solution are the scalability increase, although Zero Knowledge Proof calculations used in our solutions are expensive to compute, the possibility of computing the proof locally and uploading only the proof to main layer of the blockchain provides a relief on the network.

Secondly, the client side proving technology Developed by Miden Virtual machine provides that the local machine the solution is running the device creates the proof for the blockchain. This not only is a different paradigm in question of how current solutions work but provides a basis for running voting machines such as Brazil already use. Given that the proof is rapidly generated in personal computers, achieving this kind of solution would not be a long shot.

The negative aspect of this solution is lack of real-world testing, not being able to deploy this solution to small voting use case would provide in-depth understanding and examination of how the Solutions performs. That said, a small use case would not test the scaling capabilities shown in the solution. Other negative Aspect in the final solution is the Polygon Miden Technology being in such an early stage, by the time the solution was developed Miden was in release 0.5, that means most Virtual Machine interfaces were stable, which allowed us to build the solution, but did not allow us to deploy the solution in multiple nodes. A Test net is planned by the development team in the end of 2023 which means too late for this work to perform testing on a deployed network.

The findings from the evaluation suggested that ZK-STARK cryptography together with the Client side proving mechanism provides a good balance in security and scalability, although these technologies are still in development. The ZK-STARKS are also Quantum proof.

### **6.3. Contributions**

The findings of this dissertation were selected for publication and presentation at the 5th IEEE-International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA 2023) conference. This recognition highlights the significance and interest of this research for the scientific community. This resulted in the publication (Castro and Coutinho, 2023), providing a

basis for the development of further Decentralized Voting Solutions, using avant-garde decentralized technology with cutting edge cryptography that is at the current moment the next step for a scalable blockchain system. This work also provides a comprehensible review of current state of the art Decentralised Voting systems, their flaws, limitations, and positives.

## **6.4. Limitations**

This research has certain limitations. One of them being the lack of real use cases to test and further develop the Voting Solution. The lack of a real use case makes as the Voting Solutions doesn't explore Voter Authentication and Voter Verification, as the research only focused on themes such as: Scalability, Privacy and Decentralization. Other Limitation is the lack of security testing in the solutions. There are known and unknown security flaws in the current Virtual Machine release, and these hope to be identified and fixed before 1.0 launch.

## **6.5. Future Work**

As a proposal for future work, it would be interesting to explore on the theme of user Authentication, registration and verification, a challenge that involves not only technical mastership but also knowledge of the workings of an election. This is an essential piece of an election system and when in context of government elections. The complexity and possibility of integration of the network accounts with the government systems would be a desired or an alternative to a possible solution.

It would also be interesting to develop the network solution thinking about the scalability of a countywide voting solution. Currently small use cases could be run in in previous developed technologies, but current networks would not support a countrywide election.

## Bibliography

- Agora. (2023). <https://www.agora.vote/>.
- Al-Maaitah, S., Qatawneh, M., & Quzmar, A. (2021). E-Voting System Based on Blockchain Technology: A Survey. *2021 International Conference on Information Technology (ICIT)*, 200–205. <https://doi.org/10.1109/ICIT52682.2021.9491734>
- Banawane, A., Bhansali, Y., Dabadgaonkar, M., Javalekar, O., Patil, G., & Kumavat, Mrs. K. (2022). A Novel Approach for e-Voting System Using Blockchain. *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, 263–268. <https://doi.org/10.1109/ICIEM54221.2022.9853099>
- Barati, M., & Rana, O. (2021). Privacy-aware cloud ecosystems: Architecture and performance. *Concurrency and Computation: Practice and Experience*, 33(23), e5852. <https://doi.org/10.1002/cpe.5852>
- Ben Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., & Virza, M. (2014). Zerocash: Decentralized Anonymous Payments from Bitcoin. *2014 IEEE Symposium on Security and Privacy*, 459–474. <https://doi.org/10.1109/SP.2014.36>
- Ben-Sasson, E., Bentov, I., Horesh, Y., & Riabzev, M. (2018). *Scalable, transparent, and post-quantum secure computational integrity*.
- Brooks, C. (2023). Cybersecurity Trends & Statistics For 2023; What You Need To Know. *Forbes*.
- 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>
- Chouhan, V., & Arora, A. (2022). Blockchain-based secure and transparent election and vote counting mechanism using secret sharing scheme. *Journal of Ambient Intelligence and Humanized Computing*. <https://doi.org/10.1007/s12652-022-04108-0>
- Conference Rank. (n.d.). <http://www.conferenceranks.com/>.
- Creimer, M. (2023). *Validiums*. <https://Ethereum.Org/Pt-Br/Developers/Docs/Scaling/Validium/>.
- de Castro, A., & Coutinho, C. (2023). Electronic Voting Through Blockchain: A Survey. *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 1–6. <https://doi.org/10.1109/HORA58378.2023.10156749>
- Ethereum. (2023, August 29). <https://ethereum.org/en/layer-2/>. <https://ethereum.org/en/layer-2/>
- Ethereum Foundation. (2023). *The Merge*.
- Etherscan. (2023, October 8). <https://etherscan.io/>.
- Farooq, M. S., Iftikhar, U., & Khelifi, A. (2022). A Framework to Make Voting System Transparent Using Blockchain Technology. *IEEE Access*, 10, 59959–59969. <https://doi.org/10.1109/ACCESS.2022.3180168>
- Fernandes, A., Garg, K., Agrawal, A., & Bhatia, A. (2021). Decentralized Online Voting using Blockchain and Secret Contracts. *2021 International Conference on Information Networking (ICOIN)*, 582–587. <https://doi.org/10.1109/ICOIN50884.2021.9333966>



- Gavin Wood. (2014). <https://docs.soliditylang.org/en/v0.8.19/>.
- Heinl, M. P., Gölz, S., & Bösch, C. (2023). Remote Electronic Voting in Uncontrolled Environments: A Classifying Survey. *ACM Computing Surveys*, 55(8), 1–44. <https://doi.org/10.1145/3551386>
- Hevner, March, Park, & Ram. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75. <https://doi.org/10.2307/25148625>
- Hjalmarsson, F. P., Hreiðarsson, G. K., Hamdaqa, M., & Hjalmtýsson, 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>
- Huang, J., He, D., Chen, Y., Khan, M. K., & Luo, M. (2022). A Blockchain-Based Self-Tallying Voting Protocol With Maximum Voter Privacy. *IEEE Transactions on Network Science and Engineering*, 9(5), 3808–3820. <https://doi.org/10.1109/TNSE.2022.3190909>
- Huang, J., He, D., Obaidat, M. S., Vijayakumar, P., Luo, M., & Choo, K.-K. R. (2022). The Application of the Blockchain Technology in Voting Systems. *ACM Computing Surveys*, 54(3), 1–28. <https://doi.org/10.1145/3439725>
- Ingraham, C. (2020). *Trump's most worrying attacks on democracy, in one giant chart*.
- Jourenko, M., Larangeira, M., Kurazumi, K., & Tanaka, K. (n.d.). *SoK: A Taxonomy for Layer-2 Scalability Related Protocols for Cryptocurrencies*.
- Kamran, Nasir, M. H., Imran, M., & Yang, J.-S. (2021). Study on E-Voting Systems: A Blockchain Based Approach. *2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 1–4. <https://doi.org/10.1109/ICCE-Asia53811.2021.9641914>
- Kaspersky. (n.d.). [https://www.kaspersky.com/about/press-releases/2020\\_polys-from-kaspersky-innovation-hub-presents-first-blockchain-based-voting-machine](https://www.kaspersky.com/about/press-releases/2020_polys-from-kaspersky-innovation-hub-presents-first-blockchain-based-voting-machine). 2020.
- Khandelwal, A. (2019). Blockchain implementation on E-voting System. *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, 385–388. <https://doi.org/10.1109/ISS1.2019.8907951>
- Killer, C., Rodrigues, B., Matile, R., Scheid, E., & Stiller, B. (2020). Design and implementation of cast-as-intended verifiability for a blockchain-based voting system. *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 286–293. <https://doi.org/10.1145/3341105.3373884>
- Kitchenham, B. (n.d.). *Kitchenham, B.: Guidelines for performing Systematic Literature Reviews in software engineering*. <https://www.researchgate.net/publication/258968007>
- Kumar, M. (2021). Securing the E-voting system through blockchain using the concept of proof of work. *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, 423–427. <https://doi.org/10.1109/ICTAI53825.2021.9673389>
- Li, M., Luo, X., Sun, W., Li, J., & Xue, K. (2022). AvecVoting: Anonymous and Verifiable E-voting with Untrustworthy Counters on Blockchain. *ICC 2022 - IEEE International Conference on Communications*, 4751–4756. <https://doi.org/10.1109/ICC45855.2022.9838840>
- Li, W., Andreina, S., Bohli, J.-M., & Karame, G. (2017). *Securing Proof-of-Stake Blockchain Protocols* (pp. 297–315). [https://doi.org/10.1007/978-3-319-67816-0\\_17](https://doi.org/10.1007/978-3-319-67816-0_17)

- Li, Y., Susilo, W., Yang, G., Yu, Y., Liu, D., Du, X., & Guizani, M. (2022). A Blockchain-Based Self-Tallying Voting Protocol in Decentralized IoT. *IEEE Transactions on Dependable and Secure Computing*, 19(1), 119–130. <https://doi.org/10.1109/TDSC.2020.2979856>
- Liochon, N. (2023). *ConsensysZkEVM*. <https://Consensys.Net/Zkevm/>.
- Lyu, J., Jiang, Z. L., Wang, X., Nong, Z., Au, M. H., & Fang, J. (2019). A Secure Decentralized Trustless E-Voting System Based on Smart Contract. *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 570–577. <https://doi.org/10.1109/TrustCom/BigDataSE.2019.00082>
- March, & Storey. (2008). Design Science in the Information Systems Discipline: An Introduction to the Special Issue on Design Science Research. *MIS Quarterly*, 32(4), 725. <https://doi.org/10.2307/25148869>
- Matheus Teixeira. (n.d.). Mourão faz pronunciamento com crítica velada a Bolsonaro e diz que Forças Armadas pagam a conta. <https://Www1.Folha.Uol.Com.Br/Poder/2022/12/Mourao-Critica-Membros-Dos-3-Poderes-e-Nao-Cita-Nome-de-Bolsonaro-Nem-Lula-Em-Pronunciamento.Shtml>.
- MetaMask. (n.d.). <https://metamask.io/>.
- Nguyen, T., & Thai, M. T. (2022). zVote: A Blockchain-based Privacy-preserving Platform for Remote E-voting. *ICC 2022 - IEEE International Conference on Communications*, 4745–4750. <https://doi.org/10.1109/ICC45855.2022.9838690>
- NPM. (n.d.).
- Ohammah, K. L., Thomas, S., Obadiah, A., Mohammed, S., & Lolo, Y. S. (2022). A Survey on Electronic Voting On Blockchain. *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)*, 1–4. <https://doi.org/10.1109/NIGERCON54645.2022.9803127>
- Panja, S., Bag, S., Hao, F., & Roy, B. (2020). A Smart Contract System for Decentralized Borda Count Voting. *IEEE Transactions on Engineering Management*, 67(4), 1323–1339. <https://doi.org/10.1109/TEM.2020.2986371>
- Panja, S., & Roy, B. (2021). A secure end-to-end verifiable e-voting system using blockchain and cloud server. *Journal of Information Security and Applications*, 59, 102815. <https://doi.org/https://doi.org/10.1016/j.jisa.2021.102815>
- Pawlak, M., & Poniszewska-Marańda, A. (2019). Blockchain e-voting system with the use of intelligent agent approach. *Proceedings of the 17th International Conference on Advances in Mobile Computing & Multimedia*, 145–154. <https://doi.org/10.1145/3365921.3365927>
- Polygon. (n.d.). <https://polygon.technology/>.
- Polygon Documentation. (2023, October 8). <https://wiki.polygon.technology/docs/pos/polygon-architecture/>.
- Polygon Hermez. (2020). *Scalable payments. Decentralised by design, open or everyone.*
- Polygon Miden. (2023). *Polygon Miden Documentation: 1.1 Usage.*

- Polygon Miden VM. (2023). *Cryptographic operations, MidenVM (0.6)*.  
[https://0xpolygonmiden.github.io/miden-vm/user\\_docs/assembly/cryptographic\\_operations.html](https://0xpolygonmiden.github.io/miden-vm/user_docs/assembly/cryptographic_operations.html).
- PolygonBlog. (n.d.). <https://polygon.technology/blog/privacy-a-fundamental-right-and-a-practical-necessity>.
- Poniszewska-Maranda, A., Rojek, S., & Pawlak, M. (2022). Decentralized electronic voting system using Hyperledger Fabric. *2022 IEEE International Conference on Services Computing (SCC)*, 339–348. <https://doi.org/10.1109/SCC55611.2022.00056>
- Poon, J., & Buterin, V. (2017). *Plasma: Scalable Autonomous Smart Contracts*. <https://plasma.io/>
- Pornin, T. (2022). *EcGFp5: a Specialized Elliptic Curve*.
- Scimago. (n.d.). <https://www.scimagojr.com/>.
- Scroll. (2023). *Scroll*. <https://Scroll.io/>.
- Sguanci, C., Spatafora, R., & Vergani, A. M. (2021). *Layer 2 Blockchain Scaling: a Survey*. <http://arxiv.org/abs/2107.10881>
- Shahzad, B., & Crowcroft, J. (2019). Trustworthy Electronic Voting Using Adjusted Blockchain Technology. *IEEE Access*, 7, 24477–24488. <https://doi.org/10.1109/ACCESS.2019.2895670>
- Shanthinii, S. P., Usha, M., & Prittopaul, P. (2023). *A Survey Based on Online Voting System Using Blockchain Technology* (pp. 209–216). [https://doi.org/10.1007/978-981-19-7169-3\\_19](https://doi.org/10.1007/978-981-19-7169-3_19)
- StarkNet. (2023a). *StarkNet Realease Notes*.  
[https://Docs.Starknet.io/Documentation/Starknet\\_versions/Version\\_notes/](https://Docs.Starknet.io/Documentation/Starknet_versions/Version_notes/).
- StarkNet. (2023b). *What is StarkNet*. <https://Www.Starknet.io/En/What-Is-Starknet>.
- StarkWare. (2023, October 12). *STARK Math: The Journey Begins*.  
<https://Medium.Com/Starkware/Stark-Math-the-Journey-Begins-51bd2b063c71>.
- Szepieniec, A. (2023, August 2). *BrainSTARK, Part 0: Introduction*.  
<https://Aszepieniec.Github.io/Stark-Anatomy/>.
- Tavares, A. (2011). Estudos Eleitorais. In Tribunal Superior Eleitoral (Ed.), *Estudos Eleitorais* (3rd ed., Vol. 6, pp. 1–25).
- Team, P. (2022, July 22). *zk-SNARKs vs zk-STARKs: Comparing Zero-knowledge Proofs*.  
<https://Blog.Pantherprotocol.io/Zk-Snarks-vs-Zk-Starks-Differences-in-Zero-Knowledge-Technologies/>.
- Thibault, L. T., Sarry, T., & Hafid, A. S. (2022). Blockchain Scaling Using Rollups: A Comprehensive Survey. *IEEE Access*, 10, 93039–93054.  
<https://doi.org/10.1109/ACCESS.2022.3200051>
- TruffleSuite. (n.d.). <https://trufflesuite.com/>.
- Vilà Brualla, M. (2022). *Blockchain Layer 2 scalability solutions: a framework for comparison*.
- Vivek, S. K., Yashank, R. S., Prashanth, Y., Yashas, N., & Namratha, M. (2020). E-Voting Systems using Blockchain: An Exploratory Literature Survey. *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 890–895.  
<https://doi.org/10.1109/ICIRCA48905.2020.9183185>

- Voatz. (2023). <https://voatz.com/>.
- vom Brocke, J., Hevner, A., & Maedche, A. (2020). *Introduction to Design Science Research* (pp. 1–13). [https://doi.org/10.1007/978-3-030-46781-4\\_1](https://doi.org/10.1007/978-3-030-46781-4_1)
- Yang, X., Yi, X., & Kelarev, A. (2021). Secure Ranked Choice Online Voting System via Intel SGX and Blockchain. *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 139–146. <https://doi.org/10.1109/TrustCom53373.2021.00036>
- Yang, X., Yi, X., Nepal, S., Kelarev, A., & Han, F. (2020). Blockchain voting: Publicly verifiable online voting protocol without trusted tallying authorities. *Future Generation Computer Systems*, *112*, 859–874. <https://doi.org/https://doi.org/10.1016/j.future.2020.06.051>
- Yang, Y., Guan, Z., Wan, Z., Weng, J., Pang, H. H., & Deng, R. H. (2021). PriScore: Blockchain-Based Self-Tallying Election System Supporting Score Voting. *IEEE Transactions on Information Forensics and Security*, *16*, 4705–4720. <https://doi.org/10.1109/TIFS.2021.3108494>
- Yi, H. (2019). Securing e-voting based on blockchain in P2P network. *Eurasip Journal on Wireless Communications and Networking*, *2019*(1). <https://doi.org/10.1186/s13638-019-1473-6>
- Zaghloul, E., Li, T., & Ren, J. (2021). d-BAME: Distributed Blockchain-Based Anonymous Mobile Electronic Voting. *IEEE Internet of Things Journal*, *8*(22), 16585–16597. <https://doi.org/10.1109/JIOT.2021.3074877>
- Zhang, S., Wang, L., & Xiong, H. (2020). Chaintegrity: blockchain-enabled large-scale e-voting system with robustness and universal verifiability. *International Journal of Information Security*, *19*(3), 323–341. <https://doi.org/10.1007/s10207-019-00465-8>
- Zhang, T., Wang, Y., Ding, Y., Jiang, X., Liang, H., & Wang, H. (2022). Privacy-preserving blockchain-based contract signing with multi-party supervision. *Transactions on Emerging Telecommunications Technologies*, *n/a*(n/a), e4710. <https://doi.org/10.1002/ett.4710>
- Zhou, Q., Huang, H., Zheng, Z., & Bian, J. (2020). Solutions to Scalability of Blockchain: a Survey. *IEEE Access*, *8*, 16440–16455. <https://doi.org/10.1109/aACCESS.2020.2967218>
- Zhou, Y., Liu, Y., Jiang, C., & Wang, S. (2020). An improved FOO voting scheme using blockchain. *International Journal of Information Security*, *19*(3), 303–310. <https://doi.org/10.1007/s10207-019-00457-8>
- Zhu, H., Feng, L., Luo, J., Sun, Y., Yu, B., & Yao, S. (2022). BCvoteMDE: A Blockchain-based E-Voting Scheme for Multi-District Elections. *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 950–955. <https://doi.org/10.1109/CSCWD54268.2022.9776193>
- Zhu, L., Wu, Y., Gai, K., & Choo, K.-K. R. (2019). Controllable and trustworthy blockchain-based cloud data management. *Future Generation Computer Systems*, *91*, 527–535. <https://doi.org/https://doi.org/10.1016/j.future.2018.09.019>
- zkSync Era Block Explorer. (2023). <https://explorer.zksync.io/>.