**Abstract**

This thesis paper aims to spotlight the new cryptographic technologies and see how feasible it is to have an e-voting system powered by the blockchain technologies. The blockchain technologies the thesis will focus on are Ethereum and ethereum virtual machine compatible chains, mainly due to having a consistent view on how smart contracts operate.

The thesis will also go through the history of the  process and benefits of having an e-voting system and go through the history of the process, the pros and cons of electronic voting. We will study the complexity and security needed to implement voting systems and will go in depth on the trade-offs and analyze the blockchain capabilities to perform such a task. The thesis main aim is to look into new EVM compatible blockchains not the ethereum blockchain itself.

The first chapter will take a closer look at the voting practice. The second son will review the literature and the theoretical parts that power the blockchain and a good security for a voting system. The third one will look closer into the blockchain technologies and motivate the decision why Ethereum is not the main priority when building such an application. The fourth one will describe the solution implemented.

**Contents**

**1. Introduction**

**1.1 Generalities**    5

**1.2 Type of voting systems**   6

**2. Cryptographic tools used.**

**2.1 Hash Functions**.  10

**2.2 Digital signatures and public key cryptography**       11

**3. Blockchain technologies**

**3.1 Introduction into Blockchain**     14

**3.2 Ethereum, Smart contracts and Scalability** 21

**3.3 Avalanche** 27

**4. Solution**

**4.1 Backend implementation** 29

**4.2 Gui and user experience.**  31

**5. Other solutions implemented.**

**5.1 Blockchain governance.**    33

**5.2 Conclusions and future work**  35

**6. References**   36

**1. Introduction**

**1.1 Generalities**

Internet voting sounds like a good idea but so far it is not widely used. We have online shopping, online banking, online insurance, but voting proved to be a bigger challenge to tackle than the others. Why the main pillar of our society, Democracy, can't be digitalized also. The main risk is that they can’t be later audited because of the lack of paper trail. Elections have  abnormal requirements almost on the opposite of the spectrum to each other: trust and anonymity.

The main characteristics for a good voting system are the following. Of course it has to be secured and accurate so you are confident the vote you cast it’s the one that is actually recorded, stored and counted. It also has to be immutable so it can’t be changed later by anybody. It has to be anonymous so nobody finds out how you voted. If any of these factors are missing you can’t have a fair election.

It all started when Estonia first introduced and used officially in 2005 the internet voting system also called the i-Voting system. This system was used nationwide for all matters of voting. One of the main incentives the Estonian government  claimed to have is that it would promote voting participations and make voting more accessible to a country with a low population density and extreme weather. Several other countries have experimented with electronic and internet voting but no nation to this day has adopted the i-Voting system to a larger degree than Estonia did [5].

They are able to do this process somewhat successfully mainly because they are already using government issued electronic ID’s. The process is preceded by a pre-voting period where the voters are able to register and log in into a centralized government server. After the vote is cast the voter’s identity is removed from the ballot. Therefore they ensure that the voting is anonymous.

One issue with the internet voting is that it’s more susceptible to outside interference of bad actors. It opens up the voting system to the entire world. Anyone with an internet connection can attempt to attack the system often anonymously. Those types of attacks could be orchestrated by criminal groups or other nations governments so that they interfere and their favorite candidate wins the election.

It should be mentioned that India has developed between 1998 and 2001 their own electronic voting machines. Their electronic voting machine is a simple device used to record the votes instead of using the classical approach of paper ballots. Their machine is quite different and their style of electronic voting benefits from increased security. Their machine is a stand alone one with a basic power connection and no internet connection. Code of the machine is burnt into a microchip. They claim that burn can’t be overridden and it’s immutable. The machine had suffered from some technical downsides in the beginning. Those issues wore mainly because of the offline approach, and the fact if they were to be mass produced their hardware was not top-notch even for their times. The main feature was that a single machine can only store up to 2048 votes for a single candidat mainly due to hardware limitations [19].

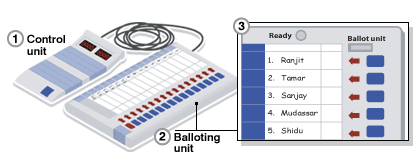


Fig 1 Design the Indian Electronic Voting Machine [15].

**1.2 Type of voting systems**

First we should start with the paper voting system, classic and well tested through the years. It’s a very simple system that it’s quite hard to compromise on a large scale. This type of system usually involves a paper ballot with the names of all the candidates and a stamp. After the paper ballot is marked it is put in a sealed box that doesn’t get unsealed until everyone with a stake in the election has a representative there to confirm the integrity. There are always people from all sides witnessing that the votes have not been altered being that they are there with the sealed box or see a tamper proof seal on the box. At any point there are not a single or a small group of persons put in charge or in a position of trust. There are always a lot of people that keep each other accountable. Even though the paper voting system is so old it stood the test of time.

The system is not a perfect one, but the key feature that still gives it the edge compared to a digitized voting system is that attacks against paper voting systems don't scale. People can be corrupted way easier than machines, but to influence an election so you can swing the results in your favor you need to change a large number of votes and you need to corrupt a large number of people. The larger the number of people you need to influence the more it will cost, it will create more points of failure, therefore the chances of the secret plot to remain a secret decrease.

The second one is electronic voting with a voting machine. The process is inspired and the voter does similar actions to the paper voting system. The machine may or may not be connected to the internet. The voter usually goes into a booth where he has some privacy to cast his vote alone. This is usually realized by a machine who already has the software installed and has all the candidates already validated. In some cases the user also has a paper trail or a punch chard that he stamps or punches the card. This is done mainly due to security reasons and to be able to conduct an audit if issues appear later on so that each vote is counted properly in case of an error or malfunction. In the most optimal scenario the code of the voting machine is open source and tested by independent testers, but in practice the code is usually not open source and the testing is done when an election actually takes place [6].

The main issue and weak points of electronic voting machines is that we need to trust the machine. We need to trust that the original software is actually correct and doesn't have any bugs. We also need to trust that the software running on the machine was not tampered prior to us casting the vote. We also need to trust that they will not tamper with the hardware afterwards so that the number of votes cast is not changed.  There is still one issue, how crect are the votes transferred to a central counting unit that aggregates all the votes and if everything is running correctly there too. There are a lot of points to attack and vulnerable points.

  Electronic voting systems sound like a better idea than the paper system until we look at what actions can a bad actor take and what are the odds of success. Unlike the paper voting system, if a bad actor tries to influence in nefarious ways the election he only needs to find a vulnerability until he succeeds on a grand scale and can dictate the election results. So with limited resources a criminal group can actually hijak the election if they find a flaw.



Figure 1.2 Paper ballots

 Image source: https://newsroom.domtar.com/paper-based-voting

When E-voting systems have failed in the past. The Belgian officials experimented in the beginnings of the 2000’s with electronic voting. On May 18th, 2003, in Belgium voters went to the polls to cast their vote. In many regions, Schaerbeek being one of them, voting was done on a computer, something the Belgian Government had been experimenting with for years.  All the voting systems had a backup. The voter would choose on a computer their vote, the vote would be casted and they would also use a magnetic card that they would drop in a box before exiting for redundancy. Later when counting the votes the people in charge found out there were discrepancies. The total number of votes was higher than the amount of people who voted. After recounting, using the magnetic cards, they found out that Maria Vindevoghel, a small candidate received more than the recounted votes by a total of 4096.  After extensive analysis of both the code and the hardware no issues were found. They were not able to replicate the error and the machine seemed to be working just fine. The conclusion they later arrived to was that the 13-th bit that stored the number of votes ( 212) got flipped randomly.

Sometimes a good old CPU might not be enough. In the late 70’s Intel reported some strange errors popping up in their new 16 kb DRAM on their processors. Some bits would spontaneously flip with no apparent cause. The issue was later discovered to be the ceramic packing of the cpu. Intel has built a new manufacturing plant on the Green River in Colorado. That manufacturing plant turned out to be downstream of an old uranium mill. Intel has discovered that the issue was caused by traces of uranium and thorium found in the ceramics. Those atoms were enough to cause small radiations and pushing a couple of electrons can cause random bit flips. So without using ecc memory which is arguably way too expensive to be used in regular voting machines, bit flipping incidents may occur inevitably from time to time.

Internet voting is a form of electronic voting that is purely online.  The strongest point about internet voting systems is the fact that they are accessible to pretty much anyone with an internet connection.  Besides an internet connection the systems usually need either an ID-card or a Mobile-ID. They levredge the security of the latest digital signature techniques so they can maintain a good level of confidence and security. The way the Estonian government implemented the system is by developing both a mobile application and a web application. After the identification process the voter just selects a candidate and then casts the vote. The vote is usually sent to a central server and after the election is over they will be counted [12].

The weak point is also that they are accessible and this way can be attacked and corrupted by everyone. The fact that the system is open to the entire world comes with terrible risks. The attackers usually can atack if form any place on the and often they can maintain their anonymity. The first type of attacks aim to corrupt the voter's own computer which on average has poor security. Even if the process is fully impenetrable, malware on the voter’s machine can lead to that machine secretly casting a different vote even if all the visual components would say otherwise. This is a big issue, because a considerable percentage, about 20%, are already corrupted and some are part of botnets [16]. Those attacks usually steal the voter secret key, digital identity and other personal components. The second type of attacks focus on corrupting the central servers that count the casted votes. These machines can either be corrupted by an inside job or a powerful attacker that exploits vulnerabilities to gain access to the system. Usually they try to corrupt the decryption process so that some votes would go to their favorite candidate instead of the chosen one.  One other security flaw is that intensive testing is never done. Usually the only full testing the systems have is when an election actually happens. Even if we try to use checksum so that we may try to check if the software running on the backend server is not compromised, that checksum can also be forged too and we are just moving the issues around. One last type of attack is on the vote itself when it is in transit from the voter’s machine to the counting server. Man-in-the-middle attacks are still relevant even to this day and may prove quite effective if the system can’t fully trust the voter’s end.

**2. Cryptographic tools used**

**2.1 Hash Functions**

 Generally hash functions are fast operations that take a set of inputs of any arbitrary size and fit them into a table or other data structure that contains fixed-size elements. A hash function is also an injective function, known as one-to-one function a function f that maps distinct elements to distinct elements; that is, f(x1) = f(x2) implies that x1 = x2 and the inverse is also true f(x1) is different than f(x2) implies that x1 is different than x2.

The first hash functions were designed in the late 70’s. There have been a couple of papers that discussed the need and usage of this type of functions in public key cryptography. Diffie and Hellman recognize the need for a one-way function to ensure the correctness and the genuineness of a digital signature. The first definition proposed by Rabin was a design with a 64-bit result based on a block cipher DES. During the 90’s the number of hash functions grew swiftly. (6) The main usage in the computer science field of hash functions is for storing sensitive data like passwords. Storing them in plain text is very dangerous if a security breach happens. The login mechanisms actually fetch the hashed password data and compare it to the hash of the user password and verify if there is a match. This way the “real” password is not sent over the login request and makes it harder for attackers.

While minimizing collisions is good practice when using hash functions in other areas, minimizing or even trying to remove them altogether is a must when we use hash functions in cryptography. This property is called collision resistance and it’s the most important one when we discuss about hash functions in cryptography. Other requirements for hash functions introduced in the literature in the beginnings were the properties of preimage resistance and second preimage resistant functions [8].

Preimaging resistance means that it should be computationally infeasible to find any input that maps to a given element. Preimage resistant hash functions require that the length of the output should be at least 90 bits [9]. Second preimage resistance refers to the fact that it must be computationally infeasible to find any second input which has the same output as a given x. So given x it is impossible to find y using just brute force such that f(x) = f(y). If a function is collision resistance that implies that it’s already 2nd-preimage resistance.

Each hashing algorithm has 2 phases: preprocessing and the actual hashing itself. Preprocessing involves 3 steps:padding the message so it lowers the chances of that function being vulnerable to length extension attacks, parsing the padded message into blocks and setting the initial hash values  [18]. The purpose of the padding is to make sure that the message is a multiple of that function input value (256 , 512 , 1024). After the padding is done then the result will be parsed and split into m-bits blocks so that the initial hashing computation can begin. So before the actual computation can take place the initial hash value needs to be set. For example the SHA-256 initial hash value consists of the following eight 32-bit words in hexadecimal.  After the initial hash values are set the actual hashing process can take place.

In this paper the main focus of this chapter will be the sha-256 hash function, because it was the first hash function used in a blockchain consensus, the proof-of-work consensum of the first blockchain, Bitcoin [7].

**2.2 Public key cryptography**

Public key cryptography was created in the 70’s also. The creation mainly was because there were 2 big issues: How to exchange secret keys between two people if they have never met before and can’t meet physically and How can you validate that a message is from a reliable source and has not been tampered with in the transit process. The answer to such problems, the public-key cryptography. A system in which keys come in pairs and they are the inverse to one another. So a user can encrypt a message using the public key and only the owner of the private key can decrypt it. The inverse was also true, anything encrypted with the private key can be decrypted with the public key. Another very important property of those keys is that if given the public key it should be infeasible to try to discover the private key [3].

The separation of the encryption and decryption processes was a brilliant idea. One person is able to send private messages to another one by just looking up that individual public key and encrypting the message using that way. One issue is that if every individual has his own pair of keys it will become very hard to practically store them. The total number of keys in a network is: n \* n-½, where n is the amount of pairs. This would result in almost half a million values stored for a thousand subscribers and about five hundred trillions for a million users.

Public key cryptography is an asymmetric type of cryptography where we use a pair of keys. It uses the public key to encrypt the message and then only the person that has the private key can decrypt it. The process of public-key cryptography ensures that the information has not been tampered by a third party within the process of transferring the message from one party to another. It also validates the message's authenticity in the case of when the sender is the owner of the private key. The receiver can check the validity of the message itself by using the public key of the sender. This way a public key system can provide an authentic digital identity for a user. In our blockchain context this is the way we secure transactions within the blockchain itself. This is essential in a point-to-point network like blockchain where nodes and miners do not trust and know each other.

* **Public key algorithms**

We will start with the most popular one: Rivest-Shamir-Adleman(RSA). It is an asymmetric algorithm and was launched back in 1977 and named after the initials of the creators Ron Rivest, Adi Shamir and Leonard Aldeman. The RSA algorithm has 2 parts: one that encrypts and decrypts messages and one that generates the private and the public key.

The algorithm for generating the keys is the following:we select two prime numbers that are considered to be large. We calculate their product and call it p. We find the totient function ϕ(p) = (a-1)(b-1) and we find a integer e that is in the interval [1,ϕ(p)] and it’s coprime with ϕ(n). We have the public key (p,e), then can calculate the private key d = (k\*ϕ(p) + 1) / e for a integer k.

The main reason why RSA got so popular is that it made the encryption and decryption process straightforward. The algorithm strength is only as strong as the keys themself. If the 2 primes are small this will leave the system exposed to attacks. The effectiveness of the RSA comes from the fact that it is hard to factor a very large number into 2 large primes, so the larger the length of the modulus, the more time it will take for an attacker to perform a brute force attack by computational power.

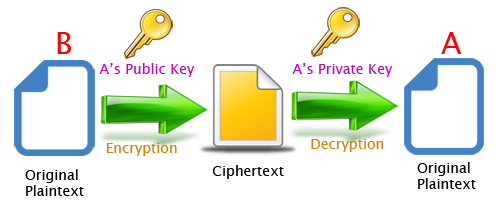


Figure 2.2 Asymmetric algorithm encryption and decryption

<https://community.ibm.com/community/user/ibmz-and-linuxone/blogs/subhasish-sarkar1/2020/06/23/understanding-the-rsa-asymmetric-encryption-system>

* **Role of the public and private keys**

A private key is usually a string of characters that is used in either the encryption or decryption. In our blockchain context, Bitcoin uses private keys that are 256-bit characters, both letters and numbers. For Ethereum a private key is a string of 64 random hex characters. On all blockchains the procedure of generating a key is an offline one, meaning there is no interaction with the blockchain.

To generate your address on the Ethereum network you need to hash the public key and get the first 20 bytes. As a conclusion, the role of a pair of keys is the following: the private key creates a signature and the public key verifies the signature.

**3.Blockchain Technologies**

**3.1 Introduction into Blockchain**

It all started in 2009 when Satoshi Nakamoto published his now famous paper “Bitcoin: A Peer-to-Peer Electronic Cash System”. His view was to create a new system that would leverage public key cryptography and the decentralized computing power to validate and store transactions.All those transactions are broadcasted to all the nodes that participate in securing the network. Each node collects and works to find the solution for the proof-of-work problem that is unique to the transactions that will be added in the next block. When a validator finds a solution to that proof-of-work problem it tells all the other nodes and that validator gets a reward for being the first to solve that block. The network would produce a block about every 10 minutes and a block would contain a big chunk of transactions.Each block would be of reasonably small size, at around a megabyte each. We can split all types of blockchains in 2 categories: *proof-of-work* blockchains and *proof-of-stake* blockchains.

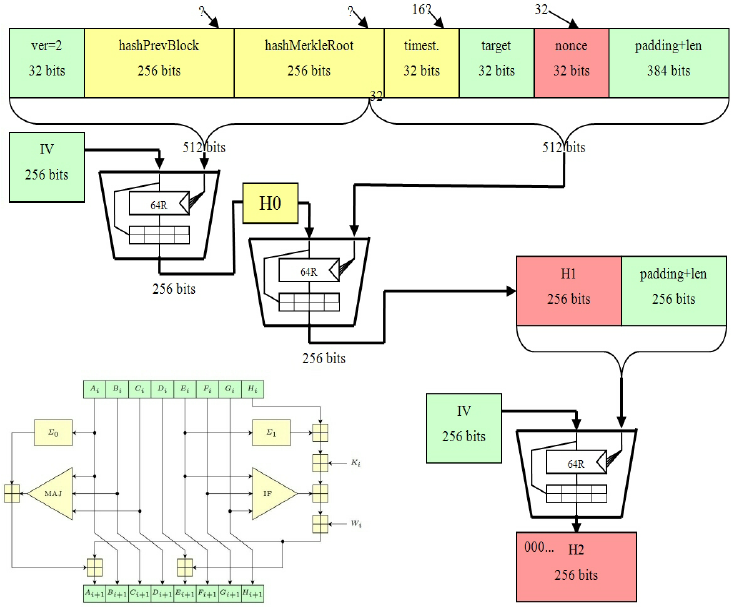


Figure 3.1 Bitcoin Hashing Functions in depth [27]

* **Proof-of-work.**

 We will start exploring the proof-of-work(POW) types of blockchain mainly because Bitcoin first popularized it and it’s the most prominent. Pow makes it that everyone is voting on the consensus mechanism using their computing power.  Bitcoin uses a hash based proof-of-work, employing the SHA-256 hash function to maintain a dynamical difficulty so no matter how many validators and how much computing power it is thrown at the blockchain, it will always self stabilize and maintain the difficulty. The concrete mechanics is that the participants should have the output of the hash function have at least a number of zeros at the beginning of the resulting block hash. This process is done by brute force, the miners start by trying all possible combinations and use their computing power to calculate the SHA-256 hash so that they are the first to find the result. After finding a result and broadcasting it to the network, other validators can compute the hash and verify that the hash computed is correct. Each block is produced at a block interval timing and that timing is usually in the range from seconds to minutes (ethereum has a block time of about 14 seconds and BTC a block time of about 10 minutes).This block interval outlines the delay of the blockchain. The smaller the interval the faster the transactions are processed and added to the blockchain ledger. There is a direct correlation between the difficulty of the POW and the block time and the size of a block, the harder the difficulty the more time it takes for a validator to find a solution.

 There is also a crucial correlation between the block size and the time. Usually the faster a block is produced the smaller it is. Here arises the inevitable question. Why not have a low difficulty so we have a small block interval and have a bigger block size so the network can process more transactions and be actually useful. Well there are 3 pillars in the blockchain dilema. A blockchain can have 2 out of 3 of the following: speed , decentralization and security. Bitcoin has a good decentralization and a very strong security but for all intents and purposes it’s very slow. If the difficulty would be reduced and the block size increased there will be a faster blockchain and arguably less secure one. But in terms of decentralization, that would be gone, because in order to have a decentralized blockchain it should be accessible to everyone participate. If the block interval is reduced and the block size increases, the size of the actual ledger of all transactions would rapidly expand to terabytes of size. This would make it impossible for everyone except very wealthy individuals and corporations to validate transactions, because in order to validate a block you must have all the previous ones on your machine locally stored [2]. The Pow security relies mainly on the assumption that no one individual can acquire more than 50% of the total computing power for prolonged periods of time. Some models that make it harder to corrupt or manipulate the blockchain for personal gains if you have temporary more than 50% computing power are: that there are a number of block confirmations before a transactions is accepted by 3-th party merchants so that the attacker can’t cash out the illicit gains and the transactions would later be reverted and the mining cost usually exceeds the block rewards and it would be financially nonoptimal to attack the blockchain if the illicit gains would be smaller than the value needed to be invested in order to perform such atack.

* **Proof-of-stake.**

Secondly we will discuss the proof-of-stake type. The main concept in proof-of-stake(Pos) is that the blockchain is incentivized to be secured by those who have a financial interest in it.  The concept was created as a countermeasure to the well known vulnerability of the Pow blockchains that can be manipulated with a so-called 51% attack.

The Pos blockchains are different from Pow in regards to security and energy consumption. The security level of the blockchain is not increasing energy consumption as more and more people try to secure the network or use the network. The main difference is that a node is no longer capable of mining and can only validate and mint new blocks. The new proces is incentivised to promote decentralization, so everyone can be a validator if it has a minimum amount of coins. This way miners don’t clump together in mining pools that

Given the fact that the validator needs to look up the coins it is in his best interest to behave correctly. Most Pos blockchains have a slashing mechanism implemented. This mechanism cuts the amount of coins that is staked by a bad behaving validator or someone that tries to corrupt the network. Compared to Pow, Pos validators don’t compete to compute the hash function the fastest and actually are selected randomly based on their stake compared to the total of the coins stacked. If we want to replace the already well tested proof-of-work model then the proof-of-stake should be at least at the same level in terms of security. If we think about switching to a proof-of-stake we should be able to prevent bad actors from creating counterfeit chains. In regards to the selfish mining behavior, the bad actor can’t influence what block he is going to mint, thus mitigating this type of attack. For a 51% attack to be successful in a proof-of-work based model, the attacker must keep an alternative chain secret. They then execute transactions and try to cash out using 3-th party merchants. Once the cash out is successful, they broadcast the longer secret chain which will invalidate the original transactions and he will get his transactions reversed, then the attacker gets his coins back like the cash out didn’t even happen.

* **Zero-Knowledge proofs**

Zero-knowledge proof is a protocol method that allows a prover to prove that a statement about a secret message is true without actually revealing the secret itself.  Zero-knowledge proofs is a way of performing authentication that allows for no passwords to be exchanged, this way the password can’t be at risk of being stolen. This is useful because it makes your communication so secure and protected. ZKP allows you to prove that you know some secret or many secrets to somebody at the other “end” of communication without actually revealing it. The very term “zero knowledge” originates from the fact that no (“zero”) information about the secret is revealed, but the second party is convinced that the first party knows the secret in question [22].

A zero-knowledge proof has the following three properties: Completeness(if both parties are honest the verifier will always be convinced), soundness that if the statement is false no cheating prover can actually be verified as honest and zero-knowledge meaning the verifier learns nothing about the secret.

* **Mining and staking**

The bitcoin system has created a distributed decentralized network. Each block contains a unique block hash and uses in its hashing process the hash of the previous block. The first block mined in the bitcoin network was called the genesis block and had a preset hash for the previous block as 0. Each block contains all the transactions and an address for the miner to be credited the reward. A miner's earnings are compounded by the block reward and the transaction fees. While the block rewards are a set and only change in the event of the “the halving”. The halving is the event in the Bitcoin network that is happening each 210000 blocks and happens each 4 years where the block rewards are cut into half.  The other part of the miners earnings are the transaction fees or so called “tips”. The miners are incentivized to include the transactions with the highest tips first in the next block so they maximize profits. This way an order is maintained so that users can speed up their transactions by offering a higher than usual tip so their transaction will be included in the next block or they can set a standard amount and the transaction will be processed when the network has a lower trafic than at that time.

 In the early days of the Bitcoin blockchain inception you could mine bitcoin and expect to get a block reward even if you were using a PC. These days bitcoin is a business and the mining process is usually a business.

The original plan of Satoshi was to create not an invulnerable security system, but one that would be too difficult and too expensive to try to take advantage of. Basically they tried to make the cost of a successful attack 51% as expensive as it gets. In the beginning it was feasible to mine alone using just a PC, but as time went on and more people started mining the chance to actually mine a block was getting lower and lower. While mining alone grants the miner complete ownership of the rewards, the actual odds of succeeding are very low, bordering the impossible. Therefore, given the fact that the cost of mining is usually very high for an individual, mining alone it’s not profitable. Given that economical context some people created and joined mining pools. A mining pool is a group where miners put together their mining resources and later share the respective rewards equitably. This makes rewards more consistent and miners don’t run at a loss for a very long time. Each mining pool usually has instant block payouts or pays out when the pool finds a block so it gets a reward. The mining pools require less individual work in terms of hardware and power and give constant rewards to the miner so it makes it more profitable to mine on the long term.

At first the mining pools sound like a perfect idea. In practice they are as all things not even remotely close to being perfect and meaby even detrimental to the network itself. While increasing profitability and consistency of rewards, the miner itself gives up on the autonomy he has in the mining process. This practice undermines the decentralized principle of the blockchain. So if the mining operator manages to get a big share of the total hashing power and has nefarious thoughts he may try to use all the mining power to attack the network. If one mining pool starts controlling even as little as ⅓ of the total hash power of the Bitcoin network, the node can act malicious and try to do what is called selfish mining.

What is selfish mining? Selfish mining is when a pool finds blocks and keeps them private therefore intentionally forking the blockchain. The honest miners will continue to mine on the public chain and the malicious ones on their own private one.This way the malicious miners find more and more blocks and build a lead over the public blockchain. If the public chain starts catching up to the private one the private one will release blocks and maintain its lead and collect all the rewards. This way will make it very hard for the honest miners to waste resources mining and hashing on blocks that will serve no purpose and will make it impossible to break even or even to make a profit if all the block rewards are manipulated and go to a malicious group.

This strategy has a critical impact over the results of mining and mining rewards and breaks the incentives for most miners to act honest. Thus this strategy is not a 100% guaranteed way to have a majority over the blocks produced on a blockchain, but this strategy yields very high rewards compared to the others. If a selfish mining pool reaches a threshold where it can hold a private chain that is longer than the public chain it will have very high financial rewards and will attract other miners that want to have the higher earnings. If the pool becomes larger and larger it will increase in size towards a majority it can perform a 51% attack.

The proof-of-stake inherently may suffer from the same vulnerabilities but can change the protocol to completely eliminate selfish mining  and make it almost impossible to perform a double spending attack. In regards to the 51 % attacks, the POS may implement a self regulating mechanism called “slashing”. This is when a bad behaving node has a portion or all of the cryptocurrencies that it has staked taken from it and usually burned or redistributed. Even if more than 51% of the cryptocurrencies staked, act maliciously they will eventually be slashed and will no longer have a majority [4].

* **Mining hardware evolution**

As intended by Satoshi himself, the first miners were the CPUs. The 2009 era high-end CPUs like intel i7 990x managed to output 33 megahashes/s. In 2010 CUDA-based Gpu miners entered the market, shortly followed by the first OpenCL miner. The technology that made mining a real competition was the ASIC miners [13]. First launched in the beginning of 2013 the BITmain Antminer S3 had a power of 478 Gigahashes/s and consumed 366 W of power. With each new generation of ASIC miners the previous one becomes obsolete and not competitive, because the more people start mining with the new and improved ASIC miners the more hash power they provide at better effective cost thus driving the previous generation not profitable. In may of 2022 the latest Bitmain Antminer S19 has a capacity to produce around 200 Terahashes/s and has a power of around 5000 W [10].



Figure 3.1.2 Mining Facilities in Kazakhstan

Source: <https://news.bitcoin.com/kazakhstan-shuts-down-over-100-crypto-mining-farms/>

* **Environmental concerns**

It’s been over 10 years since cryptocurrencies mining began. The competitive nature of proof-of-work and the fact that the miners tend to gravitate to the places with the lowest energy cost has raised environmental concerns given the fact that energy produced from burning fuels or coal it’s usually the cheapest. Until around 2021 about 70 % of the hash power came from China [20]. It’s estimated that more than a quarter of the total mining rigs used non-renewable energy. Proof-of-work mining raised concerns about incentivizing the high polluting and cheap power plants. In 2021 the bitcoin network is estimated to have consumed on average more electricity than a lot of medium sized countries.

Although the high energy consumption some people have argued that having a constant energy consumption is good for the producers and for the power plants, so they have increased revenues and can invest in renewable energy sources that have a constant power output such as: Solar, Wind, Hydro Power, Geothermal energy, Tidal energy and Wave energy.

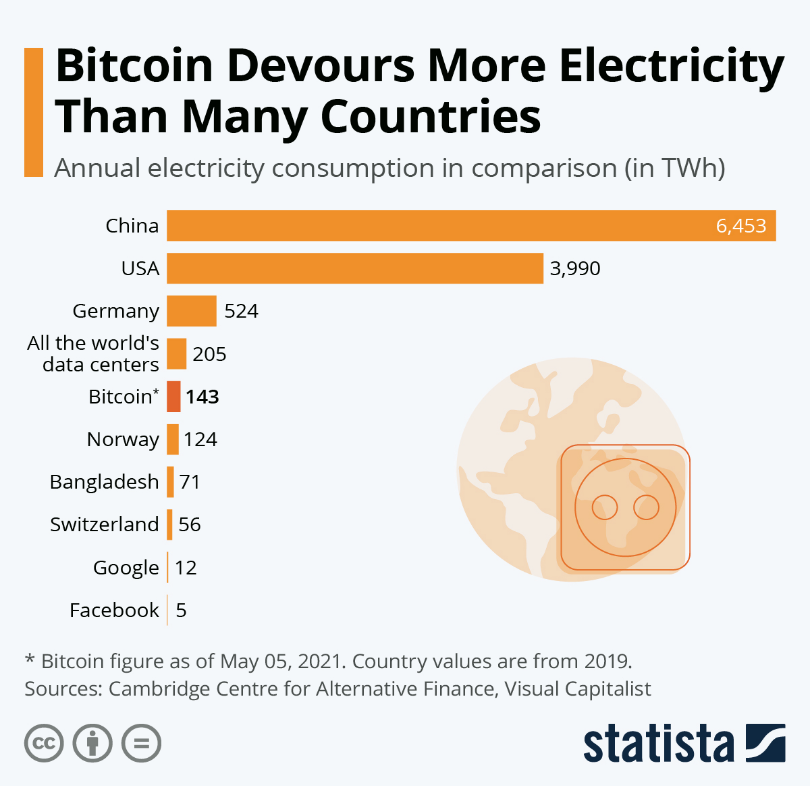


Figure 3.1.3 Bitcoin Energy consumption compared to other countries [21]

**3.2 Ethereum, Smart contracts and scalability**

ethereum is the go-to blockchain when we talk about smart contract functionalities. ethereum is currently a proof-of-work blockchain and currently has the most popular Dapps(Decentralized applications). The initial whitepaper was published in late 2013 by Vitalik Buterin. Ethereum has eight co-founders:Vitalik Buterin, the current face of the Eth foundation, Mihai Alisie, Anthony Di Iorio, Amir Chetrit, Charles Hoskinson, Gavin Wood, Jeffrey Wilcke, Joseph Lubin. All eight met in Zug Switzerland in 2014 and by 2015 they had a working prototype.

During the years the developers have maintained and upgraded the protocol a few of the most important upgrades are the Byzantium, Constantinople and London. In the London upgrade there was an Ethereum Improvement Proposals(EIP), EIP-1559 that aimed primarily at changing the mining process and incentivising the usage of the network as well as incentivizing users to hold the ETH currency by trying it to make it a deflationary asset. An important aspect of the EIP was how it changed the transaction system and introduced a token burning mechanism that burns more the more the network is used [14].

An important network upgrade that is taking place right now in the summer of 2022 is the ETH 2.0 upgrade. The project was pioneered and the first steps were launched in 2020 Beacon Chain genesis.  This upgrade is the biggest in all the history of Ethereum, it aims to change the proof-of-work consensum to a proof-of-stake, and to introduce sharding. It also claims that this upgrade will solve the current scalability and affordability issues Ethereum is currently facing. It will aim to increase the normal transaction speed to an incredible 100000 transactions/second. This will indirectly lower the premium(the tip)  the users pay to the miners to have their transactions speed up and included as fast as possible in the next following blocks. All those will be the benefits of the new shading technique.

Sharing is the action of splitting the blockchain in multiple pieces or so called shards, by storing the data in different places. This way the actual computation strain put on every computer compared to the output increases. The data moves faster and the transaction speed is increased as a result. In each shard, each node that is part of a particular shard must have the ledger of that particular shard and not the entire ledger. It is possible for a node to function properly without having all the ledger data. This is a nice improvement because with added transaction speed we increase exponentially the speed that the ledger grows. Sharding also adds to a blockchain the ability to perform parallel execution, so 2 shards can validate different transactions simultaneously. Compared to the linear model the sharding model adds a lot of value to a network that fails to scale up in the traditional way.

Another way that the scalability problem was tackled is by creating Layer 2 architecture. A Layer 2 also abbreviated as L2 is a protocol that builds on another blockchain. That blockchain that is built upon becomes known as the Layer 1 or is sometimes called the main chain and the layer 2 is considered a side chain. The target of a layer 2 is usually to help with the scalability or the lack of it. A couple of examples of layer 2 architecture is the Ethereum Layer 2, Polygon and the Bitcoin Layer 2 known as Lighting Network.

The most popular EVM compatible side-chain is Polygon, also known as Matic. Being an EVM compatible blockchain, it benefits from the fact that the ecosystem and developers don’t have to start from scratch in developing new applications and don’t need to learn a new and complicated programming language specific for that blockchain. This makes integration with the main chain much easier and facilitates new adoption. It acts as a plasma chain, a blockchain that has its own separate transactions and consensus mechanism; it employs a Pos mechanism even though the ethereum blockchain is inherently POW. The Polygon blockchain talks to the “main” blockchain and each couple of blocks it records the proof that the transactions that happened on the side chain are actually correct and they are added to the main Ethereum blockchain. This way the Layer 2 scaling solution can fully benefit from the security of Layer 1 and provide actual scalability for users with no big concerns. The Polygon scaling solution has a couple of more layers beside the one we talked about: The Security layer, it provides a “Security-as-a-Service” model that either leverages the Ethereum security directly or directly uses the security provided on the own blockchain by dedicated validators. It also has a network layer and an execution layer. The network layer is mainly responsible for block production and can manage the local consensus systems and the execution layer is responsible for executing the transactions.

* **Pros of Layer 2**

Polygon and mainly the layer 2 architecture has a lot of benefits:It is the fact that we mention already, the compatibility and interoperability with the Ethereum blockchain that facilitates growth and development. A convenience that makes Polygon popular even with not so technical people like it because of its consensus mechanism. It's a more environmentally friendly blockchain that has a Pos consensus thus no mining operations are performed and no power is used to run the nodes.(The nodes still consume power but it’s a way smaller in comparison and can be neglected).

Another benefit is that  Polygon is scalable and this implies that it’s actually affordable. On Ethereum lately it costs you an arm and a leg to do something that has actual size. A basic transaction to move funds from a user address to another user address can cost up to 50 $ if the network is clogged. A contract interaction can set you back as much as 200$ and if you want to deploy a smart contract on the blockchain the price can go up to thousands of dollars.  Transactions on the Polygon network can be as low as 0.01 dollars. So developers can build things that eat more network power without being afraid to run out of gas.

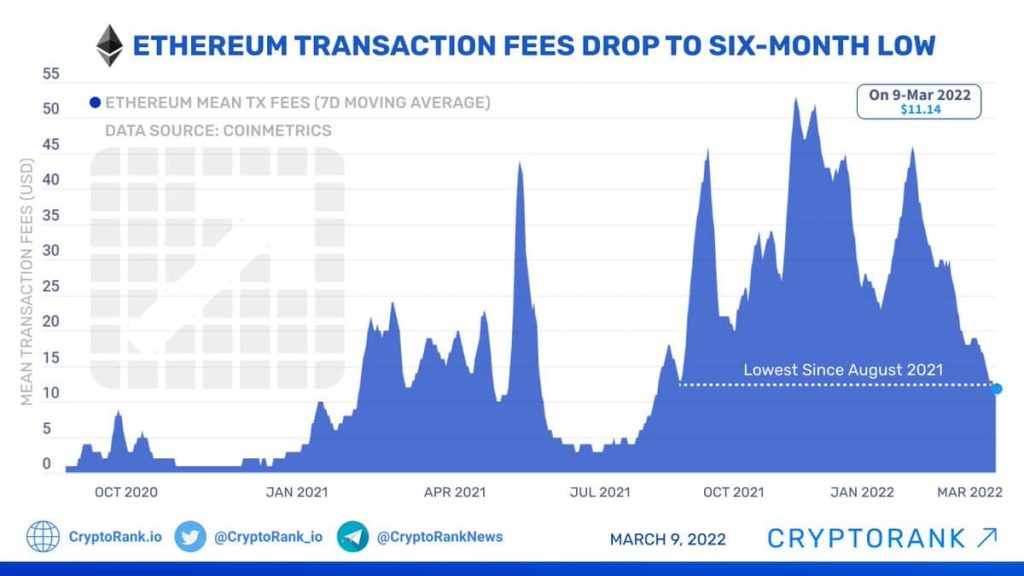


Figure 3.2 Ethereum transaction fees measured in USD [24]

* **Issues with Polygon**

The Polygon architecture does not ensure that the network layer will keep producing blocks no matter what, the software behind it it’s not considered robust enough and it’s mainly criticized in regards that it relies too much on Ethereum and the security it provides it’s not actually enough. There is a reason why the transactions are this cheap. It’s not clear what the flaws of the protocol are but we can’t be 100% sure that it’s safe to be used in a real voting application.

There is also a concern that if the Ethereum 2.0 upgrade can deliver on all the promises the Layer 2 of the Ethereum would be way slower than the actual main chain itself and the Polygon blockchain can become obsolete.

The Polygon network has suffered outages although it’s been a couple of years since they happened. We can't be sure that some bugs won’t be exploited again if the protocol proves that it’s not fully robust.

* **Smart contracts**

The concept of a smart contract was introduced to the public in the Ethereum inception and whitepaper and later crystallized in the implementation of Solidity. Solidity is a high level programming language and was created in 2014 to be used in coding of smart contracts that run on an EVM. It took inspiration from other curly braces programming languages such as C++ and Javascript. It is statically typed, meaning the variable types are explicitly declared and the language can determine if the syntax is good at compile time and throw an error, preventing flawed code to be deployed on the blockchain occupying space without any purpose. It also supports object oriented patterns such as inheritance.

The language has had a lot of changes in the past years, a few of the major releases were the 0.5, 0.6 0.8 releases. They all added important new key features but also maintained the interoperability with the older deployed contracts.

 Smart contracts are immutable, verifiable programs that run and are stored on the blockchain that are executed when some predetermined conditions are met. The data is stored in the distributed ledger of the blockchain. They are used to automate the execution of a contract or an agreement so that everyone can be certain that the outcome will be the predetermined one. It can be used to bind 2 persons to an agreement that will happen automatically once certain conditions are met. On the Ethereum blockchain the contracts are stored as bytecode and are considered public code. The contracts follow a “if/ when then” rule that governs the transactions. The contract once deployed is not controlled by humans, but rather its own internal code so it’s impossible to be manipulated. So when criteria is met the contract can automatically broadcast corresponding transactions. One useful example is the automation and usage of smart contracts in the lending market with such a protocol called DeFi (decentralized finances) that automates the process of borrowing and lending.

They can be used to store small amounts of data, like a hyperlink, but storing anything larger than a few kilobytes it’s not economically viable. As of today the cost of storing one megabyte of data on the ethereum blockchain costs at least 20000 dollars.

Smart contracts have been used in a few real world applications, namely a Digital identity, Loans and Mortgages and most importantly Financial Database Recording, in the end all transactions are public after all. The most success has been seen in the Financial sector that the smart contracts created, the decentralized finance protocols that have value in the hundreds. The most popular DeFi application is Aave protocol. It is a decentralized Liquidity provider. This is a fancy name for a place where people can lend and borrow currencies in a secure manner. The most used feature is called the flash loan. It allows a user to borrow a very high amount of ETH with no collateral and no restrictions. The only catch is that the borrow transaction and the repay transaction must happen in the same block so people don’t run off with the money. This has created both a place for people that have a good amount of currency to get a stable return and also a place for people with not a lot of capital to access large amounts of leverage so they can profit from even a small margin or a small arbitrage trade.

Smart contracts can only act upon the information that is already stored in the ETH blockchain and the transaction that they received. They can’t validate off-chain information so for that we use third party external services called Oracles.

* **Scaling issues with Ethereum.**

While providing adequate security and a very good decentralization, the Ethereum blockchain fails in one regard. The scalability issue. It’s currently expensive and very slow.  The average transactions per second is around 10-15 and around 2-6 smart contract interactions per second. This is a big issue because for the Ethereum blockchain to be used for a real world workload it should have the TPS in the hundreds at least. Currently the transaction delay is way too big. If we actually use blockchain for every step it will take hours to use a basic feature. Even if the application is nice and security is at an unbelievable level, if it takes a couple of minutes to get a response after we click something the applications will fail.

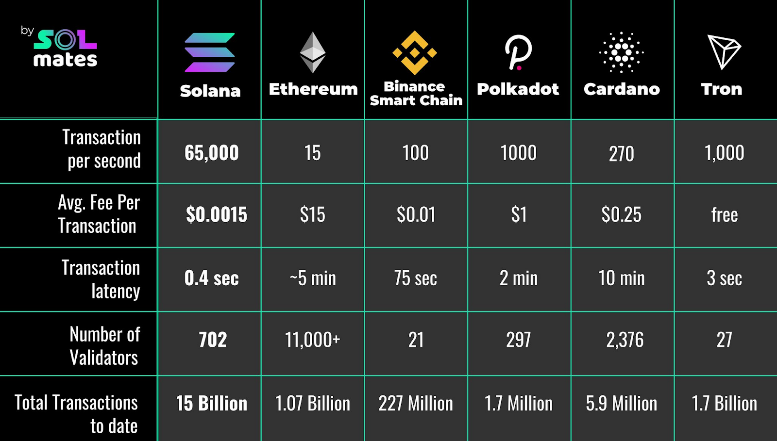


Figure 3.2.1 Solana Speed comparison Source: [**https://break.solana.com**](https://break.solana.com)

* **Other Solutions**

As we see in the graph the Solana Blockchain is a good candidate for an application. It has very high transactions per second, small fees and very low latency. It seems like the perfect blockchain. Solana employs the POS consensum layer as well as a protocol known as proof-of-history that allows the mainet nodes to have only a part of the main ledger to validate the transactions not requiring that a node must have all the transactions in the ledger. The POS also helps the environmental concerns when we talk about scaling so the carbon footprint of the blockchain is much smaller than ethereum’s.

One concern Solana has even though all the previously enumerated pros, is that it still considers itself a “beta” project. This is not a bad designation, but the stability of the beta projects is a concern. In January 2022 the blockchain was down for a couple of weeks and the network had a big outage not producing blocks at all for almost 10 days.

The Outage problems persisted in March and April were small one day incidents. The incidents forced the nodes to perform a full restart to start producing blocks again and bring the blockchain back online. This is the main reason why the blockchain can’t be used for critical operations yet but it’s a good candidate. Not to mention that Solana uses a different approach to Ethereum and the two are not compatible. (Red days means that the network was not online at all and yellow means that it was partially down, while green means it run without any issues)

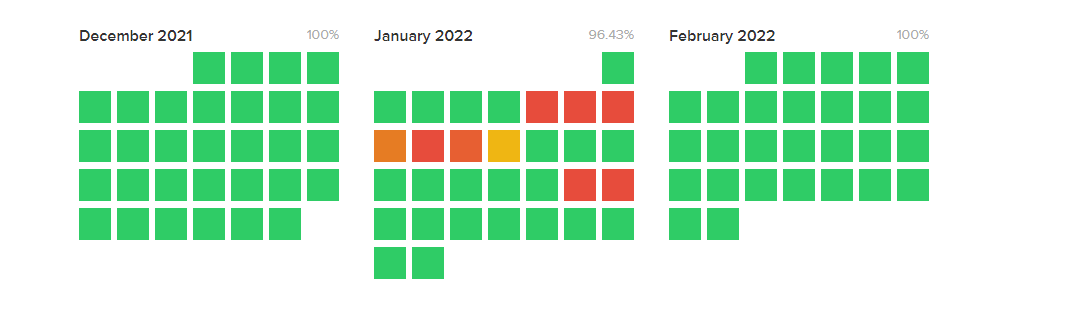
****

Figure 3.2.2 Solana network outages. Source:<https://status.solana.com/>

**3.3 Avalanche**

Another promising blockchain that it’s pretty big and has a good transaction speed is Avalanche. Avalanche claims to be a high performing customizable, scalable and secure blockchain. They use a POS protocol but they are able to write smart contracts in the most popular programming language in the blockchain space so they promote adoption for people interested in the space as Solidity runs natively. They use a democratic process based on the amount of AVAX token you have, tokens that allow you to participate in validation of the transactions and allow you to vote on new proposals.

The Avalanche protocol it’s a new approach to the blockchain problems. They have created the Snow Family of Protocols.”The Snow\* protocols operate by repeated sampling of the network. Each node polls a small, constant-sized, randomly chosen set of neighbors, and switches its proposal if a supermajority supports a different value” [11].They offer 3 built in blockchains each with its own purpose.We have the Exchange Chain, this chain acts as a decentralized marketplaces for creating and selling digital assets, The platform chain is the chain that coordinates the validators and enables the creation of new subnets.The Contract chain(C-Chain) it’s the one we will mostly focus on, it’s the chain where all the smart contract operations happen. Overall Avalanche has a very high transaction speed at around 4500 transactions per second, way faster than Eth but still not comparable to Solana.

The transactions are affordable, around 1 dollar. The transaction finality is fast 10 seconds.

The application will not be deployed on the real avalanche network, even though the transaction speeds are fast and the fees are low, the price is still considerable when trying to develop an application.The application will be deployed on one of the testnets of Avalanche network. The initial Pick is the Fuji network, the main one recommended by Ava Labs. The testnet is the same as the main net but you don’t carry the risk of losing real funds unlike the main chain. The testnet is powered by the same technologies and software, usually it is on the same version as the mainnet or on a new version that is still compatible with the old one.

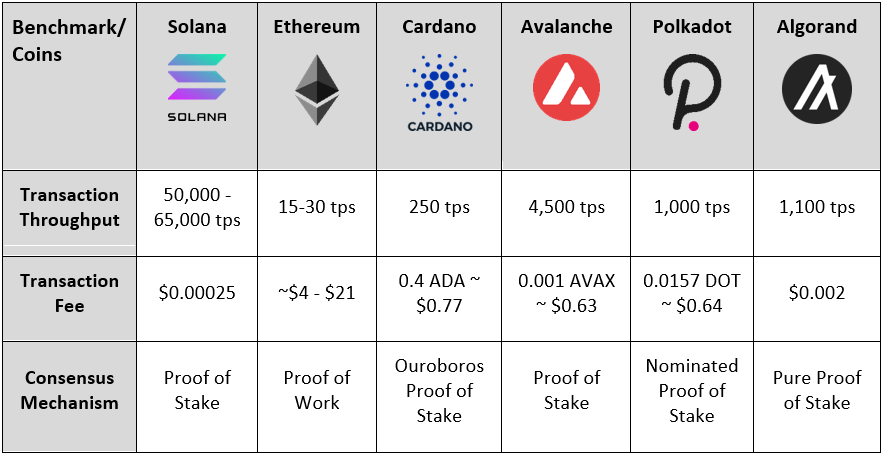


Figure 3.3 Avalanche network financial metrics compared to the rest of the top blockchains

**4. Solution**

**4.1 Backend implementation.**

The initial design of the application aimed to be a blockchain app. The aim was to make it decentralized and blockchain only. While developing and researching the voting topic is more of a sensitive topic than approximated. It requires the highest security and trust and when the question about how we are going to secure the application from outside interfaces arrived the full on decentralization model appeared to not work for this purpose. After this we have settled down to a more classical approach that would levredge the blockchain strengths and still have the security of a classical application. On the blockchain every transaction is public but the identity of the user is anonymous.

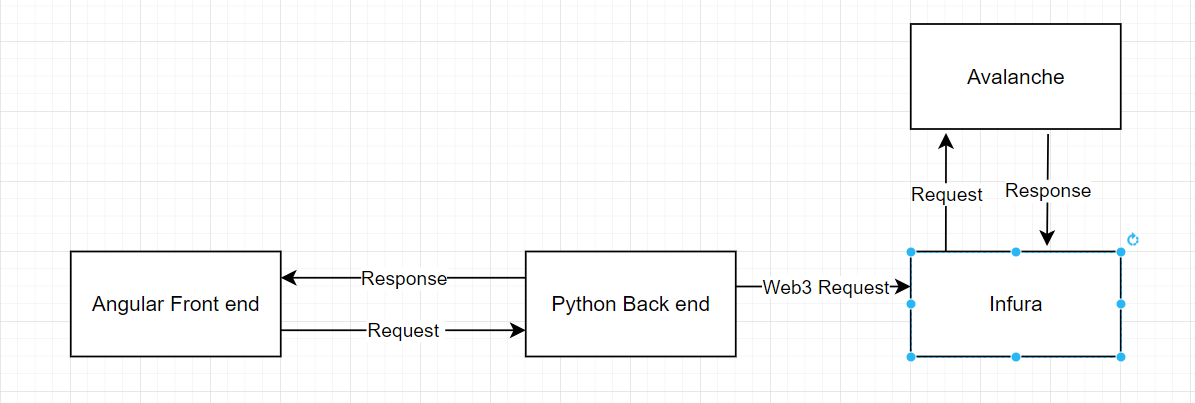


Figure 4.1. Application Design Diagram

The frameworks used in the implementation are: Solidity and main deployment on the Avalanche C-chain, on the testnet, the Fuji network. On the backend side Python is the main continent because it has the most developer support for the Web3 packaged and is the most popular one. On the front-end Angular was our pick, a reliable TypeScript framework.

In the design cycle of the application we’ve taken into account the correctness of the application. We have modularized the application into 3 parts that work concurrently. The smart contracts can work independently from the python backend and the backend can work independently of the front end.  The modularization of the entire application will lead to components always being small in size, which make the code more maintainable and can help mitigate big security aspects.

If an administrator needs to do some changes the changes can be done directly in a block explorer that allows the smart contract calls for example in <https://snowtrace.io/> we can call all the read and write functions of a smart contract as long as we have the privileges.

On the backend side is it worth mentioning that we’ve used a postgres database to store the sensitive data, like the security code. We don’t store the security code itself, only a hashed version of it. There were concerns raised alongside the development that the current census that was happening in May 2022 in Romania where you only used the security number to log in and access all the questions, the security number was only hashed once. The security number is not composed of random digits in Romania, from 13 digits only the last 4 are purely random. The others denote the gender of a person, birthplace code, birthdate and other metrics. The concern was that in total if a person knew all the information it could compute the 9999 hashes for the supposed security number and see the communications.

In this regard the application will use a salt after the security number so we can mitigate the preimage attack. The hash will have the form of sha256(security number + secret salt).  After that just to be extra safe we will hash the final result again. This process is crucial to secure a trustful login process, because without it the whole application will not be that dependable.

After the login process is done the user has full access to the application. After that the user will receive as a response a bearer token. Without that all the other requests would fail, the backend endpoint being protected by JWT(Json Web Token) authentification.

* **Voting Features**

The approach we ended up settling on about the voting system was the following. We created referendums and elections. In a referendum the people can only vote yes or no based on a question. The voting contract logs the hash of the person that voted and stores it permanently on chain.This way a person can be verified to vote only once and the voter is still anonymous, and after the vote is cast the data is recorded permanently on the ledger. The voter can thus rest assured that the vote will be counted correctly. The election voting process is a more complex one. We still use all the security mechanisms as in the referendum process but we will add a couple of new features. The voter will not have a simple question but instead will be listed a position or the stakes of the election (Mayor of Cluj or Senate of Romania). Alongside with the position the candidates or the parties will be listed and the voter can just click and submit the vote.

* **Security and Validation**

The initial request for the backend must be the login request. After that the user will receive as a response to the token request a token which will be used in the header of the next requests in the field bearerToken. The generation is automatic and the availability of the token is 60 minutes. We consider that this is enough time for a user to cast a vote and make an informed decision.

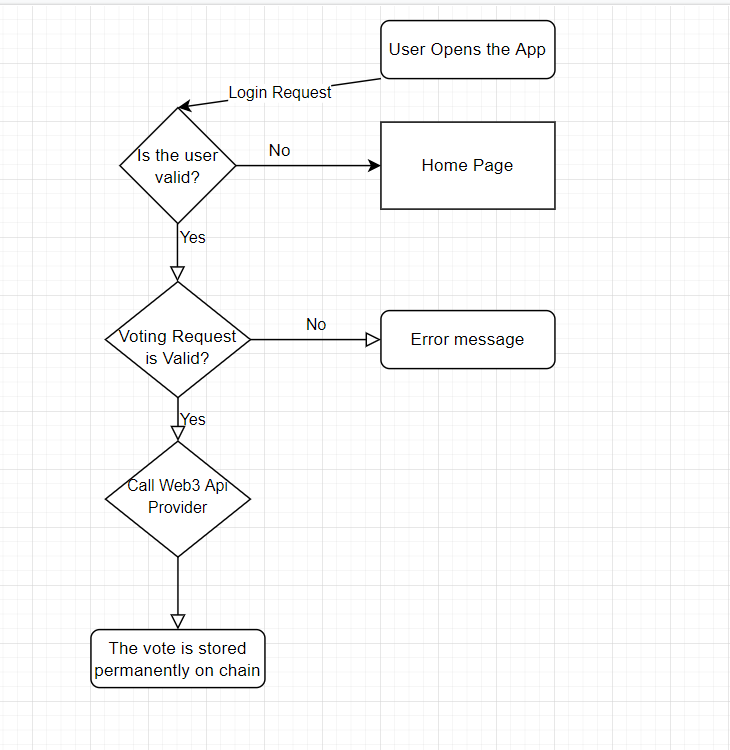


Figure 4.1 Voting Flow

**4.2 Gui and user experience**

The main focus of the application was to provide an adequate security model for the voting process. Alongside the development cycle the resources were mostly focused on the solving of other issues and the user experience. As for the user experience a few points that the development focused on were accessibility. So anyone that has an internet connection and a device can navigate the application. Another important matter that was taken into account during the development was to have a clear and understandable hierarchy and flow. So users always know where they are and where they can go next.

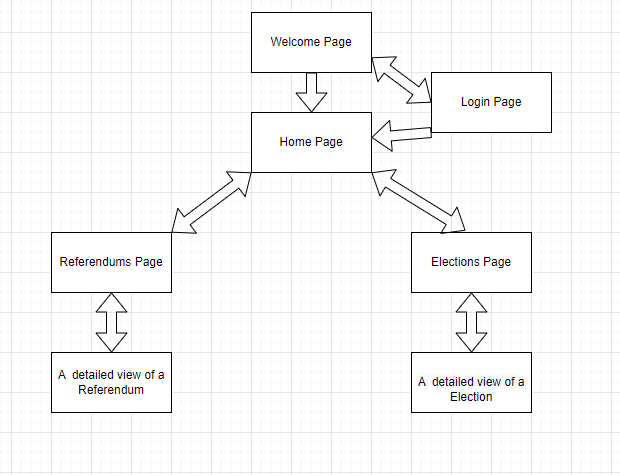


Figure 4.2 The Application pages and components and the way a user can navigate through the application.

* **Security**

As a security feature the application has guards to prevent ill intent and try to mitigate accessing endpoints that you are not allowed to. The only pages that the user can access without logging in and getting a token from the backend are the page and the welcome page.

Any attempt to access another endpoint is promoted to an error page that will print the message that the page the user wants to access is impossible. The error code returned is 403 meaning the server understood the request but it did not authorize the request.

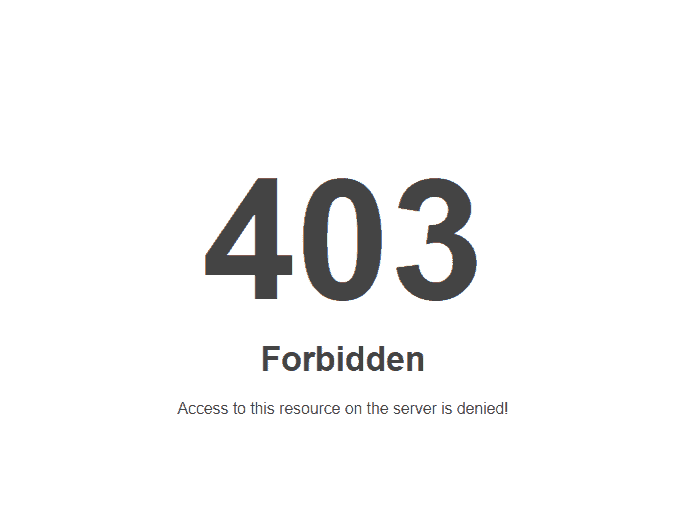


Figure 4.2.1 Error page when a user attempts to make malicious requests

**5. Other solutions implemented**

**5.1 Blockchain governance**

One of the first successful blockchain projects that implemented a good governance system was Uniswap. Uniswap is a decentralized cryptocurrency exchange. It facilitates automated transactions between users, taking the role of an Automated Market Maker. This way it incentivizes users to provide liquidity to the protocol and have as low as possible operational fees [26].  They claim that the protocol is publicly owned because it’s governed by the community. Their process of governance is based on their own token ownership, the UNI Token. One user can vote on any proposal based on the amount of UNI tokens that the user’s wallet had at the time of the snapshot, the block where the proposal was submitted.

The process can be outlined in 3 easy steps: A user submits a proposal. The user then gets in touch with the moderators responsible for the Uniswap community. After that the user submits a forum post and publishes a snapshot. The voting time for a proposal is between and has a few requirements. The initiator needs to look up at least 1000 UNI tokens. For the proposal to pass there should be at least 25000 UNI tokens that voted.  A couple of proposals that were recently passed were the deployment of the Uniswap protocol to the Polygon blockchain and a new way to leverage the liquidity pools for lending and borrowing [1].

The initial distribution backs the story, more than half of the tokens are owned by the community. The developers only have about 21 % of the total voting power.

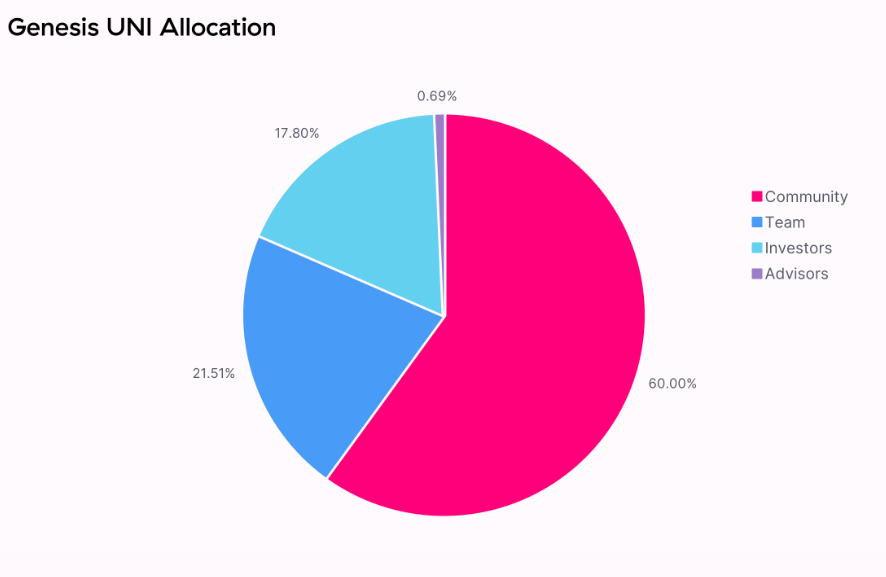
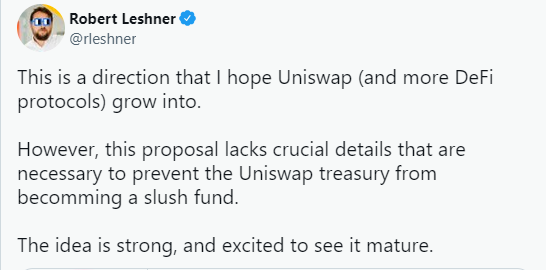


Figure 5.1 Initial UNI token distribution

Since an incident where a big whale created a proposal to allocate Uniswap development funds to their own cause and then voted massively in favor the voting power of big entities has been reduced so their power is lowered in some cases.

In the summer of 2021 the Harvard Law department tried to create a proposal  to allocate to themself around 10 million dollars over a vesting schedule of 4 years and pledged around a million Uni token in support. The users quickly reacted negatively and voted against the proposal ending it and passing a new proposal that states that the treasury funds will have stricter allocations rules.

The practice of having a community centered governance process is in principle a good one. But we need to keep in mind that the people dabling their savings in cryptocurrencies are not the wisest and the most popular beliefs can lead to tragic outcomes.

Another protocol that had the same community gouvernance methods failed in a spectacular way. That protocol is named Anchor. It keeped having proposals that increased the yields of the users and of course everyone became a victim to their own greed and voted in favor. The protocol was not designed to do that and when the yield reserves started to dry up a real “bank run” happened and people started to cash out and the money was not there.

Due to faulty founders and bag comunity proposals a coin that was meant to be pegged to the US dollar went from 1 dollar to 0.01 dollars in the span of weeks.



Figure 5.1 The crash of a stable coin UST.

**5.2 Conclusions and future work**

In this thesis we have introduced the concept of a blockchain system. It is a good alternative in a couple of specific cases, mainly referendums or low stakes elections.  As blockchain technologies evolve and the security behind it evolves it may be possible in the future to try to digitalize the voting procedures. The only network that has a security that is good enough for our topic is the security provided by the Bitcoin network. The network is a smart contract environment by definition but it is a simple and primitive version and it is not Turing complete and we can’t create smart contracts that would be capable of performing such an endeavor.

The main player to win the web3 race is Etereum. Even if the Ethereum Virtual machine has Solidity which is a more powerful Programming Language. Ethereum suffers from way too many scalability issues, not to mention huge transaction fees and a very high transaction finality. We discussed a couple of promising protocols that have raised very high hopes in solving that. Solana managed to solve the issues but has a constant battle to keep the blockchain online. Avalanche on the other hand, had lower expectations in terms of technologies but was stable in the last year without any outages on the main net.

The application presented serves as a proof of concept that web3 applications can find a purpose in our environment, but we will need the technologies that power it to mature and start competing with real world applications.

As for future development we can create a better design for the user experience and we can stress test the smart contacts so we can simulate real world events.

**6. References**

[1]  Hayden Adams,  Noah Zinsmeister  Dan Robinson Uniswap v2 Core Available: <https://uniswap.org/whitepaper.pdf>

[2] V. Buterin, “Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform,” 2013. [Online]. Available: <https://www.ethereum.org/pdfs/EthereumWhitePaper.pdf>

[3] WHlTFlELD DIFFlE The First Ten Years of Public-Key Cryptography

[4] Eyal, I., Sirer, E.G. (2014). Majority Is Not Enough: Bitcoin Mining Is Vulnerable. In: Christin, N., Safavi-Naini, R. (eds) Financial Cryptography and Data Security. FC 2014. Lecture Notes in Computer Science(), vol 8437. Springer, Berlin, Heidelberg.

<https://doi.org/10.1007/978-3-662-45472-5_28>

[5] D. W. Jones and B. Simons. Broken Ballots: Will Your Vote Count? Stanford University Center for the Study of Language and Information, 2012.

[6] Sanjay Kumar et al. / International Journal on Computer Science and Engineering (IJCSE) Mr. Sanjay Kumar1 Department of Computer Engineering, M. M. University, Mullana (Ambala) 133207, India Dr. Ekta Walia (Professor & Head)2 Department of Information Technology, M. M. University, Mullana (Ambala) 133207, India

ANALYSIS OF ELECTRONIC VOTING SYSTEM IN VARIOUS COUNTRIES

[7]  M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In Proceedings of the Twenty-first ACM Symposium on Theory of Computing, 1989. 378, 386

[8] Preneel, B. (2010). The First 30 Years of Cryptographic Hash Functions and the NIST SHA-3 Competition. In: Pieprzyk, J. (eds) Topics in Cryptology - CT-RSA 2010. CT-RSA 2010. Lecture Notes in Computer Science, vol 5985. Springer, Berlin, Heidelberg.

[9] Robshaw M.J.B. (2011) One-Way Function. In: van Tilborg H.C.A., Jajodia S. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA.

[10] Rogaway, P., Shrimpton, T. (2004). Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In: Roy, B., Meier, W. (eds) Fast Software Encryption. FSE 2004. Lecture Notes in Computer Science, vol 3017. Springer, Berlin, Heidelberg.

[11]  Kevin Sekniqi, Daniel Laine, Stephen Buttolph, and Emin Gun Sirer Avalanche Platform 2020/06/30 Available at : https://assets.website-files.com/5d80307810123f5ffbb34d6e/6008d7bbf8b10d1eb01e7e16\_Avalanche%20Platform%20Whitepaper.pdf

[12] Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat Harri, Hursti Margaret, MacAlpine J. , Alex Halderman, Security Analysis of the Estonian Internet Voting System

[13] Michael Bedford Taylor, The Evolution of Bitcoin Hardware University of Washington 10.1109/MC.2017.3571056

[14] G. Wood, “Ethereum: A Secure Decentralized Generalized Transaction Ledger,” 2017. [Online]. Available: https://ethereum. github.io/yellowpaper/paper.pdf

[15] <https://aceproject.org/electoral-advice/archive/questions/replies/130400041>

[16] <https://securelist.com/it-threat-evolution-in-q2-2021-pc-statistics/103607/>

[17] “Solidity — Solidity 0.4.21 documentation.” https://solidity.readthedocs.io/en/v0.4.21/

[18] Federal Information Processing Standards Publication 180-2 2002 August 1 Announcing the SECURE HASH STANDARD

[19] Prototyping of Indian Electronic Voting Machine -A step towards ASIC in voting IJERD JOURNAL

[20] Cryptocurrency & Its Impact on Environment

International Journal of Cryptocurrency Research ISSN: 2790-1386 Int.J.Cryp.Curr.Res. 1(1) (2021) 1-4, DOI: <https://doi.org/10.51483/IJCCR.1.1.2021.1-4>

[21]<https://www.statista.com/chart/18632/estimated-annual-electricity-consumption-of-bitcoin/>

[22] https://hackernoon.com/eli5-zero-knowledge-proof-78a276db9eff

[23] https://e-estonia.com/story/

[24] <https://cryptorank.io/performance>

[25] https://break.solana.com/

[26] <https://uniswap.org/governance>

[27]Nicolas T. Courtois, Marek Grajek and Rahul Naik,  The Unreasonable Fundamental Incertitudes Behind Bitcoin Mining