UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

TÍTULO: GENERATION OF NATIVE TOKEN USING BLOCKCHAIN AND DISTRIBUTION ON FAUCET

Trabajo de integración curricular presentado como requisito para la obtención del título de Ingeniero en Tecnologias de la Informacion

Autor/a:

CHAMORRO GARCIA BRYAN MOISES

Tutor/a:

Phd - JULIO ARMAS

Co-Tutor/a:

Phd - PINEDA ISRAEL

Urcuquí, Diciembre del 2022

Autoría

Yo, **BRYAN MOISES CHAMORRO GARCIA**, con cédula de identidad 1724440514, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la autor/a del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, Diciembre del 2022.

BRYAN MOISES CHAMORRO GARCIA

CI:1724440514

Autorización de publicación

Yo, **BRYAN MOISES CHAMORRO GARCIA**, con cédula de identidad 1724440514, cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación

Urcuquí, Diciembre del 2022.

BRYAN MOISES CHAMORRO GARCIA

CI:1724440514

Resumen

El desarrollo de este trabajo se enfoca en el analisis, diseno y contruccion de una plataforma que sirva de guía y experimentación en ambientes de blockchain, para que más personas puedan tomarlo como inicio en el desarrollo de proyectos y desarrollos científicos. Para este caso se a utilizado la arquitectura blockchain basada en Proof of History (PoH), una prueba para verificar el orden y el paso del tiempo entre eventos. PoH se utiliza para codificar el paso del tiempo sin confianza en un libro mayor, una estructura de datos solo para agregar. Cuando se usa junto con un algoritmo de consenso como Prueba de trabajo (PoW) o Prueba de participación (PoS), PoH puede reducir la sobrecarga de mensajería en una máquina de estado replicada tolerante a fallas bizantinas, lo que resulta en tiempos de finalización inferiores a un segundo. La implementación ha utilizado los recursos de la red de Solana con entornos de comprobación como clústeres y nodos que aceptan solicitudes HTTP utilizando la especificación JSON-RPC 2.0. Esta implementación han hecho que se desarrolle de manera más dinámica y rápida la faucet para distribuir el token nativo creado. La criptomoneda se desarrolló y registro en la red de Solana de una manera rápida por medio las devtool de Solana. Solana ha creado todo un ambiente de desarrollo para que pueda integrarse a la blockchain. Por medio del comand line (CLI) se pueden generar native token y añadir al nuevo proyecto por medio de dos lenguajes principales como Rust(es un lenguaje de programación compilado, de propósito general y multiparadigma) o Java Script por medio de @solana/web3.js. Se han logrado implementar el faucet en dos clusters, garantizando por etapas el testeo (Testnet) y producción final (Devnet). Dando como resultado final un web faucet con capacidades de conectar wallets, enviar tokens nativos, registrar y abstraer del bloque generado información, y como último mostrar todas las transacciones realizadas a detalle.

Palabras Clave:

Blockchain, Proof of History, Proof of Work, Proof of Stake, fallas bizantinas, JSON-RCP, Solana,Rust.

Abstract

The development of this work focuses on the analysis, design, and construction of a platform that serves as a guide and experimentation in blockchain environments so that more people can take it as a start in developing projects and scientific developments. For this case, the blockchain architecture based on Proof of History (PoH) has been used to verify the order and the passage of time between events. PoH is used to encode the trustless passage of time into a ledger, an aggregate-only data structure. When used in conjunction with a consensus algorithm such as Proof-of-Work (PoW) or Proof-of-Stake (PoS), PoH can reduce messaging overhead in a Byzantine fault-tolerant replicated state machine, resulting in lower completion times of one second. The implementation has used network resources of Solana with testing environments such as clusters and nodes that accept HTTP requests using the JSON-RPC 2.0 specification. This implementation has made the faucet to distribute the created native token more dynamic and faster. The cryptocurrency was quickly developed and registered on the Solana network using the Solana Devtool. Solana has created an entire development environment to be integrated into the blockchain. Through the command line (CLI), native token can be generated and added to the new project through two main languages, Rust or Java Script through @solana/web3.js. The faucet has been implemented in two clusters, guaranteeing the testing (Testnet) and final production (Devnet) in stages. As a final result, a web faucet with capabilities to connect wallets, send native tokens, register and abstract information from the generated block, and finally show all the transactions carried out in detail.

Keywords:
Blockchain, Proof of History, Proof of Work, Proof of Stake, JSON-RCP, Solana,Rust.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

In the early 1960s, one of the most important inventions of all time called the Internet was developed [1]. The Internet has been a structure for spreading information. In 1960 when computer network terminals were replacing perforated cards facilitating time-sharing and online editing programs [2], There was some initial difficulty in the experiment because the cables that were used to generate the Internet information were owned by the telephone company [2]. Then in the late sixties the Advanced Research Projects Agency Network (better known by its acronym ARPANet) was created and initially connected four universities in the United States: the University of California (UCLA), the Stanford Research Institute, the University of California at Santa Barbara (UCSB), and the University of Utah. In addition, it can be said that the ARPANet was idealized as a military communication system using a cycle compartment of networked computers with the TCI/IP protocol technology [1]. The connotation of the Internet from the beginning is simply the transmission of information but with the characteristic that this is something exclusive for a limited number of people. Universities and the United States government made large investments of money that created systems and entities for the development and innovation of the Internet, for example, BITNET. The BITNET system was essential for developing the networking system between universities. Another important factor that marked a new era in 1981 was Nobel physicist Kenneth Wilson's theory on supercomputers [3].

The eighties are also characterized as the time of microcomputers; they were fashion-

able as substitutes for secretaries since they had emulation terminals and email and text software [3]. Computer scientists ran into various problems during their investigations. One of the main problems that affected the network was the computer's speed [4]. The solutions were translated into physical equations, which made massive parallel computing possible. Professor King said that massive parallel computing was the best way to solve complex problems. Thus, a new era of parallel computers began, spearheaded by Nobel laureate Kenneth Wilson and his research center. In this period, the type of protocol for supercomputers and networking was chosen: once again, TCP/IP was chosen with the support of academics[5]. At that time the Internet was presented as a very basic structure, but it took a long time to develop and invest. Without a doubt, it is a path in which they did implement many efforts, but one of the most important events in that a representation of this entire conglomerate of institutions, universities and companies that we're developing it was sought. Since there was an imminent regularization by the government of the United States, in mid-1988, Al Gore, former Vice President of the United States, championed the idea of a computer network, connecting everyone in a chain, and championed this idea in his National Information Superhighway project [6]. His Internet democratization project was approved in 1991 in the US Congress and noted that Al Gore defended progressive ideals concerning new technologies throughout his political career. With the implementation of the Internet, more connections were made in a more efficient and better commercialization way [7].

The ARPANet, which had been configured during a great period of its development, has ceased. The MILNET and the Defense Data Network have continued for the US Department of Defense. In contrast, the NSFNet, used for public purposes, was later replaced by Internet Service Providers. In addition, the emergence of applications for the World Wide Web can also be seen in this period. With the creation of the Web at CERN by Tim Berns-Lee it was possible to develop browsers, which has caused trade wars between Netscape and Microsoft. Starting in the second half of the 1990s, applications were created to improve email, the Web, instant communicators such as ICQ, and MP3 data transmission [8]. Only between 1995 and 2001 the Internet began to be an instrument used by the financial market and also several Internet companies had their trading indexes in the stock market. Currently Internet innovation occurs every minute. Kurose and Ross

believe that there are three merits that deserve special attention [9]:

- The proliferation of high-speed network access.
- Ip Security.
- P2P networking.

Access of high speed networking especially via cable and DLS (Digital Subscriber Line) has revolutionized the means of communication enabling users to use video, television and make phone calls. In addition, the field of security is an important issue, as there is an increasing number of attacks in cyberspace from denial of service attacks to worm and virus infections. Finally, P2P networking is an application that explores the resources of its own users on the Internet and allows users to intercommunicate, the most famous of which were Skype and Kazaa. Without a doubt, the Internet began to be more than a tool for use or experimentation, it began to be a fundamental part of commerce, economy, and also known as a right [10].

In 2009, during the presidency of Barak Obama, or better known as from the perspective of the United States, the Internet is connoted as "But none of this progress would be possible, and none of these 21st century challenges can be fully met, without digital infrastructure of America" [11].

The implementations, the new forms of use and development have focused on a new mechanism that also has implications for society. In the area of human relations, the distancing of space and the characteristic of the "instantaneous" have made it possible to bring people closer together. In fact, people who live in different countries have the ability to communicate through applications such as Facebook, Skype, and MSN, among others, also contributing to maintaining personal networking [12]. Currently, it is seen that the Internet replaced the use of the telephone in many countries, but not the use of the mobile phones; since there are much more people with mobile phones than laptops due to the fact that it is cheaper; multinationals have invested more in mobile phones that are capable of connecting to the Internet.

However, the Internet facilitated in such a way the interaction of information between people since they have made the revolutions and promotion of strikes and other social

movements. The ease of grouping that occurs on platforms such as Facebook provides that such movements are more organized when the Internet is not censored [13]. Among them, we can mention the events that occurred during the Arab Spring.

The Internet is now a useful tool for knowledge, which was the original aim of American teachers [14]. The Internet is a global network of billions of computers and other electronic devices. The Internet is an aggregation of various histories: **official, oral, of the oppressed, etc.** It is a space where all the sciences of the past and the present meet, and everyone can consult it. In a general way, the pioneer teachers achieved with their expectations to be possible the Internet as an academic world as well.

Like any human space, the Internet is also apt to commit illegalities such as crimes that move from reality to enter the plane of virtuality, such as crimes of human trafficking, theft of money online, fraud schemes, contracting and etc [15]. The Law itself represents the transformations of society, and its essence must always be dynamic. The Internet is also changing the laws to adapt to the new norms of society, called Computer Law. However, it can be seen that countries are increasingly concerned about cyber crimes, and in the international sphere they are trying to use the existing legal structure to avoid new negotiation processes, since the internal sphere is approving its own laws and some countries. They are considering Internet access as a Human Right. The Internet was intended by its founders to be a decentralized system to prevent the entire system from being overthrown in the event of problems or attacks. In this way, similar to the International System, the current Internet can be characterized as an anarchic space where there is no official central body to represent it.

However, as the Internet is also a space that generates power: economic, political, military power, and perhaps the power to influence the entire Internet society, there is a concern on the part of the countries to control [16]. As the Internet was created mainly with the encouragement of private initiative and its innovation depends directly on them, the lobby they exert against the ambitions of state control is enormous. It is believed that the Internet space should e maintain their freedom according to their initial project, since pure state control would weaken their ability to innovate and also to avoid uncontrolled interference by states such as censorship. An antecedent to be able to better represent the reasons or justifications for why interactions with the Internet should be controlled is

due to the attack as an act of vandalism by the pro-Russian group in Estonia that was nervous about the removal decision of a statue that symbolized the former USSR [17]. This group of vandals started attacks on the Estonian cable system for a duration of three weeks. The result of these attacks was the fall of the entire Internet system in Estonia: banks, companies, TV channels, and websites. Governments were all overthrown, enormous economic losses occurred during the lack of Internet in the country [18], and since the cable system also has a connection to the NATO system, this organization became concerned with the impact of the attacks. In the end, the Estonian government blamed Russia, claiming that it was related to the attacks and the group. However, as it is impossible to identify those responsible for the attacks due to Internet failures, the Estonian government searched for a culprit and sentenced himself. It is believed that it was only an act of symbolic injustice. The cyber attacks in Estonia were followed by others, such as the Stuxnet and the Flame virus, which will be seen later [18].

One of the fathers of the Internet, the American Vinton Cerf, said in several of his lectures that when the Internet was developed, they opted for it to be a space without central control and also thought about the fact that they were not directly identified people, because, as the military would use it, they could have problems in case one country accused another of attacks, espionage, etc [19]. With this background in mind, the important thing is that the numerical series in bills that aim to contain counterfeiting and protect intellectual property began. It is called the Stop Online Privacy Act – SOPA, but it was rejected in 2011. People against this Law were in defense that it went against freedom of expression and innovation [20]. Vinton Cerf said that the Internet as an information exchange platform would exist without intellectual property registries so that it would be a free space of knowledge. However, in a letter, SOPA explained why this Law did not work and was only going to bring prejudice “I continue to have concerns regarding the efficacy and wisdom of this legislation”[21].

SOPA explain 3 ways in which rights are violated. First, attempts to manipulate DNS (Domain Name System) will reduce the utility of DNS as our chief mechanism for locating sites, and encourage abusers to adopt alternative mechanisms, such as IP address lists. Second, clients of the infringing content can readily change their DNS settings to utilize offshore DNS resolvers. Third, sites dedicated to infringement have many options for evad-

ing these measures, such as registering multiple domain names with offshore registries in order to stay ahead of court orders. Fourth, falsifying responses to domain name resolution requests will compromise the "downgrade resistance" of next-generation improvements to DNSSEC, because systems that do not receive a signed answer from a resolver will fall back to accepting unsigned responses to resolve a domain name [22]. Here is evident how various actors that have helped in the evolution of the Internet, have different interests.

Thus, the future of the Internet will not be defined by a lack of technology, vision or motivation. One of the main concerns that arise from all these data can be thought of what will be the future of the Internet and how it will change. But from a more personal perspective it is mainly considered how his develops

1.2 Problem statement

With the advancement of the internet it is a goal to maintain a safe and reliable source of development, search or storage. Therefore, technological means have been generated to ensure its operation, these problems also are called ASO problems of authentication, integrity, and nonrepudiation, which are solved by digital signatures. The definition of a Digital Signature is a "mathematical scheme which ensures the privacy of conversation, integrity of data, authenticity of digital message/sender and non-repudiation of sender" [23].This problem is solved by a third party that certify the origin and the trustworthy. Normally the Digital signature uses a key factor of digital signature and various methods and procedures. But when a third party certifying authority is used, how can we assume that this can not be altered or can be attacked by others?. The Digital Signature is widely used to different platforms and proposes by the characteristics that this provide, as privacy, authentication, integrity, non-repudiation. These are also backed by encryption methods that make them very secure. There are many uses for digital signatures, but several difficulties do not allow us to use this technology or preserve the security chain. There are different emitters for a digital signature that lacks the origin trust. The user can give access or a password to others, or the electronic signature is not in possession of the client. But this can differ according to the application of the technology[24].

One of them is called PKI (Public Key Infrastructure) which in a few words is a set of

laws, roles or policies that are required in the creation, distribution, management and use, which makes it essential for its use in any type of data exchange that is confidential digital certificates become the heart of PKI. But something fundamental to be able to complete the reliability of the PKIs are the dates on which they are generated. Yet the digital signature guarantees nothing about the time when the document was signed: the timestamp requires trust in the party that signed it. In the case of financial transactions and other forms of legal contracts, time is of the essence, and the order of those financial transactions needs to be independently certified to be auditable[25]. We can give an example in which we want to sell a house in which there are two buyers who do not want to buy the property, whoever buys it first will be the owner, but what happens if the time of purchase is not considered within this type of transactions, this would be considered a scam in which the seller is committing a scam due to the fact that he could have made a double sale without taking into account who is the person who first obtained the property. The same thing can happen in a bank where we deeply trust the transaction process and that the tellers do not steal our money. What is deeply worrying is the fact that the only ones who have access to these transactions are the banks and that only they can modify them without our knowledge. Currently, Blockchain is mentioned mainly as a trust in decentralization systems[26]. Blockchain is based on the problem of creating a distributed storage of timestamped documents where no party can tamper with the content of the data or the timestamps without detection.

Within the development of smart contracts, this process is closely linked to people who know about the subject and are aware of the technology. Many people who do not know about this technology and its implementations cannot use them or the qualities of each blockchain. Also, for both audiences, the fee paid to be part of the chain has not been developed an effective method that can calculate how much it consumes and the cost it entails to register agreements with said blockchain.

The values vary from the base Cryptocurrency that the network has and the file size it intends to be added to or from new policies that can add difficulties or increase operations with this technology [27].

The term to define how much is going to cost is called Storage Rent Economics. Each transaction that is submitted to any ledger imposes costs [28]. Transaction fees paid by the submitter, and collected by a validator, in theory, account for the acute, transactional,

costs of validating and adding that data to the ledger. What has not been considered in this process is the mid-term storage of active ledger state, necessarily maintained by the rotating validator set. As active state increases, so does data transmission and validation overhead, therefore this sort of storage imposes costs not only to validators but also to the larger network. The construction of a distribution technique called a "faucet," which is nothing more than a website that distributes free tokens or other cryptocurrencies to any connected wallet or IP address, is how we plan to cover these costs [29].

1.3 Objectives

1.3.1 General Objective

Generate native tokens and their distribution through a web faucet from the main Solana Blockchain.

1.3.2 Specific Objectives

1. Search documentation for the development of an application compatible with a specific Blockchain.
2. Understand of the development environment within a specific blockchain.
3. Create a beta crypto asset (Native Token) for subsequent publication.
4. Unify the project with the web Faucet.
5. Make Transaction to real Wallets from the Web Faucet.
6. Publish the final code as open-source.

Chapter 2

Theoretical Framework

Blockchain is an implementation of a fault-tolerant replicated state machine. Current publicly available Blockchains do not rely on time or make incorrect assumptions about the ability of the participants to keep time [30]. Each node in the network typically relies on its local clock with no knowledge of the clocks of other network participants. Each network node typically relies on its local clock, with no knowledge of the clocks of other network participants. Because there is no trusted source of time, there is no guarantee when the message timestamp can be used to accept or reject that message [31] every other network participant will make the same decision. For that reason, different methods of proof were created; for example, the PoH presented here is intended to generate a ledger with verifiable time passage. In other words, duration between events and message ordering. It is expected that every node in the network will be able to rely on the recorded passage of time without requiring trust.

The Blockchain is a chain of interconnected blocks that contain data. This distinctive feature and novelty lie in the fact that technology works in a decentralized way. The database is distributed in different blocks, not concentrated on one unique server, and nobody can control it in its totality: neither the State, the bank, or any other body [32]. All network users have equal rights. One of the main advantages of this technology lies in the fact that users interact with each other directly, without the participation of third parties, which facilitates transactions of different kinds. In addition, all the information is freely accessible to all users who can read each block without knowing to whom said block belongs. In other words, in Blockchain, anyone can see that someone has a million [33].

Nevertheless, this anyone will not be able to know who has explicitly it until the owner of the million lends him a particular key, confirming that it is precisely he who owns that million. However, such transparency immediately raises the need to protect data. In what way the technology guarantees the protection of user data? The security and protection of the Blockchain are guaranteed by special cryptographic keys that help verify the authenticity and certainty of the data in question. It is about a many-digit number formed with a special algorithm. The key changes completely even when the amendment to be introduced in the information within of the block is minimal. One of the most important components of the Blockchain is Cryptocurrency, particularly Bitcoin [34].

Nowadays in the world, there is a wide variety of digital currencies, but the most known of them is still Bitcoin. On January 9, 2009, Satoshi Nakamoto, then unknown creator of the first Cryptocurrency, reported on SourceForge the following: "I present Bitcoin, the first electronic payment system that uses a peer-to-peer network to prevent expenses secondary. The system is completely decentralized, lacking a central server or authority" [35] . The peculiarity of the Cryptocurrency is that: first, the payment made is irreversible (that is, once remitted, the money can no longer be returned); second, all operations are anonymous; Third, each user can use the Cryptocurrency without the need to have the permission of no competent entity. Furthermore, no matter where in the world the person is: territorial borders absolutely do not count in such a case. However, in addition to Bitcoin, different platforms work based on the Blockchain. For example, Ethereum. Blockchain correlates to Bitcoin like the Internet to e-mail. It is a large electronic system at the top of which It is possible to create various applications.

Open software based on Blockchain technology allows users to create and develop new applications. There is also its own currency, the Ether, which is used to pay servers on the Ethereum network. The distinctive feature of this platform is that it has a smart contract system. It is a self-governing program that activates automatically as long as specific contract requirements are met. "A contract intelligent" is the possibility that people all over the world planet conduct business with each other, even if they live in different places or speak different languages or use different currencies. It is always programmed in advance and works without the interference of all censorship or third parties. Their operation is carried out regardless of whether a computer is on or stays off.

2.1 Blockchain Origins

The first thing you think of when you hear the word Blockchain is bitcoin or crypto assets. Indeed, Blockchain is associated with bitcoin, as it is the technology that underlies Cryptoactive transactions. Blockchain is a decentralized registry technology that operates through a chain of blocks and can serve multiple purposes, such as payment systems, accounting entries, and for the case at hand, the development of smart contracts, among others. Hence, in certain cases, it is preferred to use the concept of "distributed ledger technologies" or "DLT", for its acronym in English (distributed ledger technologies). In this context, the Blockchain or DLT is a decentralized registration system that peers validate through cryptographic procedures and that provides a chronological and permanent record, publicly visible, of all transactions.

The origin of the Blockchain is linked to the birth of Bitcoin. The Bitcoin protocol was created in 2008 by an unknown person named Satoshi Nakamoto. Nakamoto was part of a cipher-punks movement, which sought to replace the central authorities and organize systems where personal information was protected through anonymity. Nakamoto reacted to government rescue of too big to fail entities in the financial crisis of 2008. Central governments were using the resources of individuals to save financial entities in crisis within an unstable financial system. Confidence in banks, in general in financial entities, and in central banks was undermined[36]. Nakamoto proposes a system that does not depend on any central authority in response to the lack of regulation and supervision of the Government. Bitcoin born as a currency free from the authority of governments in order to break the monopoly of monetary sovereignty of states. On October 31, 2008, Nakamoto published a white paper called "Bitcoin A Peer-to-Peer Electronic Cash System"[37]. Bitcoin was intended to operate based on a technology called Blockchain. In effect, the Blockchain serves as a record book for bitcoin transactions. The Blockchain was born to prevent the duplication of expenses (double spending) of crypto assets in a system without a central authority or intermediary controlling the issuance and transfers of said assets. The Blockchain makes it possible to solve this problem without the need to go to a central authorityor intermediary by making all the transactions visible to all interested parties. Indeed, the public registry of transactions makes it possible to dispense with authority or

intermediary within a payment system.[36]

2.2 Properties and Characteristics of the Blockchain

This technology is giving amazing capabilities to different sectors, now is introduced to economy, architecture, manufacture, science and different disciplines include Arts. Blockchain technology presents a series of characteristics that allow information to be managed in a secure and traceable way, maximum transparency and privacy for the participants. These features are very interesting in the business environment and allow companies to improve some of the existing business processes, in addition to defining new business models based on collaborative environments supported by Blockchain technology[38] [39]. The following subsection present the main characteristics associated with Blockchain technology:

- Secure
- Traceable
- Privacy
- Confidence
- Transparency
- Circumvention of Censorship
- Transparency
- Registration And Control Of Assets
- Vote

2.2.1 Secure

Cryptography is an essential element in Blockchain technology. Thus providing security on the information that is stored in the Blockchain and the information shared between the nodes of the network. Every Blockchain uses a set of asymmetric keys, and not all Blockchain use the same asymmetric key format. In Blockchain, all transactions are signed

by the private key of the sender. Within the transaction, the public key is included, which allows for verifying the content of the transaction and detecting if the transaction has been manipulated. Hash functions are another of the ones that ensure the security of the Blockchain[40]. Hash functions allow the generation of unique identifiers of the content of the blocks.

These block identifiers are used to connect blocks, offering a mechanism to identify alterations in the chain. Those depend on the specific Blockchain, but the purpose is to prevent alternations. The blocks, data, and transactions are validated by the entire network of nodes by the validator[40], providing security on the information incorporated into the Blockchain. Security lies in the ability of nodes to quickly detect data changes, rejecting the transaction or the block.

2.2.2 Traceable

Blockchain allows the movement through the chain of blocks and trace all the operations that have been carried out on a certain address or go back in time and review all transactions that were made on a certain date. It required exploring all the blocks generated on a specific date.

Query operations, we have to keep in mind that those queries are not stored, so they are not auditable by querying the Blockchain [41]. Each node responds to the information query that is sent to it. Since this query is not registered in the chain, it is impossible to know all the queries made in all the nodes. In Blockchain all consolidated transactions are stored on Blockchain.

This chain is growing in size and is fully stored by a large number of nodes that make up the Blockchain network. This Blockchain feature makes all the processed information traceable[41], being able to consult all the operations carried out using a Blockchain explorer.

2.2.3 Privacy

In public Blockchains, those addresses are not linked to the identities of the people who control each of the Blockchain addresses. To be able to operate in a public Blockchain, it is

necessary to have a pair of public keys and private keys that allow access to the Blockchain address.

Mathematics is the key to the success of any Blockchain. Based on mathematical operations, the generation of the set of keys and the address of a Blockchain is simple but expensive in the process. The process that allows you to generate the keys and the address does not require any personal data.

This Blockchain privacy is not available in all distributions since, in some cases, in order to operate on a Blockchain network, prior identification is required. Privacy is one of the main characteristics of Blockchain and another of the reasons for the success of the Blockchain in its beginnings since, in some cases, this privacy was used by some people to carry out operations considered illegal. This is one of the reasons that made some countries reject the use of Blockchain as a means of payment.[42]

2.2.4 Confidence

Confidence, does not presuppose an acknowledgment of risk, but rather an attitude of assurance. As opposed to trust, confidence does not operate under a condition of uncertainty. When used to describe a relationship with other people, institutions or systems, a state of confidence involves a sense of predictability, which significantly contributes to reducing the feeling of risk and uncertainty that would otherwise be felt in entering into such a relationship [43]. The mechanism to execute any operation in the Blockchain is public you can have the code, this is open-source, and is available for everyone. This characteristic is complemented with transparency.

2.2.5 Transparency

Transparency in the Blockchain is achieved by publishing the rules with which the operation of the Blockchain is defined. This is achieved by making the software code needed to run the Blockchain public and by creating a community of nodes and developers that follow this principle of transparency. In Latin America, transparency can benefit both parties: the citizens who control the Government and the Government that inspects citizens. In the Blockchain, the decentralized database is invariable and equally accessible to each

individual who enters the net. This means that any system transferred or created based on the Blockchain and its transactions can be displayed. If government transactions are registered in one of the Blockchains, it will be guaranteed that no one will appropriate illicitly from state money.

2.3 Possibles applications

Beyond bitcoin and cryptocurrencies that use Blockchain, it is essential to name a new way Blockchain can be used. The utility always comes hand in hand with the need, but the analysis must be carried out since there are projects that do not need the integration of Blockchain and only use it to acquire financing or relevance.

2.3.1 Circumvention of Censorship

Cryptocurrency has become globally recognizable when Bitcoin became famous as a means of payment for anonymous transactions and began to be used to circumvent censorship and political sanctions. In Latin America, inefficient economic policies and political instability sometimes demanded solutions and alternatives. One example of this is the market. Currency of Argentina, where consumers are forced to participate in the process of change that is little efficient and often does not allow them to trade at bargain prices. Market. Multiple cases of illicit exchange occur when individuals exchange bills of a value greater than 100 dollars since higher value bills tend to cost more than value smaller (because while the number of banknotes in cash, it is easier to hide the flow of illicit money). In Venezuela, such a practice is very widespread: for example, if a businessman is paid a sum in bolivars (local fiat), this immediately buys Bitcoin from the miner (those who create Cryptocurrencies) at a discount and sends the money abroad. Bitcoin and Ethereum have won popularity in Venezuela due to the low cost of electricity[44].

2.3.2 Registration And Control Of Assets

One of the advantages derived from the character of decentralized and unchanging Blockchain technology consists in that the data is not subject to changes in the political regime in the country. This means that changes in Government, no matter how radical they may

be, cannot lead to the erasure or change of records. In a region where there is political instability, citizens find it very beneficial to own assets registered in the Blockchain since this guarantees that they will lose their assets as a result of political convulsions. For example, uncertainty regarding land ownership is a problem throughout the world, but in Latin America, that problem is even more acute since this region thrives on corruption, poor property registration, and parcels are frequently confiscated. In most cases, this is due to political instability[45]. all these facts make it difficult to track scams and the illegal sale of Earth. This problem not only increases the risk of investing capital in real estate, which significantly worsens the investment climate and creates a bad institutional environment but also hinders access to financial services, such as property loans.

2.3.3 Vote

Holding fair elections is a problem that has been faced throughout Latin America for many years. Exist numerous factors that make it impossible to carry out honest elections in this region: fraudulent bulletins, poor counting of votes, falsifications, difficulties, and obstacles that do not allow citizens to vote safely and in comfort. The voting system based on Blockchain technology may be one of the most revolutionary ideas in the world of new technologies. In times of crisis, recession, and when governments and banks abruptly stop their activities, crypto can be vitally important[46]. In countries like Venezuela, where hyperinflation makes it practically impossible to buy foodstuffs and basic necessities, citizens decided to resort to the use of Bitcoin, although until At the moment, this has not helped them get out of the financial crisis. We would like to highlight again that more than 45 of the Latin American population is not integrated into the system financially. It is about 207 million people who do not have access to bank accounts or financial institutions. This leads to unemployment, illicit activities, and overall marginalization. It is necessary to take into account the fact that 55 percent of people lack the possibility of joining the financial system if they have access to the Internet. That means you can increase the accessibility of financial services via the Internet[46]. Blockchain and financial technologies are tools capable of helping unleash the potential of this rich continent but are insufficiently developed.

Those characteristics makes more reliable an infinite of possibilities of application and

uses every transaction. we can use not only on value of the

2.4 Current Blockchain Applications

No one knows how many Latin Americans do not have a bank account. However, according to preliminary data, they reach 50, that is, more than 300 million people, which constitutes a huge market for taking advantage of new possibilities. As we have already said, many Blockchain companies have entered the Latin American market to offer services to the population lacking access to banks. These projects help find new pathways to global markets. For example, Argentina is one of the countries with the highest number of Blockchain enthusiasts per capita in the world. It can be said that the real number of Bitcoin users in Argentina is not considerable, but this country is one of the few in the world where ordinary people regularly use Bitcoin coins in real transactions. The employment of Bitcoin in countries like Argentina allows users to circumvent stringent government restrictions to receive money from abroad. The development of the Blockchain industry (unlike the introduction of its Blockchain technology) in America Latina is moving fast. This occurs not so much because there is interest on the part of the population but because there is a need to have alternatives to traditional banking systems, especially in countries like Venezuela, Argentina, Brazil, and Mexico [47]. To better understand the development of technology, Blockchain and Bitcoin in Latin America seem to us convenient to examine some of the best companies in emerging markets, stock markets, and companies in this region:

1. Gravel Ripio, formerly known as Bitpagos [48], is a financial payment company that provides decisions on the matter of electronic payment to companies in Latin America. Ripio offers a series of financial services with the use of Blockchain technology to developing markets. The company helps sellers process transactions internationally by means of credit cards or currencies of Bitcoin, charging a small part of the cost. The users can also instantly buy the coins of Bitcoin by paying in cash in Argentina, Brazil, Chile, and Ecuador. Recently the Ripio company has collected the series A for an amount of US2.25 million in order to continue financing its international expansion. The server monthly processes several million dollars.

2. Bitso is the first Mexican exchange of the currencies of Bitcoin[49]. For the majority of Mexicans who do not enjoy financial services is a platform to change the Bitcoin coins for Mexican pesos. Today, thanks to volume of its financing, which reaches the amount of US 4.35 million, Bitso allows the use of Bitcoin to reduce expenses and speed up transactions. The company started its activities at the beginning of 2014 and states that the total of its monthly operations has grown more than 40 compared to to the year 2014.

3. BitInka/Inkapay Founded in 2013, BitInka [50] is a platform that facilitates the trading of Bitcoin coins throughout Latin America. BitInka currently operates in Argentina, Bolivia, Brazil, Chile, Colombia, Spain, Peru, Venezuela, and the US, offering the trading of Bitcoin coins with national currency. The company plans to expand its popular payment application to of Bitcoin Inkapay by introducing integration with credit cards credit, as well as the new API interfaces intended for the integration with electronic commerce. Inkapay too intends to launch a mobile application in September.

4. Volabit is an exchange of Bitcoin coins located in Mexico. It is generally used to make money transfers and conduct global trade by the people who own, so there is least basic knowledge of the technology Blockchain. Volabit accounts can be completed by through an online bank draft or in cash at any 7-Eleven, Farmacias Benavides, Farmacias del Ahorro o Extra, in Mexico [51].

2.5 How works a Blockchain?

A bitcoin unit is nothing more than a number, but only certain combinations of numbers are valid bitcoins. These are solutions to a well-defined equation, and whoever discovers a new one owns it (this process is widely called mining). Once discovered, a bitcoin can be traded, with transactions recorded in a ledger. To avoid nonrepudiation, transactions are digitally signed with the credentials of the seller. There is no centralized ledger because users would not trust one, and there are too many transactions to store all of them in one place [52]. Hence Cryptocurrencies provide a distributed ledger where every computer

involved in the transaction (coin or fraction of a coin) keeps a copy of that transaction history. Blockchain technology ensures that no party storing this history can tamper with it without being detected. Hash Functions are extremely important because they are the data units containing the transaction details and a timestamp. Both can be represented as numbers or strings in a computer. A Blockchain is analogous to a three-column table, with each row representing a particular transaction, the first column storing the timestamp transaction, the second column storing the detail transaction, and the third column storing a hash of the current transaction plus its predecessors.

When a new record is added to a Blockchain, the most recently computed hash is broadcast to all interested parties. Every party does not need to keep a copy of the entire transaction history; a few parties are sufficient. Because everyone is aware of the most recent hash, anyone can verify that the data has not been altered, which would be impossible without obtaining a different and thus invalid hash. The only way to change the data while keeping the hash is to find a collision in the data, which is computationally impossible. It would necessitate so much computing power that it would be practically not viable.

A hash is an encrypted version of the original string that cannot be deduced from it. One method of computing the hash of a string is to encrypt it and then scramble and xor the output bits. A hash is produced mathematically by a hash function, f, which must have two important properties: the size of the input and output spaces must be large; it must be minimal impossible to match collisions, that is, two inputs x1 and x2 that produce the same output $f(x1)$ $f(x2)$. When you register on a website, you do not want the site to store your password p in its database, so you use hash functions because anyone with access to the database could read it. The website should save the password hash. When you log in, the hashed password p is compared to the stored value. For practical purposes, the probability of an incorrect password producing the same hash value y as the actual password is zero [52]. Secure Hash Algorithms (can be SHA1, SHA128, SHA512, and so on) are examples of hash functions that are implemented in the standard Python module hashlib. They can accept any string as input and always return an output string that is a hexadecimal representation of the output function number with a fixed number of digits.:

```
1 >>> print (hashlib.sha1(  hello   world ).hexdigest())
```

```
#print of the transformation of the string "hello world" in hex
>>> 2aae6c35c94fcfb415dbe95f408b9ce91ee846ed #output
```

A simple implementation of a Blockchain in Python can be helpful. As a first step, we can define a function that we call bhash that, given the timestamp and details (a string or other serializable object) of a new transaction along with the hash of the the previous transaction computes a different hash using the SHA1 algorithm:

```
import hashlib, json, time
def bhash (timestamp, details, prev_hash):
 token = json.dumps([timestamp, details, prev_hash])
 return hashlib.sha1(details).hexdigest()
```

We use the JSON serializer to combine the elements into a hashable string, which we then pass to the hash SHA1 hash function. Our decision to serialize in JSON is an implementation detail, not the only way to accomplish the goal.

To mimic the next step, we can create a Blockchain class to encapsulate a list of blocks:

```
class Blockchain(object):
 def __init__(self, details= new - c h a i n ):
 self.blocks = [(time.time(), details,    )]
 def record(self, details, timestamp = None):
 timestamp = timestamp or time.time()
 prev_hash = self.blocks[21] [2]
 new_hash = bhash(timestamp, details, prev_hash)
 self.blocks.append((timestamp, details, new_hash))
```

The constructor of the class, “init”, creates a list of blocks and stores the first block in the list. This first block contains a timestamp and some details but no hash. In the case of bitcoin, this would keep track of the discovery of a new unit and its owner. The class also includes a second method, “record,” which stores them in a new block when given the details of a new transaction and an optional timestamp (otherwise computed automatically).

```
>>> bc = Blockchain( A  found  $ 1  )
>>> bc.record( A  gives $1 to  B  )
>>> bc.record( B  gives $1 to  C  )
>>> bc.record( C  gives $1 to  D  )
```

Now is available to print the blocks in the Blockchain:

```
>>> print bc.blocks
 [(1495941516.704196,  A  found  $ 1  ,    ),
 (1495941516.704201,  A  gives $1 to  B  ,   a75a9227f  ...   ),
 (1495941516.704277,  B  gives $1 to  C  ,   ca911be27  ...   ),
 (1495941516.704290,  C  gived $1 to  D  ,   cb462885e  ...   )]
 #output
```

'cb462885e...' is the final hash. To make this technology work, we must broadcast the last hash and have a few copies of the entire chain stored by different parties. The parties in this context are the peer-to-peer network computing nodes in charge of recording and storing transactions. This is a network issue that is beyond the scope of this article.

It also is fundamental that every party can verify the chain integrity. This is executed by using the next function:

```
def verify(Blockchain):
 prev 5 Blockchain.blocks[0]
 for block in Blockchain.blocks[1:]:
 new_hash = bhash(block[0], block[1], prev[2])
 if block[2] != new_hash: return False
 prev = block
 return True
```

The last code looped through all of the blocks beginning with the second recomputing each hash, then compared it to the one stored in block[2], the third column. If the code comes across a hash that does not match, it returns False; otherwise, it returns True. On our Blockchain, we can refer to this code as:

```
>>> print verify(bc)  #output result as True
```

From a technology perspective, there is a lot more than this network. There are different algorithms for data distribution, syncing nodes, efficient storage and querying, debugging conflict resolutions, and so on.

2.5.1 Operation of distributed ledger systems

The Blockchain comprises a group of interconnected blocks (hence its name "Blockchain"). Each block contains a list of past transactions, that is, transfers of tokens from one person to another. Likewise, the blocks are identified with a kind of fingerprint called a hash,

together with a timestamp and the identification (or hash) of the previous block. Thus, "While a book relies on page numbers to order its internal content (which makes it possible for anyone to put together a book in the proper order), the Bitcoin block chain depends on the data stored in the header of each block to organize the shared database, which includes a hash of the previous block and a timestamp, creating a sequentially organized chain" [53].

Create a new block, it is necessary for each node or participant, to offer a verification called "proof-of-work", which guarantees the integrity and security of the system. Indeed, in order to create a new block, a hash must be generated through a complex mathematical procedure to solve a puzzle. This operation is known as "mining". The protocol adjusts the level of difficulty of the puzzle depending on the number of participating nodes (or miners): the greater the number of participants, the greater the difficulty in creating a hash for a block. When a miner has found a valid hash, it communicates this situation to the Blockchain network so that the other nodes verify that the hash meets the requirements of the protocol. Finally, the miner receives remuneration as an economic incentive to keep the scheme running, called a "block reward". Currently, groups of miners (mining pools) have been created to combine efforts and resources and distribute the rewards.

It is, therefore, a system that operates as an unalterable, replicated, and accessible database. It is an unalterable record, as long as it is guaranteed by means of cryptography, which allows the generation of data chains -or blocks-, by virtue of which the successive blocks that are formed require incorporating the previous ones, identified with their corresponding hash. In this sense, if a person tries to modify a transaction registered in a block, they will break the chain since when modifying the block content, the protocol will immediately create a new hash to identify the new modified block, which will be impossible, to link to the chain of subsequent blocks.

The more transactions that take place and are recorded on the Blockchain, the more difficult it is to modify a block retroactively. Thus, to modify a block without affecting or breaking the chain of blocks, the interested party would have to generate a new hash for each subsequent block in the chain. Likewise, and taking into account that it is a registry replicated in all the participants since it operates through a standard communications protocol based on decentralized applications, the person interested in modifying a block

must modify the hash of most nodes to validate the change. In short, it is extremely difficult, and therefore it will become extremely rare, for a person to be able to modify or delete the information that is registered in the Blockchain: its technical design favors the status quo and makes the information resistant to change[54].

Confirmation or validation of the information is established through a distributed consensus, that is, the confirmation is carried out when the majority of the nodes verify that a given block has complied with the so-called proof-of-work. In this sense, the Blockchain replaces the need to trust third-party intermediaries or central authorities, with trust in technology. Including a smart contract in the Blockchain would eliminate the need to go to a third party to ensure its inalterability and execution, unlike models that operate under the client-server model (client-server model), such as eBay, Uber, and Spotify. In this order of ideas, it is a registry accessible to all participants, where there is no central authority or an intermediary to confirm the information, which allows them to interact directly with each other.

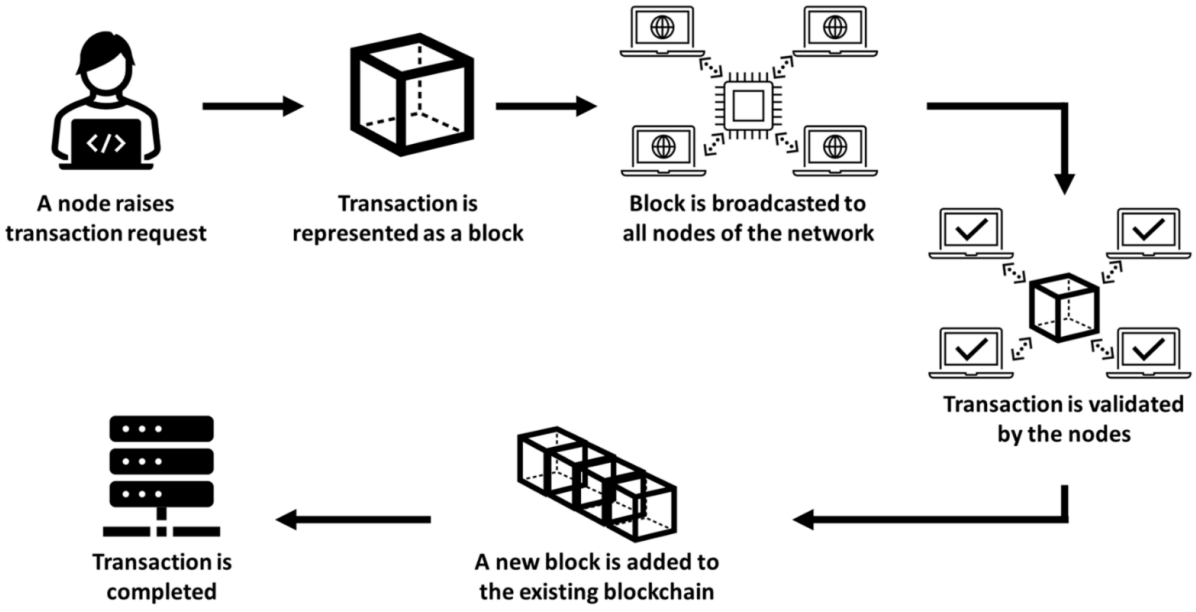

Figure 2.1: Operation of distributed ledger systems

The absence of a central authority also implies that it is a transnational scheme in which the participants can access disintermediated global services. In this context, we are faced with a distributed system because there is no central server or cloud where a central authority stores the information.

2.5.2 Blockchain allowed and not allowed

There are two types of Blockchain: permissionless Blockchain and permissioned Blockchain. The first, the non-permitted Blockchain, refers to the one that is open and accessible to all people, such as Bitcoin and Ethereum. In this type of Blockchain, anyone with Internet access can download the corresponding software (open source software) and become part of the network, even without revealing their personal information or identity and without requesting permission to do so. It is the purest expression of a Blockchain: decentralized, pseudonymous and accessible to anyone. However, it is the one that generates the most concern, since it facilitates the commission of money laundering, terrorist financing or, in general, illicit activities.

On the one hand, since it is a system accessible by anyone and that it is replicated in each of the nodes or participants, we are faced with a database with information that can be used for advertising, big data studies, dates, and even crimes. Thus, certain transactions that the parties do not want to make public cannot be carried out through a Blockchain system. The operations that are executed by smart contracts will be registered without it is possible to eliminate said registration, and given the pseudo-anonymity of the parties, it will be possible to identify the transactions with their owners. Hence we are faced with the negative side of transparency. In this order of ideas, regulators will have to find a way to integrate the right to be forgotten or removed with the immutability of the Blockchain [55].

In response to the inconveniences or concerns that the pseudo-anonymity of the Blockchain may cause, a new type of decentralized registries has been emerging: permitted Blockchains. These are new schemes where the characteristics of the Blockchain are nuanced to mitigate the risks associated with its operation and improve its efficiency. Indeed, in this type of Blockchain, access is limited. Not just anyone can be part of the network but, on the contrary, it is necessary for a central authority to authorize the parties that will participate in it. Thus, the central authority will have knowledge of the identity and characteristics of the people who interact within the network [55]. However, inside the network, the parties may preserve their pseudonymity when interacting with each other. Well, these access restrictions allow easy tracking of those responsible for any illicit activity carried out through

the network.

Furthermore, it should be noted that permitted Blockchains operate faster and more agile than disallowed Blockchains. Since the non-permitted Blockchains are open and accessible to anyone who is interested in being part of the network, the number of participants or nodes, necessary to reach a consensus to validate the transactions will be greater. Hence the process validation takes longer; while, on the contrary, in permitted Blockchain, the validation of transactions is faster, while the number of people needed to reach a consensus will be considerably less.

Although these are two different types of Blockchain, nothing prevents them from working in an articulated and complementary manner. It is reasonable to consider that non-permitted Blockchains can be used as the basis on which different types of permissible Blockchains operate. Thus, the not allowed could be used in a general way; while the permitted could be used to operate on specific transactions that require or justify a higher level of security [55].

2.6 Smart Contracts

In 1996, Nick Szabo, a lawyer, and computer scientist, first introduced the concept of a smart contract [56]. With robust cryptographic protocols, Szabo recognized the possibility of writing computer software that resembled contractual clauses binding on the parties and reducing their chances of non-compliance [56]. In this sense, he argued that the digital revolution offers the possibility of creating new institutions through smart contracts, in the sense that they are more functional than those incorporated on paper. Likewise, he recognized that the term did not involve the use of artificial intelligence but rather the use of computational algorithms that would eventually be used in all types of contracts. Although a novel idea was carried out in the 1990s, the necessary technology for its adequate development was not available. It was only in 2008 that the development of Blockchain technology offered the necessary platform and ecosystem for smart contracts. Today we are facing, perhaps, the greatest technological revolution since the appearance of the Internet, a model conceived in 2008 that has the potential to transform the way we live and interact with others in every one of the social environments. In this context, the

definition of the term "intelligent contract" is of the utmost importance for the object of study of this document since its grammatical reading can lead to an error. Indeed, the first thing that comes to mind is that these contracts involve using artificial intelligence or intelligent software. However, as seen below, they are not smart contracts. Thus, some define smart contracts as "systems that automatically move digital assets according to pre-specified arbitrary rules" [57]. Others define them as "agreements whose execution is automatic" and are "usually carried out through a computer code put into operation that has converted the legal prose into an executable program" [58]. For his part, Nick Szabo has spoken on the matter on several occasions, as follows: in 1996, he said that "an intelligent contract is a set of promises specified in digital form, including the protocols within which the parties fulfill with these promises" [59]; and in 1997 he said that "Smart contracts combine protocols with user interfaces to formalize and secure relationships across computer networks. The goals and principles for the design of these systems are derived from legal principles, secure protocols, and theories of reliable and economic theory" [60]. This concept reflects the importance of digital revolutions, assuming that technology can remedy non-digital system shortcomings.By the preceding, it is reasonable to conclude that, for this article, a smart contract is software that allows the automatic execution of codes that incorporate obligations between previously agreed parties and that are stored in a decentralized registry, upon verification of the codified conditions. The first comment that the concept deserves is that it is not intelligent because it does not think for itself, as a lawyer is supposed to do. Perhaps one day technology will allow the existence of contracts that think for themselves through artificial intelligence. While that happens, smart contracts are limited to being mechanisms for executing computer codes. Second, the use of the term "contract" can lead to confusion: it is not a contract in that it is not a source of obligations, but rather a mechanism for executing contractual obligations. Indeed, the term "contract" is used loosely and informally, leading to views that smart contracts will eliminate the need for lawyers and judges by guaranteeing and automating the execution of contracts. Decentralized ledger technologies (DLT) and Blockchain Smart contracts are planned to operate in conjunction with Blockchain technology, precisely to execute the transactions that are in the decentralized registry; hence, it is essential to carry out a brief analysis of these records.

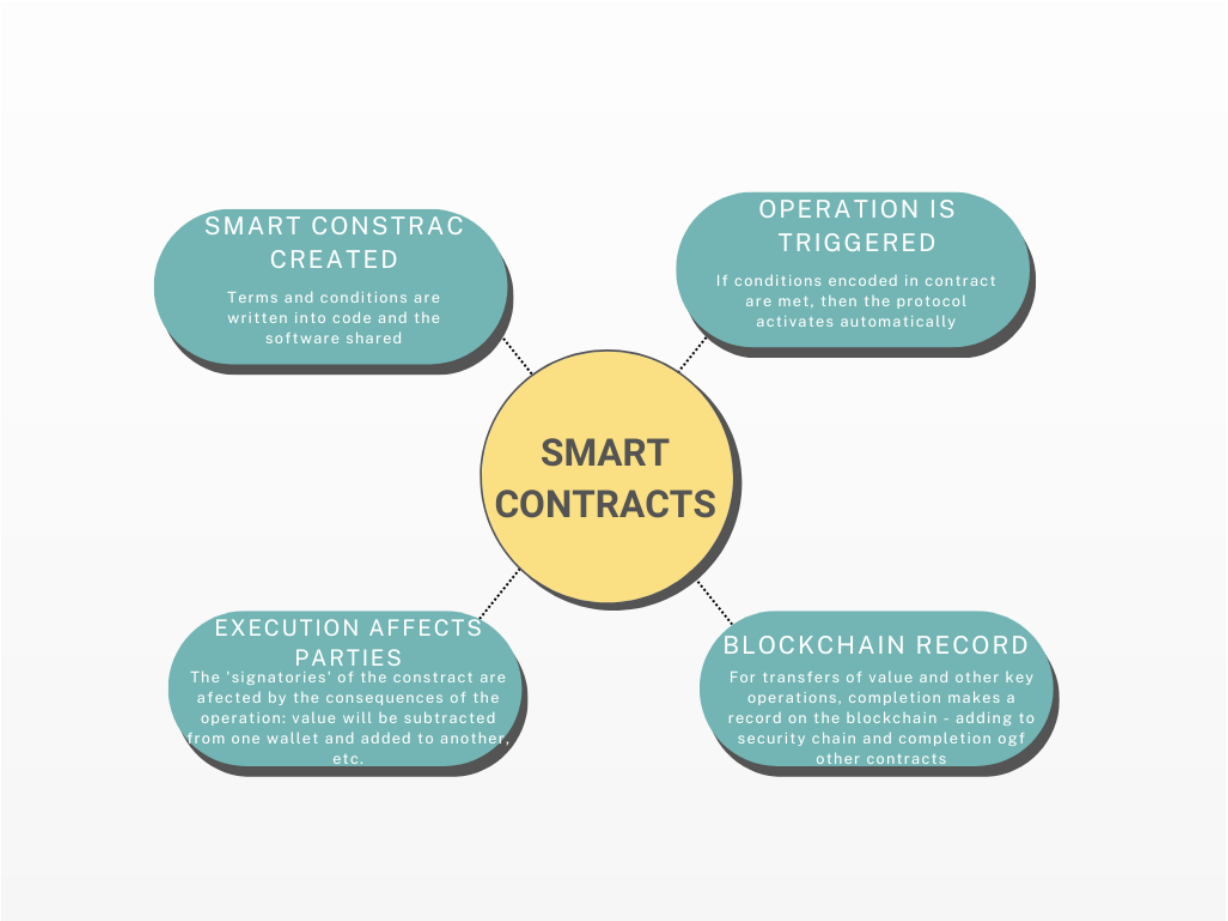

Figure 2.2: Smart Contract - How They Work.

As a good example, You could also store a promise to do something later, with the promise stored as code that would execute the promise automatically. This is a smart contract; for example,“Maria promises to pay Bryan 1 if it rains in Chicago on January 1, 2028.“ If the promise is in the Blockchain and an API can check whether the conditions are met, the system can execute the transaction automatically if the condition is met. The bitcoin network was the first, but new ones are constantly emerging to trade and specifically handle smart contracts, which are defined as “applications that run exactly as programmed without any possibility of downtime, censorship, fraud, or third-party interference” [52] On the one hand, trading Cryptocurrencies may seem like stamp collecting, but the underlying technology has only recently begun to revolutionize contracts and human interactions. Many white collar jobs will be displaced in the same way that robots have displaced blue collar jobs. It will also generate new jobs we cannot even imagine.

2.6.1 Automatic execution of smart contracts

One of the main elements of smart contracts is their self-enforcement capability. Smart contracts are self-executing in that they automatically execute a transaction upon the occurrence of previously defined events. Said characteristic is intended to prevent man from intervening in the contractual execution, presumed to be partial and unreliable, by introducing an algorithm or code that cannot change his mind and refuses to comply with his obligations [61]. The code is impartial and objective, which guarantees that the contract will be executed to the letter, without alterations or modifications to its content or unforeseen circumstances. In this order of ideas, it is reasonable to consider that the need to go to the judicial system to request the fulfillment of obligations is highly reduced if it is not eliminated.

Although immutability may be initially attractive, it is an attribute that can present several problems. In this context, smart contracts cannot be modified or stopped, or, what amounts to the same thing, they are immutable. Once the code is entered into the decentralized registry, its protocol is unstoppable no matter what changes may occur in reality. problems.

2.6.2 Challenges of self-execution

The first thing that must be pointed out is the problems associated with the automatic execution of contracts; if the fulfillment of the contract depends on a computer code, it must be guaranteed that said code does not contain errors. Although the immutability of the smart contract dispenses with human intervention in its execution, it does not eliminate the possibility that the code contains programming errors (bugs) [62]. Thus, the contract may be executed incorrectly due to an error in its programming, which generates inconveniences for the contractual parties, especially when it is impossible to roll back the operation or modify it, even if it has identified the error before its execution. In this order of ideas, it is appropriate that the parties agree on the contract and who will assume said risk. In conclusion, even if smart contracts imply the self-execution of contractual obligations, they do not guarantee a perfect fulfillment of the obligations derived from the contract.

Second, the code needs to reflect the will of the parties adequately. Whoever codifies the contractual conditions and translates them from a natural language to a code language may generate results that are far from the true intention of the contracting parties. On the one hand, it must be remembered that smart contracts are translated into computer codes that are not created by lawyers, but by systems engineers who probably do not have legal training, which is why they can interpret and translate inadequately. contractual provisions. On the other hand, it is normal that the parties or their lawyers do not have the necessary and sufficient knowledge to verify that the codification fully reflects their will.

Thirdly, an immutable contract requires that all possible events that could affect its development be foreseen and incorporated into it [62], since once entered into the Blockchain, it will not be possible for the human being to intervene or modify its content. Well, in traditional contracts (as opposed to smart contracts), it is normal for the parties to modify their clauses so that they can be adapted to external circumstances, such as changes in regulation or economics. It is even normal for the parties to tolerate non-essential breaches without the need to modify the contract. In smart contracts, the parties do not have such possibilities, which is why it has been considered that the main virtue of these contract execution mechanisms can become, at the same time, one of their greatest drawbacks: "it can then be argued that smart contracts are rigid and can easily be disconnected from the transactional reality on which they operate because it is not technically possible to make adjustments" [63]. Such rigidity deprives the contractual parties of the possibility of deciding whether or not they wish to fulfill their contractual obligations. Indeed, it is possible that the fulfillment of the obligations derived from a contract is much more onerous than its breach, and the parties decide to breach the contract (deliberate non-performance) and assume the consequences. The efficient breach theory establishes that for one of the parties to a contract it may be more profitable to breach its obligations than to comply, even if such conduct becomes morally reprehensible [63]. This possibility is prohibited in the case of smart contracts. In short, smart contracts sacrifice the flexibility and dynamism of contracts, for immutability and self-execution.

Fourth, the inflexibility of smart contracts ignores the reality of commercial relations, where often the fulfillment of the obligations of one of the parties is analyzed under reason-

ableness or good faith. These are situations where it may be more efficient for the parties to assess compliance with obligations ex-post and not ex-ante. This parameter offers flexibility to the parties to adapt the fulfillment of the obligations to the vicissitudes of the markets and to the transactional contexts, a circumstance that, due to its conditional structure, is not allowed in smart contracts. Well, the semantic flexibility that smart contracts lack allows the interested parties to contract in volatile scenarios or situations and avoids unnecessary negotiation on all the possible eventualities that may arise in developing the business relationship, which would raise transaction costs. In short, the inflexibility of the contracts.

2.6.3 Coding of Natural Language-Contractware

As has been mentioned, smart contracts are expressed through computer codes immersed in a Blockchain. In this sense, and since said technological platform has its own language, it is necessary that the language of the contract that is to be executed through an intelligent contract must be translated into code or programming language (contractware). Indeed, it is possible that the smart contract is linked to a pre-existing contract written in natural language or that, on the contrary, the contract that is intended to be executed has been conceived in code language.

2.6.4 From the real world to Translation to code language

Indeed, the smart contract formation iter assumes that the terms and conditions of the contract (which are generally agreed in natural language) are valid in accordance with the provisions of article 1502 of the Colombian Civil Code [64]; and that the execution of the services of the parties will be carried out by means of a computer code before the verification of the information that triggers it. The foregoing entails the need for the language that is translated into the code language to be conditional (if-then-rules) or to be in terms of false and true, since they must be able to be expressed in a way that a computer is capable of. to read them and to execute a command against the input of the information that activates it.

In this context, the reader will be able to sense from now on that it is impossible to

translate the entirety of a contractual document into a code language without considerably compromising its content, especially when the codifiers should not affect or influence the substantial aspects concerning the obligations that are intended to be executed via smart contracts. It must be reiterated that programmers have no legal training and therefore do not understand legal language, which otherwise has little or no tolerance for errors. Indeed, programmers tend to read contracts from the point of view of conditional statements or true and false, and leave aside the fact that legal language often makes sense from the context that surrounds it, which is why the translation from this to the code language can be complex. Well, in this type of linguistic structure the expressive limitation of signs is exacerbated, especially when, on the contrary, in traditional contracts we find ourselves faced with a need for space for ambiguity and for interpretation. Ambiguity is the anathema of the code language. In addition, the fact that the contract is executed through smart contracts does not eliminate the need to interpret the contractual provisions to fill the obligations of the parties with content, for which it is necessary to have the necessary legal knowledge, of which, it is reiterated , the programmers lack [65].

Lawyers and the parties can solve the above problem by performing an ex ante interpretation and setting the scope and interpretation that the person executing the smart contract must give to the terms of the contract. Indeed, it is possible that a term is subject to various interpretations, hence inconveniences may arise regarding the scope of the terms when translating the contractual language into code language [65].

2.6.5 Code Formality

An alternative to natural language coding is to write the smart contract directly in code language (direct coding) without the need to have a contract written in natural language from the start. This alternative could reduce the ambiguity of contracts since the code language is binary, and it would reduce translation errors, as the contract would be intelligent from its conception. Once again, we are faced with the need for lawyers to think in binary terms when drafting contracts and for programmers to understand the basic principles of contract law [66].

2.6.6 The Oracle Problem

As has already been mentioned, smart contracts are mechanisms for executing obligations that operate automatically upon verification of the fulfillment of a condition. For example, if a smart contract is designed to make payment against the delivery of merchandise, the system must know if the delivery was made. Once you are certain that the delivery was made, the smart contract will have to be executed and proceed to payment; hence, a source of information from the real world is required to transmit the necessary data to the Blockchain so that the smart contract can be executed. This makes it necessary that the event that catalyzes the smart contract can be verified based on available information and that said information can be transmitted to the Blockchain.

To verify compliance with a condition, the Blockchain must have contact with the real world and a source of information for this purpose, a source that has been called an oracle. Oracles are programs, companies or even natural persons that transmit information from the real world to the Blockchain so smart contracts can be executed. In this way, it is possible to distinguish between events that occur inside the Blockchain (on-chain events) and those that occur outside the Blockchain (off-chain events). Thus, for example, a Blockchain system may have information from Bloomberg, Augur, a central bank, another Blockchain, or even temperature sensors (internet of things), to trigger the execution of smart contracts. It only remains mentioned at this point that it is important that the parties to the smart contract designate a trusted oracle to serve as the source of information and agree that they irrevocably accept the information provided by a said third party.

Indeed, a smart contract designed for the purchase of shares of a company listed on the stock exchange that has to be made effective when its listing price falls below a certain value requires obtaining said information from an oracle so that the operation can be carried out. out in real time. This example does not present any major inconvenience, as it is information that is public and easily accessible to the oracle. However, different situations may arise where there are different sources with divergent information, hence the importance of the parties agreeing in advance on the source of information they will use. It is also possible that there are situations in which there is no information about an event that occurs outside the Blockchain or it is difficult to access.

On the other hand, it is possible that the parties to a contract assign to an oracle the verification of conditions that escape the binary nature of smart contracts. Thus, the parties could agree so that a third party, outside the Blockchain and the smart contract, determines if the obligations of a smart contract were fulfilled in a reasonable manner, with the best efforts or in good faith, verification that results impossible to perform to the Blockchain. In this order of ideas, it is possible to make smart contracts more flexible, sacrificing their independence from the outside world, while human judgment would come into play again as a catalyst for the execution of smart contracts. Thus, for example, establishing whether a seller delivered goods in accordance requires an examination of the content of the goods to verify their conformity with the provisions of the contract, hence human judgment is necessary [67].

Finally, it is possible for the parties to assign conflict resolution mechanisms through third parties, recognizing an arbitrator as an oracle. One of the main challenges associated with Blockchain technology and smart contracts is how disputes arising from them will be resolved. Due to the technological component involved, it is advisable to submit such disputes to an arbitration tribunal, whose arbitrators must meet certain qualifications that allow them to have the necessary knowledge to best resolve the differences between the parties. Well, the designation and linking of said parties to the execution of smart contracts could be carried out through the figure of the oracle, as a third party that resolves a dispute and transmits said information to the Blockchain [67].

2.6.7 Validation aspects

To get more in touch and get rapidly idea of the ecosystem I will explain the main concepts but focuses in an specific block chain and constrat with the different main Blockchain. In this case we will use the Solana Blockchain. Solana is rapidly expanding ecosystem and its versatility have inevitably drawn comparisons to Ethereum, the leading Blockchain for decentralized applications (dApps). Both Solana and Ethereum have smart contract capabilities, which are crucial for running cutting-edge applications like decentralized finance (DeFi) and non-fungible tokens (NFTs). But there are some fundamental differences between the two. Any transaction or event on the Solana Blockchain generates a hash based on the SHA256 encryption algorithm. This algorithm takes an input and generates a

unique output that is very difficult to predict. What Solana does is use the output hash of a transaction and use it as input for the next hash. What this does is introduce the order of transactions in the following output.

What this hash mechanism does is generate a long and uninterrupted chain of transactions. This allows a clear and verifiable order of transactions to be generated, which is then added to a block by a validator. Through this mechanism it is no longer necessary to generate a timestamp as it happens in Bitcoin, Ethereum, Litecoin, etc.

Each hash in turn requires a certain amount of time to complete. This element also allows validators to verify, very quickly and easily, how much time has passed.

Proof-of-History (PoH) therefore differs greatly from the PoW consensus of Bitcoin or Ethereum. These two cryptocurrencies aggregate the transactions in blocks without the slightest order. What PoW miners do is add a timestamp, which is the time and date the block was generated, according to the clock of each Blockchain. The timestamp can vary by node and can even be false, forcing nodes to verify that the timestamp is valid.

By hashing transactions, validators have less information to process in each block. Using a hashed version of the last state of a transaction reduces confirmation times for new blocks.

Indicate that PoH is not exactly a consensus mechanism, but a solution that saves time and resources to confirm transactions. It really is a plugin that is added to the proof-of-stake consensus that simplifies the random selection of the next validator. It allows nodes to validate the order of transactions in shorter periods of time, making the network faster. Unlike Solana, Ethereum is a proof of work (PoW) Blockchain, wherein miners must compete to solve complex puzzles in order to validate transactions, making this technology more energy intensive and hence detrimental to the environment.

What is turbine Block propagation protocol that facilitates the distribution of information to nodes, helping to maintain consensus. This should be a quick process, as blocks in Solana spawn every half second or so. The block propagation process is required to be faster than the block generation.

What Turbine does is divide the problem. Specifically, what it does is divide the block information into small sections that are distributed on the network. These “chunks” are rebuilt by the nodes according to their own states.

The whole block is not really sent, only a portion of the information block and each node must reconstruct it. If the node does not have the information to "rebuild" the block, it can request it from the rest of the network. This process is done in parallel, thus reducing bandwidth consumption, maximizing speed and thus maintaining consensus.

What is Gulf Stream? It is a network transaction caching protocol. It is in charge of receiving the transaction and sending it to all the nodes, prioritizing the generating nodes. It allows all the nodes of the network to access the information necessary for the recreation of the blocks. Solana creates the blocks through a quorum-based election that has the power to generate a block and broadcast it to the network.

But the role of the generator nodes is not only in charge of creating the blocks, but also of being selectors of the next group of validator nodes. This allows knowing at all times which nodes will generate the next block. It allows the nodes to receive the transactions and route them to the following generators. A mechanism that allows to reduce the generation time of the next block.

System tampering is prevented by a transaction lifetime of just 24 seconds. When a transaction is not confirmed in this period of time, an output can be generated: a transaction failure and the need to resend the transaction. This situation of failure in the transaction validation time can only occur if the transaction capacity of the Solana network is exceeded.

What is Sea Level? Solana offers the ability to parallelize transaction validation and the ability to run smart contracts. The idea of this new Blockchain is to compete with Ethereum, especially in terms of DApps and DeFi support.

Solana smart contracts make use of the C language and Rust, to create a unique smart contract programming ecosystem. This offers a great ability to palellize the execution of smart contracts. Sealevel is the name Solana developers have given to these capabilities.

This feature enables the ability to read, execute, and write instructions in parallel within the Solana smart contract execution layer. A smart contract can execute multiple actions simultaneously, while in Ethereum and EOS only one action can be carried out at a time.

Sealevel what allows Solana is greater scalability than other Blockchain networks. It could come, with the integration of enough high-performance nodes to support up to

500,000 transactions per second. Additionally, the need for a second layer is eliminated to improve scalability.

What is Tower BFT Solona Protocol against Byzantine Fault Tolerance that is combined with PoH to help keep consensus secure and decentralize the network. It is an evolution of "Practical Byzantine Fault Tolerance" (PBFT), which is a well-recognized Byzantine fault tolerance protocol within distributed computing.

Tower BFT acts as a "judge" within the timestamp system running on the Solana network. A synchronized clock is used between all the nodes that serves as a point of control, verification and acceptance of the work done by the nodes. Thus, it makes it possible to create a decentralized consensus on the work and its acceptance by the network. As long as the work respects the consensus rules of the Solana network.

This PBFT-derived mechanism is really fast and has also been optimized by the Solana developers. Tower BFT and PoH are elements that allow Solana to have very low spawn times and maintain consensus.

2.7 What is a Wallet?

A cryptocurrency wallet is a program that keeps track of a group of keys and can be used to send, receive, and transfer digital currency. Wallets come in a variety of styles. A wallet can be a piece of paper, a directory or file in the file system of your computer, or a specific item called a hardware wallet. Wallet creation and management are also made simple by a number of computer programs and smartphone apps. A keypair consists of a securely created private key and the public key it derives from cryptographically [68]. A keypair is made up of a private key and the associated public key. One or more keypairs are collected in a wallet, which also offers a way to interact with them. The receiving address of the wallet, or simply its address, is referred to as the public key (pubkey). The wallet address can be freely shared and shown. The receiving address of a wallet is necessary when another party want to transfer it some cryptocurrency. The address can also be used to read some information about a wallet but cannot modify the wallet or withdraw any tokens, depending on how a blockchain is implemented. The private key is used in any digitally signed transactions to send cryptocurrencies to another address or to make any

changes to the wallet. Never share the secret key with anyone. Anyone having the private key to a wallet has the ability to withdraw all of the tokens it holds. Tokens delivered to the address of the wallet are permanently lost if the private key for that wallet is misplaced. As for keypair security, working with the keypair, and signing transactions to use/spend the tokens, various wallet systems offer various methods. Using some is simpler than using others. Some people more securely store and backup private keys [68]. Solana supports a variety of wallet kinds so you can find the ideal mix of ease and security.

2.7.1 File System Wallet Security

The most practical and least secure type of wallet is a file system wallet [69]. Because the keypair is kept in a straightforward file, it is practical. You are free to create as many keys as you wish, and you can easily backup those keys by copying the files. Because the keypair files are not encrypted, it is insecure. An FS wallet is a good option for modest amounts of bitcoin if you are the only user of your computer and are certain it is malware-free. But if your computer is online and has malware on it, the software could upload your keys and use them to steal your tokens. The keypairs are similarly kept on your computer as files. An alternative and suggestion to maintain the password secure is the Paper Wallet. A paper wallet is a compilation of paper-based seed phrases. A seed phrase is a set of words that can be used to instantly generate a keypair (usually 12 or 24 words).If we talk about convenience versus security scale from an FS wallet. Although extremely cumbersome to use, it provides superb security. When offline signature is combined with paper wallets, the level of security is increased even more. Solana has his own tool to create this security method called solana-keygen. It is possible to create fresh seed phrases with the solana-keygen tool as well as create a keypair from an already-existing seed phrase and (optional) passphrase. Together, the seed phrase and passphrase form a paper wallet. You can access your account using your seed phrase and passphrase as long as you store them securely [70].

Chapter 3

State of the Art

Different projects have been carried out with the infrastructure of multiple blockchains, ranging from the sale of cryptocurrencies to works of art and different product-origin traceability projects. In Ecuador, there are multiple visions of how transactions should be regulated and how they are valid here and anywhere in the world. Currently, a way is being sought to ensure that the transaction is carried out correctly and that it is included in the block registry without generating losses or unnecessary expenses or fees since we know that one block will have priority over another depending on what characteristics it takes into account on the blockchain. Currently, it is about solving problems through developing algorithms known as the Max-Min Fairness algorithm. Since the transaction spaces in the blocks, the storage, and the executions of smart contracts consume blockchain resources, options for a fair division of the block feed have been raised. Some have been able to obtain different algorithms for different use cases as optimal solutions. The respective studies can improve each case and his own absolute gas consumption values, taking the cost growth structure. This structure is not new since it is basically for distributing resources or equipment for certain system requirement tasks (e.g., CPU scheduling, bandwidth allocation, etc.). It can also be considered for the fair distribution of currency in faucet systems.

During the research was found different implementations of faucet, trying to minimise the cost of fees or how those fees can be distributed in a fair manner.

One of the articles that mention the problem is Max–min fairness based faucet design for blockchains by Serdar Metin and can Ozturan. They mention different improvements during the creation of algorithms where the fundamental part was focused on what they

can do in the blockchain and what it is not allowed. They mention that in the Ethereum blockchain Ecosystem every executed instruction has a gas consumption or cost. We can imagine during every transaction has a cost, each line of the routing address or registering into the blockchain has a cost. This can be a problem when the purpose of a Faucet is distribute different tokens and this transactions can cost. The challenge of fairness continued to grow in scope as a result of advances in distributed systems and new paradigms in cluster and high-performance computing, and new issues emerged. In this situation, service providers often charge customers for the shared resource that is allotted to and needed by them. The same query is now asked in terms of fairness in charging: how much should each demand cost, in order to be equitable among clients? Should all resources be priced equally? If not, how are they traded?. Those questions are part of the fundamental goal of the Max-min Fairness scheme is to increase the minimal share supplied to a user. Its technique is based on a trivial Fairness scheme, where resources are allocated equally among the demanders, with each demander receiving 1/n of the resource. On the basis that not every demander will demand as much as the share that is designated for them, Max-Min Fairness enhances the trivial method. To reallocate unused shares of the underdemanders among the overdemanders, the Max-min Fairness allocation algorithm repeats this process over the list of demanders.

On a typical gigabit network, a database execute around 710,000 transactions per second the average transaction size is no more than 176 bytes. Using the distributed system method known as optimistic concurrency control, a centralized database can also duplicate itself and maintain high availability without significantly affecting that transaction rate [71]. At Solana, we are proving that blockchain on an adversarial network is subject to the same theoretical bounds. When nodes are unable to rely on one another, figuring out a means to share time. 40 years of distributed systems study suddenly become applicable to blockchain once nodes can rely on time! all the process, algorithms and functions works decentralized to produce a centralized record of transactions. All processes operate asynchronously, while our method produces a globally synchronous one in which every process does the same thing .at (approximately) the same time. Our method seems to contradict the whole purpose of distributed processing, which is to permit different processes to operate independently and perform different functions. The nLocktime mechanism used in

Bitcoin allows transactions to be postdated using block height rather than a timestamp. If you do not rely on the network, a Bitcoin client would utilize block height instead of a timestamp. Block height turns out to be an example of what is known in the cryptography community as a Verifiable Delay Function. It is a secure technique to express the passage of time using cryptography. In Solana, we checkpoint the ledger and manage consensus using a SHA 256 hash chain, a far more precise verifiable delay function. We now have Optimistic Concurrency Control in place and are rapidly approaching the theoretical limit of 710,000 transactions per second thanks to it.

3.1 How Solana works?

How Solana works This project makes use of Proof-of-Stake (PoS) consensus with a plugin called "Tower CBFT". Through this mechanism, distributed networks are allowed to reach a consensus that prevents the attack of the Byzantine Generals (BFT).

Solana implement BFT resistance to adds a global source of time to the Blockchain using a second novel protocol called Proof-of-History (PoH).

Tower BFT makes use of this clock that features synchronization for reduced processing power requirements to verify transactions. This is possible thanks to the fact that it is not necessary to calculate the timestamp of the previous transactions. A mechanism that allows Solana to have a higher performance than other Blockchains.

It is not the only difference between Solana and the rest of the Blockchain. The parallelization capacity of the transactions that receives the name of Sealevel is also added. This mechanism allows the execution of smart contracts in parallel, thus optimizing the need for resources and reducing time. Solana supports horizontal scaling across GPUs and SSDs, allowing the platform to scale to meet network demand.

In addition, in Solana the mempool that is used in Bitcoin, Ethereum and other cryptocurrencies is eliminated. What is done is to distribute the transactions among the validators, even before they have finished validating the previous batch of transactions. Thanks to this, the speed of confirmations is maximized and the number of transactions that can be handled simultaneously and in parallel is increased. This element is called the "Gulf Stream".

3.1.1 POH example

Suppose Ana wants to send a transaction (TX) to Juan to pay for a repair on his laptop. At a specific time and date, Ana sends the TX data in the metadata of said TX. The Solana network takes that TX and begins its processing; however, in the entire network, there are hundreds and even thousands of operations, each with a specific creation time and date. What Solana does with PoH is to take each of these TXs and create a chronology, ordering them in the most exact way possible.

At this point, when the network orders the TXs for its next block, the validators apply in parallel a series of VDF functions to mark said block and begin with the generation of the next block. Here the differences of 0,1 seconds, can put you ahead or behind many other transactions within the network. This is vital because it prevents Ana from double spending (moving money that is temporarily not under her property, or altering a previous TX utilizing an attack vector such as that of Vector).

This other point is vital, because the temporary generation of the first block and its marking will be essential to create the cryptographic relationship with the second. In any case, the VDFs are in charge of verifying the TX timestamps, creating a unique hash for that block along with its own timestamp, and creating not only a chronology in the transaction block, but also a temporal order with the own blocks of the Blockchain.

That is, the TX and the blocks are temporally organized to the point that the temporal differences between them do not exceed 1 second, allowing the Solana Blockchain to be read as a timeline continues second by second that is recorded and is unalterable. In all this, The TX of Ana is validated and Juan receives his money. Although this explanation is extensive, this entire process within Solana has barely taken an average of one second to complete.

The previous example already tells us a lot about PoH and its speed, and it is that the block production time of the Solana network is by far one of the fastest that exists, of only 400 milliseconds. Solana can generate two blocks per second, with some time to build a third block. Put that way, Solana seems like science fiction for Blockchainers, but no, it is something so real that we can simply feel it while using the network.

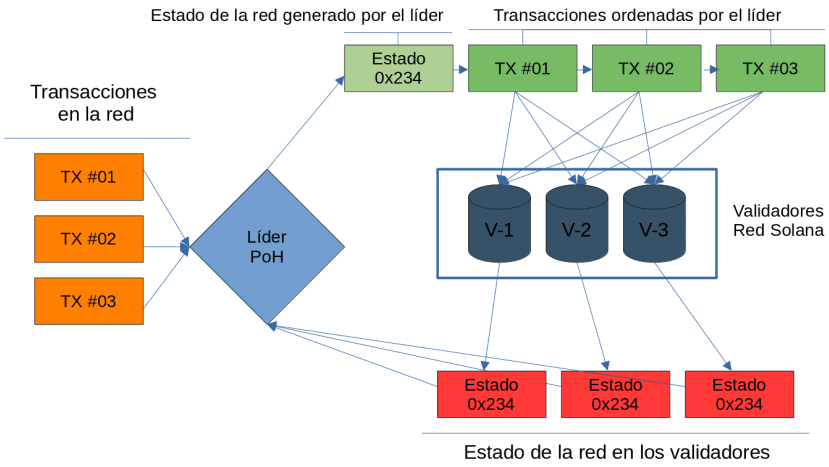

Figure 3.1: Transaction schema

3.1.2 Peculiarities

The main problem with current Blockchains using proof-of-work consensus is that they support very few transactions per second. Bitcoin supports between 5-8 transactions per second and Ethereum supports between 12-20 transactions per second. When these figures are exceeded, we enter into network congestion, which is nothing more than more transactions than can be processed. This entails increases in transaction fees so that they are validated before the rest of the issued transactions.

Solana differs from the rest of the Blockchain in offering, in theory, a greater processing capacity. The developers indicate that the figure of 50,000 transactions per second can be exceeded. Complying with this figure would mean that it would be the fastest and most powerful Blockchain today.

It is not the only difference between Solana and the rest of cryptocurrencies. Solana generates a block at 400-800[72] millisecond intervals. Bitcoin generates a block every 10 minutes (give or take) and Ethereum every 20 seconds (give or take). Transaction fees on Solana are also reduced, going to an average fee of 0.000005 SOL. These aspects would potentially allow the development of any DApp, DeFi solution or any game on this Blockchain.

It does all this without the need to resort to a second layer, as is the case with the Lightning Network for Bitcoin, a Layer 2 solution. In addition, it is not necessary to fragment the Blockchain into parts, as proposed by Ethereum 2.0 or Polkadot, among

others [72].

Solana also allows anyone to enter as a network validator and help protect the Blockchain. It is a permissionless system, but it requires a minimum amount of hardware that is described on the website project. There are currently more than 1,000 validator nodes on the network.

3.1.3 Differences between Proof of Work and Proof of Stake

Even though they are both consensus mechanisms that ensure the security of the blockchain network, there are certain differences between them. The main difference is, of course, how PoW and PoS determine which participant validates new transactions. For a clearer understanding, this is the table where the comparison is made:

Differences between Proof of Work and Proof of Stake		
	Proof of Work (POW)	Proof of Stake (POS)
Who can mine or validate blocks?	The higher the computational power, the higher the probability of mining a block	The more coins you have in stake, the more likely you are to validate a new block.
How is a block mined/-validated?	Miners compete to solve complex mathematical puzzles using their computational resources	Normally, the algorithm determines the winner randomly, taking into account the number of coins staked.
Mining Equipment	Professional mining hardware, such as ASIC, CPU and GPU.	Any computer or mobile device with internet connection.
How are rewards distributed?	The first person to mine the block receives a block reward.	Validators can receive a portion of the transaction fees collected from the block they validated.
How is the network protected?	The higher the hash, the more secure the network.	Staking locks cryptocurrencies on the blockchain to protect the network..

3.2 Solana clusters

A cluster is a collection of connected computers that operate as one unit when viewed from the outside. A group of independently owned computers that collaborate (and occasionally compete) to verify the results of dubious user-submitted programs is known as a Solana cluster. Any time a user needs to keep an immutable record of past events or programmatic interpretations of those events, they can use a Solana cluster. One purpose is to keep track of which computers performed significant effort to maintain the cluster. Tracking the ownership of tangible items may be another purpose. Each time, the cluster creates a

ledger, which is a log of events. Solana maintains different clusters with specific purposes.

3.2.1 Devnet

the Devnet is similar to a playground where any one who wants to take a similar environment as Solana. The Devnet is Solana for a test drive, as a user, token holder, app developer, or validator.It is necessary to any application or potential validator start working with the Devnet some characteristics for this cluster are:

- Devnet tokens are not real
- Devnet includes a token faucet for airdrops for application testing
- Devnet may be subject to ledger resets
- Devnet typically runs the same software release branch version as Mainnet Beta, but may run a newer minor release version than Mainnet Beta.

There is an specific port and direction for this Devnet or entrypoint: ***entrypoint.devnet.solana.com:8001.***.

3.2.2 Testnet

The Solana core contributors stress test recently released features on the "Testnet," a live cluster, with an emphasis on validator behavior, network performance, and stability. Some characteristics are:

- Testnet tokens are not real
- Testnet may be subject to ledger resets.
- Testnet includes a token faucet for airdrops for application testing
- Testnet typically runs a newer software release branch than both Devnet and Mainnet Beta

There is an specific specifications for this Devnet or entrypoint:

```
1 \textbf{\textit{entrypoint.testnet.solana.com:8001}}.
```

Also there is an specific RPC URL for Testnet:

```
1 https://api.testnet.solana.com
```

Here is important to follow the example due to is necessary to understand the command line configuration:

```
1 solana config set --url https://api.testnet.solana.com
```

3.2.3 Mainnet Beta

The Mainnet Beta cluster is a durable, permissionless cluster for token holders, builders, validators, and users of Solana.

- Real SOL tokens are issued on the Mainnet Beta.
- The location of the Mainnet Beta gossip server is: entrypoint.mainnet-beta.solana.com:8001
- Also there is an specific RPC URL for Mainnet Beta: https://api.mainnet-beta.solana.com

To use this specific cluster follow this example:

```
1 solana config set --url https://api.mainnet-beta.solana.com
```

Ensure Versions Match

An important aspect to take in mind is to ensure the versions match between the cluster version and the local cli version. Although it is not technically necessary, it is often desirable for the CLI to run on the same version of software as the cluster. Run: to obtain the CLI version that is installed locally.

```
1 >>> solana --version
2 >>> solana cluster-version
```

If the version is different is a requirement that the local CLI version is greater than or equal to the cluster version.

3.3 Send and Receive Tokens

Make sure you have generated a wallet, have access to its address (pubkey), and have the signing keypair before you start. Check out our keypair entry conventions for various wallet kinds. [73]

3.3.1 Testing your Wallet

Is mandatory to start sharing the public key generate during the creation of the wallet.You might wish to confirm the validity of the key and that you actually own the accompanying private key first. The way to check the wallet is valid is transferring some tokens from other wallet. In this context is necessary to use an existed wallet or create a new one in test using Devnet, or developer Testnet Do not worry if you lose any of the tokens you received from Devnet; they are invalid. As a first step is requiered to airdrop yourself some play tokens on the Devnet using the following command line.

```
1 >>> solana airdrop 1 <RECIPIENT_ACCOUNT_ADDRESS> --url https://api.
    devnet.solana.com
2 ###where you replace the text <RECIPIENT_ACCOUNT_ADDRESS> with your
    base58-encoded public key/wallet address.
3 >>> solana confirm -v <TRANSACTION_SIGNATURE>
4 >>> solana balance <ACCOUNT_ADDRESS> --url https://api.devnet.solana.
    com
5 >>> solana-keygen new --no-passphrase --no-outfile
6 ### output pubkey: GKvqsuNcnwWqPzzuhLmGi4rzzh55FhJtGizkhHaEJqiV
7 >>> solana transfer --from <KEYPAIR> <RECIPIENT_ACCOUNT_ADDRESS> 0.5
    --allow-unfunded-recipient --url https://api.devnet.solana.com --
    fee-payer <KEYPAIR>
8 ###where ‘‘RECIPIENT ACCOUNT ADDRESS" is the address of your second
    wallet, and ‘‘KEYPAIR" is the path to a keypair in your first
    wallet.
9 >>> solana balance <ACCOUNT_ADDRESS> --url http://api.devnet.solana.
    com
10 ### where <ACCOUNT_ADDRESS> is either the public key from your keypair
     or the public key.
```

This step is required to start working with any app, development or proof inside of the system is a form of test.

3.4 Programming Model

A Solana cluster can be contacted by an app by sending it transactions with one or more instructions. These directives are sent to applications that have already been deployed by app developers via the Solana runtime. A software may be instructed to move lamports from one account to another, for instance, or to establish an interactive contract that specifies how lamports are transferred. For each transaction, instructions are carried out atomically and sequentially. All account changes made during the transaction are discarded if any instruction is deemed incorrect.

3.4.1 Transactions

A transaction is sent to the cluster to start the execution of the program. Each instruction in the transaction will be processed atomically and in order by a program that is run by the Solana runtime. a transaction has an structure defined to get performed. one of them is the format:

Transaction Format

A message is followed by a small collection of signatures in a transaction. The given digital signature message is represented by each item in the signatures array. The Solana runtime checks to see if the number of signatures matches the number in the message, the first 8 bits of the header. Additionally, it confirms that each signature was created using the private key that has the same index in the array message of account addresses as the public key. Signature Format: each signature use the ed25519 binary format and use 64 bytes

Message Format

The message is the way of how is processed the addition on a new block, because is requiered specific information. The information is composed by a header, a compact array of account addresses, a recent blockhash, a compact array of instructions.

Message Header Format consists of three 8-bit unsigned values. The number of necessary signatures for the enclosed transaction is the first value. The number of read-only associated account addresses makes up the second value. The number of read-only account

addresses that do not require signatures is the third value in the message header.

Account Addresses Format The account address array starts with the addresses that need signatures, followed by read-only accounts and addresses asking for read-write access. Similarly, read-write accounts come first, followed by read-only accounts, for addresses that do not require signatures.

Blockhash Format A 32-byte SHA-256 hash is present in a blockhash. It serves as a reminder of the last time a client saw the ledger. Transactions will be rejected by validators if the blockhash is too old. [74]

3.4.2 Instruction Format

A compact-array of account address indexes, a program id index, and a compact-array of opaque 8-bit data are the components of an instruction. A software that can decipher the opaque data is located using the program id index. The array message of account addresses contains an account address that the program id index, which is an unsigned 8-bit index, points to. An unsigned 8-bit index into that same array is what the account address indexes.[75]

- Compact-Array Format .- Each array item is serialized after a compact-array, which represents the length of the array. A unique multi-byte encoding with a 16-byte length is known as compact-u16.
- Account Address Format .- 32 bytes of random data make up an account address. The runtime interprets an address that needs a digital signature as the public key of an ed25519 keypair.

3.4.3 Instructions

A single program, a subset of the account transaction to be provided to the program, and a data byte array are all specified for every instruction. The program decodes the data array and executes the instructions on the accounts that are mentioned in the instructions. Either a successful return or an error code may be made by the software. Any transaction that receives an error return immediately fails. [76]

Program Id.- Which program will execute this instruction is indicated by the program id of the instruction. The data contains details about how the runtime should carry out the program execution, and the account owner selects which loader should be used to load and execute the program. The BPF Loader is the owner of on-chain BPF programs, and the account data contains the BPF bytecode. When program accounts are successfully distributed, the loader permanently marks them as executable. Transactions that specify non-executable programs will be rejected by the runtime. Native Programs are handled differently from on-chain programs because they are integrated into the Solana runtime.

Accounts.- The accounts that an instruction refers to are the inputs and outputs of a program and represent on-chain state.

Instruction Data.- The accounts are supplied to the program along with a general-purpose byte array that is carried by each instruction. The information contained in the instruction data is program-specific and is often used to specify the operations the program should carry out as well as any additional details such activities might require beyond what is provided in the accounts. The method of information encoding into the instruction data byte array is left up to the programs. Since data decoding is handled by the application on-chain, the method of encoding should take that into account. Some popular encodings, like Rust bincode, have been found to be extremely wasteful. The Token program from the Solana Program Library is one illustration of effective instruction data encoding.

Multiple instructions in a single transaction.- Instructions can be included in a transaction in any sequence. This means that a malevolent user could create transactions that could contain instructions that the application is not designed to handle. Programs should be strengthened so they can handle any potential instruction sequence correctly and safely. Account deinitialization is one obscure illustration. By setting an lamports account to zero, some programs may try to deinitialize it in the hopes that the runtime will destroy the account. This presumption might hold true while comparing transactions, but not when comparing instructions or cross-program invocations. The application should also explicitly zero out the data account to defend against this.

Signatures.- All account public keys referred to in a instructions transaction are explicitly listed in each transaction. Each of those transaction signatures is attached to a subset of those public keys. These signatures inform on-chain programs that the transac-

tion has been approved by the account holder. The authorization is typically used by the software to allow debiting the account or changing its data.

Recent Blockhash.- In order to avoid duplication and provide transactions with lives, a transaction includes a recent blockhash. Adding a newer blockhash permits numerous transactions to repeat the same activity as any transaction that is entirely similar to a prior one is refused. As any transaction whose blockhash is too old will be rejected, transactions also have lives that are determined by the blockhash.

3.4.4 Issues Facing Enterprise Adoption of Blockchain

This paper will not cover whether blockchain technology fits enterprises from the business perspective, although that should be the first question to ask. We will focus on non-functional requirements that describes the system's operation capabilities and constraints that enhance its business functionality. The non-functional requirements needed for application will of course depend on the business context and the outcomes to be achieved, particularly as there are so many that can be applied. In this paper, we will only elaborate a few most critical ones. Performance. All enterprise systems should be designed and built with an acceptable standard of performance as a minimum, while taking into accounts problems such as scalability, latency, load and resource utilization. Many factors could negatively impact performance, including high numbers of API calls, poor caching, and high-load thirdparty services. It's critical to ensure the end-user experience or integration of multisystems across the entire eco-chain is not affected by any such issues. Prevailing business transaction systems have been capable of processing thousands (Visa, for example) or millions of transactions (online market place such as Amazon or Alibaba) per second without any failure, most of the current blockchain platforms depicted a remarkable slowdown, making them unviable for large-scale or performancesensitive applications. For example, Bitcoin can only process roughly 3 to 7 transactions per second, with Ethereum about 15 to 20 transactions. Such poor performance and cumbersome operations are mainly due to the complexity with encrypted and distributed nature in blockchains. Although it is not at all suitable for high-frequency transactions, ways to improve its transaction performance, including throughput and latency, is always a hot topic. Compared to "traditional" payment systems such as cash or debit cards, it could take hours or even days to process

some transactions. When more users join the network, its performance will be further degraded due to the existence of consensus latency from nodes with low processing power. As a result, the transactions cost is higher than usual, further limiting more users onto the network.

Scalability Scalability is the second big issue that needs to be addressed, as this is one of the core reasons why organizations still hesitate to adopt blockchains.

The system must be able to accommodate ever-increasing volumes (number of users/devices/integrated applications, data and throughput) over time, and is able to scale up and down quickly as the number of users change drastically, as needed. Security and Integrity. Requirements such as confidentiality, authentication and integrity ensure that valuable (private and confidential) information is protected. Blockchain benefits primarily derive from the trust it fosters, its built-in privacy, security and data integrity and its transparency, as it incorporates a flow of data from complex mathematical operations that cannot be changed once created without being detected, and every transaction is encoded and connected, and therefore it is significantly more reliable than traditional journal methods. This unchangeable and incorruptible characteristic inherently make blockchains safer and better protected against tampering and hacking of information. Various software engineering tactics can be employed to safeguard valuable/transactional data at many integration points. System architects need to understand legal and compliance requirements and communicate these clearly to the development team, so that the necessary levels of security can be established and enforced jointly. With blockchains, an external audit can be provided from the distributed ledger. This will inherently enhance privacy and avoid corruption, and help confirm the legitimacy of transactions and offer indisputable proof of transactions. Availability/Reliability/Resilience. The system must be available for use, and the downtime must be reduced to an acceptable level under any circumstances. For example, mechanisms to avoid single points of failures, and adequate timeouts could be used to enhance system availability and reliability. Feasibility. Feasibility considers issues such as technology maturity, time-tomarket, total cost of ownership, technical knowledge, and migration requirements. Commercial-off-the-shelf (COTS) solutions, managed services and cloud-native functions where appropriate, and close collaboration with development partners with suitable architecture and solution components and services will definitely

help address those issues. This paper surveyed several important blockchain platforms covering the years of evolution from the original Bitcoin system to the more advanced recent offerings. Hopefully, with the information we collected and analyzed, it could help enterprises to make better decisions, while also directs new players where to innovate in order to make blockchain well fit into most enterprises business needs. We will review the chosen frameworks, especially their data structures, processes and algorithms involved in creating a new transaction record (block), and how conflicts or disputes could be resolved in We will also raise concerns on several key issues related to the afore-mentioned NFRs, especially the performance and scalability. Section 3 then proceeds to analyze the selected platforms and discuss, from the evolutionary nature of blockchain technology since its inception, how critical issues such as performance, and scalability etc. were addressed, especially the most recent advances from Solana, where 24 order of magnitude of improvements has been proved possible. Section 4 will first present a quick summary view on how enterprises could leverage the information collected and analyzed in the maker to choose "better-fit" platforms, then points to some remaining issues that should be further evolved or even revolutionized to truly meet some fundamental NFR requirements for enterprise adoption. Some alternative approaches to achieve the secure and immutable nature of the distributed ledger is also included. Section 5 conclude the paper with a few quick remarks.

Chapter 4

Methodology

4.1 Phases of Problem Solving

To solve the problem, it is necessary to be able to identify the actors that generate difficulties for development. Knowing the infrastructure and operation of the Blockchain, it will be possible to access and complete a transaction successfully.

The analysis for its implementation entails the knowledge of how it works and the requirements that this system needs to accept records and transactions within the Blockchain.

4.1.1 Description of the Problem

The registration and use of Blockchain technology entails many operations, validations, entry into internal production, integration of technologies and above all a series of new developments since what is proposed is the integration of a web page with the chosen Blockchain in which in real time can calculate the different costs of transactions, queries and integration of new records to the Blockchain, this is of great importance for development since its integration cost cannot be validated before it is measured Its cost. The hardest part of the development is to respective know everything about his complex systems and how implement the idea as best it can be implemented, during the process of recapitulation and extraction of the content was possible to understand and get a better idea of how to do it.

4.1.2 Analysis of the Problem

1. Assume the technology is ready for production use. The blockchain platform market is huge and largely made up of fragmented offerings trying to differentiate themselves in various ways. Some focus on confidentiality, some on tokenization, some on universal computing, method of validation. Most are too immature for the large-scale production work that comes with the accompanying systems and requirements, security, and network management services. However, this will change in the coming years. CIOs should monitor the evolving capabilities of blockchain platforms and align the blockchain project duration accordingly.

2. Confusing a protocol with a business solution. Blockchain is a basic technology that can be used in a variety of industries and scenarios, from supply chain to management and medical information systems. It is not a complete application as it must also include features such as user interface, business logic, data persistence and interoperability mechanisms. When it comes to blockchain, there is an implicit assumption that the underlying technology is not too far from a full application solution. This is not the case. Blockchain is understood more as a protocol to perform a certain task within a complete application. Nobody will assume that a protocol can be the only basis for an entire electronic commerce system or a social network.

3. See Blockchain simply as a database or storage mechanism. Blockchain technology was designed to provide a reliable and immutable record of events arising from a dynamic collection of untrusted parties. In its current form, blockchain technology does not implement the full "create, read, update, delete" model found in conventional database management technology. Instead, only "create" and "read" are supported. Programmers and CTOs or CEOs need to assess the data management requirements of their blockchain project. A conventional data management solution might be the best option in some cases, in our case is necessary to use based on the investigation realized during the background

4. Assume that interoperability standards exist. While some blockchain technology platform providers talk about interoperability with other blockchains, it is hard to imag-

ine interoperability when most platforms and their underlying protocols are still being designed or developed. Organizations should view vendor discussions of interoperability as a marketing strategy. It is supposed to benefit the competitive position of the vendor, but it may not necessarily benefit the end-user organization. In my case never select a blockchain platform with the expectation that it will interface with next technology year from a different vendor.This technology can be improved in different ways for that reason is important to select the blockchain thinking into the versatile.

5. Assuming that smart contract technology is a solved problem. Smart contracts are perhaps the most powerful aspect of blockchain enabling technologies. They add dynamic behavior to transactions. Conceptually, smart contracts can be understood as stored procedures that are associated with specific transaction records. But unlike a stored procedure in a centralized system, smart contracts are executed by all nodes in the peer-to-peer network, leading to scalability and manageability challenges that have yet to be fully addressed. Smart contract technology will continue to undergo significant changes. CIOs do not yet need to plan for full adoption, but they should conduct small experiments first. This area of blockchain will continue to mature over the next two to three years.

6. Ignore governance issues. While governance issues on private or permission blockchains will generally be handled by the blockchain owner, the situation is different with public blockchains. Governance on public blockchains like Ethereum and Bitcoin is primarily aimed at technical issues. Human behaviors or motivation are rarely addressed. CIOs need to be aware of the risk that blockchain governance issues can pose to the success of their project. Especially large organizations should consider joining or forming consortia to help define governance models for the public blockchain.

4.2 Analysis of Solution

Based on the previous sections, the following analysis can be performed. Keep in mind that a user can invoke programs (or smart contracts known in Solana) by sending a transaction

to a cluster. A transaction can be a single instruction, such as querying the value token, or it can include multiple instructions, each with its configuration and pointing to a different program. Every time a program runs, the Solana Runtime processes the instructions in order and in an atomic way. Each execution has to be processed correctly in its entirety. If a small part fails, the whole transaction fails. The complexity of creating the faucet In order to carry out this implementation, the following technical aspects had to be taken into account. Each instruction will be processed as most basic unit of operation in Solana. Each instruction must contain: The programid (program id) of the program it points to An array of all accounts (accounts) where it will read or write. An instructiondata (instruction data) which is a byte array for the specific program. Multiple instructions can be carried out at the time of carrying out a program or transaction. Each transaction contains a series of essential elements: An array of all accounts (accounts) where it will read or write One or more instructions, A recent blockhash. One or more signatures, the execution of the instructions are carried out one by one and in order of writing in the code. If even one part of the instructions fails immediately, the transaction fails. Instructions contain a maximum of 1232 bytes.

4.2.1 Solution Design

The solution design is based on the development of a easy connection of an specific wallet that contains all the tokens generated, the characteristic is that every transaction has to have a minimun amount of cost and has a high performance. For this solution is required use the official documentation about the tech and systems. We are going to consume some resources from different APIs or clusters and connect to different platforms and wallets just to check that the process is running correctly. Then the process can be described in the following steps and then connect between them to get the expected functionality:

- Implementation of the native tokes.
- Implementation of the web Faucet.
- Connection, communication with the Solana Blockchain.

- Proof of the records-transactions inside of the Blockchain represented in the Web Faucet.

4.2.2 Implementation Analysis

The Solana Runtime requires statements and transactions to specify a list of all accounts to read from or write to. By requiring these accounts in advance, the Solana Runtime can parallelize the execution of all transactions. When a transaction is submitted to a cluster, the Solana Runtime will process its instructions in order and atomically. The called program will interpret the data array and operate on the specified counts for each instruction. The program will return success or can be informed by catch error code. The entire transaction will fail immediately if an error is returned. Any transaction that debits an account or modifies its data requires the account signature of the holder. Any account to be modified will be marked as writable. An account can be funded without the permission of the owner if the fee payer covers the rent and transaction fees. Before sending a transaction, a recent block hash must be sent. The recent block hash is used to prevent duplicates and eliminate stale transactions. The maximum recent block hash time of a transaction is 150 blocks or approximately 1 minute 19 seconds at the time of this writing. Each transaction requires considering the charges to be executable since it can be an operation with writable or readable attributes. The Solana network collects two types of charges:

- Transaction fee to propagate transactions (*i.e.*,"gas fees")
- The rental charge for storing data on the blockchain

At Solana, charges are deterministic, there is no concept of a charging market where users can pay higher rates to increment their chances of being included in the block. Currently transaction fees are determined solely by the number of signatures required (*i.e.* lamportspersignature), not the number of resources used. This is because there is currently a maximum limit of 1232 bytes on all transactions. All transactions require at least one account to be writable to sign the transaction. Once submitted, the signer account that is serialized first will be the payer of the charge. This account will pay the cost of the transaction regardless of whether the transaction succeeds or fails. The transaction will be

canceled if the charge payer does not have a sufficient balance to pay the transaction fee. At the time of this writing, the validator that produces the block collects 50 percent of all transaction fees, while the remaining 50 percent is burned. This structure incentivizes validators to process as many transactions as possible during their slots in the lead program.

4.2.3 Faucet Implementation

After the generation of the native development tokens, they are placed into a specific digital wallet address. Any platform can be used to be able to store them, of which we must be clear about the address that contains the tokens. It is required that access to the tokens be available at any time and have the ability to be transferred to a specific account. For this, there must be a continuous login to the account and that the deposit addresses of the tokens can be embedded at that time. The complication falls on the permanence of the wallet and that the accesses to the system are not recorded within the code. the platform must present sufficient information detailing how to carry out the transaction and at the same time be able to have access to the tokens of the project. The development of the faucet could have been defined as a reward system, in the form of a website or an app, from which small amounts of a certain Cryptocurrency can be obtained. These prizes are delivered to the user after performing a task that could include: Waiting a period of time since the last reward or receiving the reward only once. Solve a captcha, simple tasks (such as watching ads) or games Make deposits within the platform to receive more rewards.

This is certainly something that starts to promote a new Cryptocurrency or a project is highly attractive and the more that coin is distributed, it gains popularity and more value. Its execution entails development and being able to make the necessary connections so that the transactions can be carried out with the different users. for the faucet it is developed with css and html modules and features in JavaScript. the node modules to be able to control the flow and consumption of different APIS for the development of token transfers. Risks of faucets Downloading suspicious files We can find some faucets that ask us to download some type of file. It is possible that it contains some kind of virus or malware. Theft or misuse of personal information Some faucets require the user to register with an email and password in order to start using it. This may mean being a victim of Phishing attacks or misuse of the information given to the platform. scam faucets As we

have mentioned in several points, it may happen that the faucet does not pay what was promised and, therefore, the time invested in the platform has been in vain.

4.3 Implementation

On any blockchain like the Ethereum Mainnet, Solana or Bitcoin, the fees represented in the units of the blockchain cryptocurrency must be paid in order for transactions to be performed and recorded. For that reason this project its going to be pivoting from Testnet of Solana to Devnet when the process are success those changes and final code can be proved.

4.3.1 Native Token Implementation

The implementation process is described as a serial of steps than can give an approximation and explanation of what is every part that compose a block chain in this case Its used the Solana Blockchain that has many qualities that improves the development and implementation of native tokens and, smart contracts

System

Linux machines or Debian based machines the usage of this operative systems are structured to be suported on most of the Blockchain

Creation of Eternum

SOL (Solana) at least five dollars is required to create a native token. You can buy SOL on Coinbase or any platform that use the SOL as current Cryptocurrency, always is good to know the amount that you can use for those minimum transactions, those depends on the platform but usually you can spent at least 1 to 2 dollars per transaction.

```
1 https://www.coinbase.com
```

STEP 1 - Setup the Linux Machine on Linode or use a Linux distro machine for the following steps. To use Linode is really easy and fast to configure any distro machine, you just pay the amount the time that you use the machine. The other face of the coin is that

you have to pay for the usage but relatively this can also be really cheap. The following link can be used by 100 dollars in credit.

```
Linode ---> https://ntck.co/linodeaff
```

STEP 2 - Create a Solana Wallet (from the command)

1. Run this command to install Solana tools

```
--->$sh -c "(curl -sSfL https://release.solana.com/v1.8.6/install)"
```

2. Exit the terminal and reconnect.This is required to see the creation of new files that allows the creation of the wallet.

3. Create your Solana Wallet. The creation of the wallet will create a 12 word pass code in an order in specific, for this step is requiered to save manually in any book those 12 words to recover the wallet as a recover factor. Also this step will require a password.

```
--->solana-keygen new
```

4. Check your Solana balance.

```
sunshine balance
```

STEP 3 – Buy Solana and send it to the Wallet created in the terminal. The account that was created with the *You can buy Solana on Coinbase

```
---> https://www.coinbase.com/join/moline_6
```

STEP 4 - 1. Update repositories

```
--->sudo apt update
```

2. Install RUST.- The current version can be found in the main page of rust, also here is providing Playground without installing anything on your computer. As was recommended in the previous steps the best way to uses Rust is to apply a subsystem for Linux or any of their distro. Also is recommended if the usage of Rust its going to be deploy use the Rust installer and version management tool this is located in the main page but is quite difficult to develop on this environment, the recommendation its going to be the usage of the Windows subsystem for Linux.

```
---> curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

3. Select 1 for the standard installation option. 4. Exit the terminal and reconnect. 5. More things to install:

```
1 --->sudo apt install libudev-dev -y
2 --->sudo apt install libssl-dev pkg-config -y
3 --->sudo apt install build-essential -y
```

STEP 5 – Install the Solana Token Program (CLI tools)

1. Install the tools (this takes a moment)

```
1 --->charge install spl-token-cli
```

STEP 6 – Create your token!

1. Just one command:

```
1 --->spl-token create-token
```

---

2. Create an account for the token

```
1 --->spl-token creat-account yourtokenaddress
```

---

3. Print (create) some tokens!!

```
1 --->spl-token mint yourtokenaddress amountyouwanttomint
      yourtokenaccount
2
3 Example:
4
5 --->spl-token mint vXV3qR4WifEPiteogi9FDQveVthwrvEeRxpjyQdDHwq 500
6 3PgHtwNjNx4QzL5MfeTGLJkLCVwA7sotvm5MVXSyQvfj
```

---

4. Check your token account balance.

```
1 --->spl-token accounts
```

---

5. ***OPTIONAL***Disables future printing to limit number in circulation.

```
1 --->spl-token authorize yourtokenaddress mint --disable
```

STEP 7 – Transfer your token to another wallet.

1. Create another wallet (phantom wallet) https://phantom.app
2. Transfer some tokens to that wallet:

```
1 --->spl-token transfer --fund-recipient --allow-unfunded-recipient
2 yourtokenaddress amount recipientwalletaddress
```

---

example:

```
1 --->spl-token transfer --fund-recipient --allow-unfunded-recipient
2 vXV3qR4WifEPiteogi9FDQveVthwrvEeRxpjyQdDHwq 10000
3 6Cc2MFmjjyXyenyEcKoBSq1Rdyw9t65yjXXBQWn34oLY
```

STEP 8 – Add your Token to the Solana Registry This step is really important to show the actual permanence of the token creation. The repository used is a accessible to every person but the admission of changes is precise and are controlled by the creator of Solana system.

- 1. Create your logo (less than 200kb and 300x300px) in this case I create a new one that represent the idea of decentralization and the main factors on the Solana Blockchain.

Figure 4.1: Eternel Coin Logo.

- 2. Create a GitHub account (if you do not have one yet)

```
1 --->https://github.com .
```

- 3. For the Solana Labs Token List repository go to:

```
1 https://github.com/solana-labs/token-list
```

This is the main repository that its going to be a fundamental part of the register of the native token, inside of the main net of the Blockchain the registers are going to be part of how to identify special tokens who is the owner and also where can

- 4. Type the key . to enter the web VScode editor ADD a capture
- 5. Create a folder in assets/mainnet with the address of your token as the folder name. in this case the folder name has to be verified with the previous commands ADD the capure
- 6. Inside the folder create a new folder called logo, in this folder has to be added the logo that you already created in the step 1 also called as logo, the extension of the file could be png or jpg

4.3.2 Implementation of the Web Faucet

For the development of an application within the Solana blockchain, three actions will always be carried out: 1 add a wallet, 2 send money and 3 query the blockchain to know the prior transactions, look at the balances, and look at metadata. These three actions are basic for the development of any application. The final code will be available for testing.

```
1 https://github.com/8rch/api.git
```

Front-End

The following components work together but are separated. Each one will be described its functionality and the importance it has:

- sender.tsx .- It is the main graphic part that shows in the form of boxes the elements that are required to carry out the Transaction.
- transactionView.tsx .- This is the graphic part that will represent all those lists of transactions that have been made.
- app.tsx .- Where the connections with the Wallet and the Cluster that will be required are initialized and configured, I seek to have the list of transactions updated so a function is actively updating the list of transactions only when the money has been sent.

This is how the first part of the FE is basically developed, which is the visual part whose objective is to show in a simple way an interface that has the required information.

After this development comes the implication of analysis closer to the Solana environment and how I carry out the guided implementation of the documentation that the Solana blockchain has.

wallet.ts

The ability to connect a wallet and send money from that wallet has been developed. The easiest way to connect to an electronic wallet is a wallet adapter. In this case, an import is made from a project called serum import Wallet from "@project-serum/sol-wallet-adapter". In order to carry out the configuration, we create a type due to the fact that Type-Script will be used, and the wallet adapter is still completely developed in Java-Script added a @ts-ignore due to the fact that, as mentioned, the types are not described in the wallet-adapter, and the Type-Script compiler must ignore it.

The documentation tells us that in order to make the connection, we must create objects and complete the necessary data structures or properties, one of them being very important: public keys. For the wallet, which is the main address, this is when you want to sign for something, so if you are trying to debit from your account and send money to somebody and then this is you must call this in order to make the actual connection to the network you can not just create an instance of this guy you have to actually make the call here. Inside the configuration is used the Devnet cluster and a default pair equal to confirmed that is required to inject queries into the blockchain. In my opinion, an instance of the wallet is required where in my case, 'Sollet' has been selected, which requires being joined to the name of the cluster to be used in order to control the number of connections. In this case, the code only loads at the first load. Within this file, I will include the function to send money that requires the address and the number of tokens that are required to be sent. It is necessary to carry out control of the values that are required and to be very careful since the addresses can be entered in the boxes because they may contain spaces, and errors must be contained. The functions will not be created from scratch like a Smart Contract because the development by Solana streamlined many functionalities, and therefore only the use of the necessary ones will be invoked to be able to carry out these transactions and queries to the blockchain. In order to validate said Transaction, a slightly longer process must be carried out, so a new section called Transaction will be

created, which will take part of the information received in instructions, modify them and give way to a new object called Transaction. A new transaction instance is created, add the new instruction inside it, the feePayer parameter will be defined as the carrier of the wallet. One of the most important parameters for validation is the hash, which, in addition to being the only way to connect to the block, makes it the only one that guarantees the continuity of block generation (function called getRecentBlockhash()) As a parameter that must be passed clean, it is also the origin of the hash block. It must be abstracted as a part of the object that we are going to pass as a transaction. Once the new object has been obtained and which will be used to initialize the transaction process, it must pass one more validation through the signature that will be of type string For this signature function, a transaction injection is required and it must be serialized in order to obtain its value and complete the object asynchronously. As one more method in this part, it is necessary to be able to know the result, if it is made or not, the connection will skip the send money message and the result of the connection. To control errors, a catch error is defined. So in this developed file basically three main things have been ensured, such as: we are attempting to transfer the money with an instruction we update the Transaction and we sing it If something during the process fails, it is contained within the catch error.

transaction.ts

section has built in the ability to query the blockchain and get that resulting transaction information this functionality is basically contained within the getTransactions() function The first part of this code was to create a custom type that would represent a bit of data that came as a request. This constructor takes the data already obtained from the wallet.ts file and uses it in order to get a clean query that goes to the blockchain. So in order to build the getTransactions() function we need which cluster to request, which address was the one that made the sending. we will obtain as a result a list of signatures. We will iterate this list of signatures by means of a for which will allow us to enter the Transaction individually for each signature and we will validate said Transaction with the previously created function getConfirmedTransaction(), because if this Transaction is not null or empty we can populate our function with the values we require.

4.3.3 Testing

For these space I propose 2 parts to realize the testing section, the fist one is to check the creation of the native token inside the Solana Blockchain and the second one is the validation of the transactions inside the Faucet web page, to present this part is valid to mention that its going to be used the playground in Solana, that can be used due to the cost of the transactions, to perform the testing it can be executed on the Devnet of Solana.

The testing of the creation of the token Eternum can be performed with the following step:

```
1 Go to:
2 https://solscan.io/token/Bo1hj1f1ms62Cbps9yJxUwCYmsDjXQqjJBq4K9kRKG9w
```

There you can find the register of the token into the blockchain but if you want to see some Eternum in your wallet you can use the step number seven of the section '4.3.1 Creation of Eternum'

For the transaction using the Faucet is requiered to use the repository attached here, where the configurations are based on the following steps:

1. Clone the repository.
2. Install the package with npm install from the root folder of the cloned repository.
3. Comment line 19 or 20 of the 'wallet.ts' file to do the tests from Testnet or Devnet. If we use Tesnet it is required to use lamports (it is a fraction of the value of a Solana, and in Testnet you can use several without transaction cost [77]) as a transaction token, and if we use Devnet we can use the token created in this repository using the wallet that we created in the project.

4.4 Model Proposal

I became more interested in this sector due to how rapidly blockchain technology is developing and widely used. The need for blockchain engineers is always growing, and I believe this trend will continue for a long time. For that reason, I present a reusable implementation for projects that can use this code for any commercial purpose or investigation. The

creation is primarily based on Solana Documentation, and the difference is based on the capability to switch between Devnet and Testnet. It can be used locally for testing and checking if the procedure is correct or move directly to the real Blockchain and use different tokens.

The implementation section of the thesis was intended to address the practice-based research questions by describing how to programmatically produce native tokens and how to build a faucet website that will be used to advertise native tokens following best practices.

4.5 Experimental Setup

For this section, we can describe that we use the Solana dev tools, which are installed in the windows WSL, since it is easier to install dependencies and run commands within Solana. Also, the computer used for this thesis has the following characteristics.

4.5.1 Machine Specifications.

- Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz
- Installed RAM: 16.0 GB
- Edition: Windows 11 Home
- System Type: 64-bit opertaing system, x64-based processor

4.5.2 Clusters used.

The cluster is based on the stages in development specially on the cases of testing and pushing to development.

- Testing = RPC URL for Testnet: https://api.testnet.solana.com
- Development = RPC URL for Devnet: https://api.devnet.solana.com

4.5.3 Libraries

The Solana-Web3.js is the library to provide complete coverage of Solana that is used in this project. The library was built on top of the Solana JSON RPC API. Using the Wallet

adapter from “@project-serum/sol-wallet-adapter” this is the easy and safe way to create the wallet connection component.

Chapter 5

Results and Discussion

5.1 Generation of the Tokens

Generation of tokens is a process directed utilizing a series of specific conditions of the Solana Blockchain. It must have been generated locally from the computer to demonstrate its existence. Perform a transactionability test and then register within the Solana network. To register within the Solana network, we must go to the following link.

```
1 https://github.com/solana-labs/token-list.git
```

When registering, you can see the name, icon, the number of generated tokens, and at the same time, how many transactions have been made. It is interesting how a platform already has an interface on how to perform the validation of this data. Initially, the size and weight of the file were not specified in the logos, and a trick that needs to be mentioned is that the specification of the token name should not be put at the end of the queue because it is ignored by configurations to carry out the token registry merge. At the same time, it was a process of reviewing documentation and creating PRs addressed to Solana reviewers who support commits.

5.1.1 How to verify Ethernel coin?

You can use the following link to view the generated token information.

```
1 https://solscan.io/token/Bo1hj1f1ms62Cbps9yJxUwCYmsDjXQqjJBq4K9kRKG9w
```

Or go directly to solscan.io and write in the browser the following

```
1 Eternel Coin (ETHN)
```

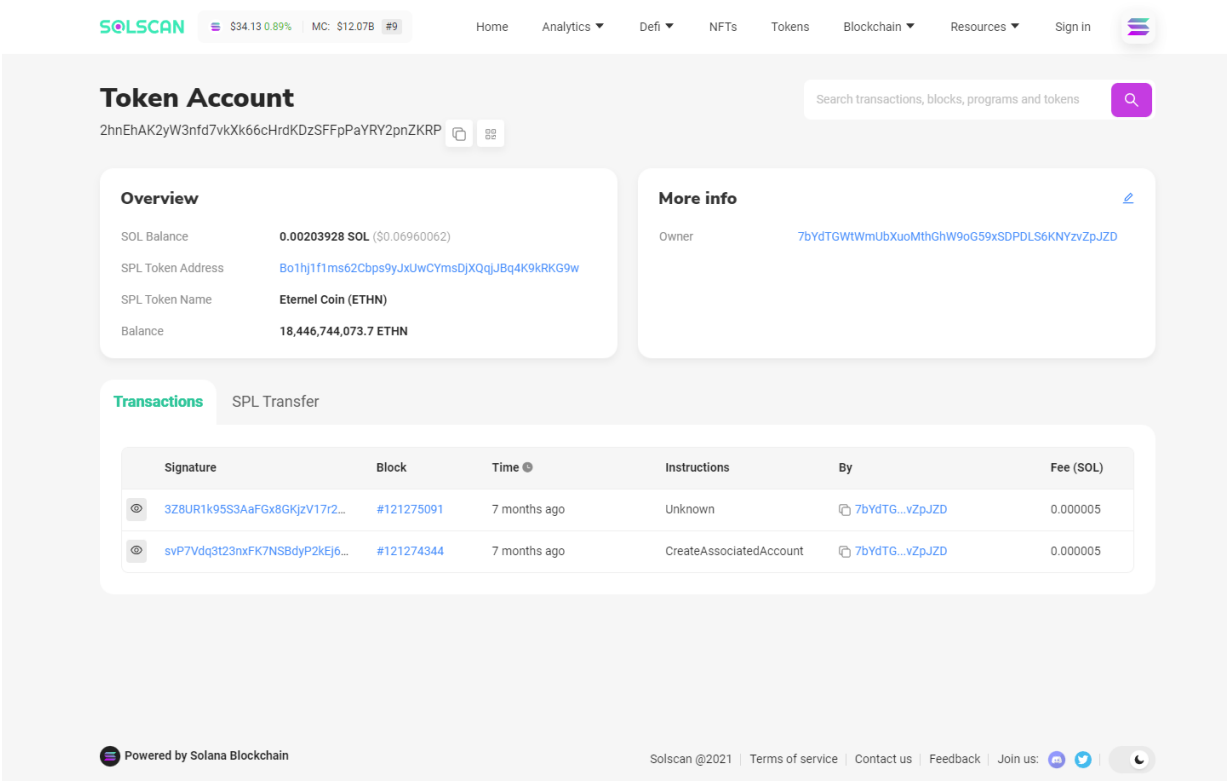

Figure 5.1: Eternel Coin Native Token Creation. Supported and Registered in Solana Blockchain

5.2 Faucet Validation

The transactions of the faucet can be verified by the implementation of the code. The code is provided at the end of this document and looks like this.

In order to visualize the development of the operation of the faucet, the first tests were carried out within the Testnet, which used tokens without value or airdrops, or lamports, and were locally deployed. The process involves the initialization of the Testnet cluster, which is used at the first stage of the process. The Testnet cluster requirements have the same structure as the other clusters, so you should only point to development when the transaction tests are complete and, at the same time, have validated the Transaction made with another wallet. At this stage, the aim is to solve any error that could interrupt the transactions. As evidence of this process, we can see the following repository, which

contains the request structure and instructions.

```
//src\helpers\wallet.ts ...
const cluster = "http://devnet.solana.com";
// const cluster = "http://tesnet.solana.com";
const connection = new Connection(cluster, "confirmed");
const wallet: WalletAdapter = new Wallet("https://www.sollet.io",
    cluster);
...
```

The code in the preceding paragraph constructs a Transaction using a Transaction-Instruction received from SystemProgram, then sends it over the network. You specify the Solana network, such as mainnet-beta, testnet, or devnet, you are connecting to using Connection.

As each Transaction must have a sender and a receiver, we are going to make the connection with a specific wallet that is going to be the sender so that we can carry out the transactions from the faucet. We can use different wallets to be able to review the transactions correctly. Here is the prompt of the wallet when the connection is required to be accepted.

It is required to choose the respective cluster to start the process in the faucet, click over the Solana logo and choose the respective cluster that is configured on wallet.ts.

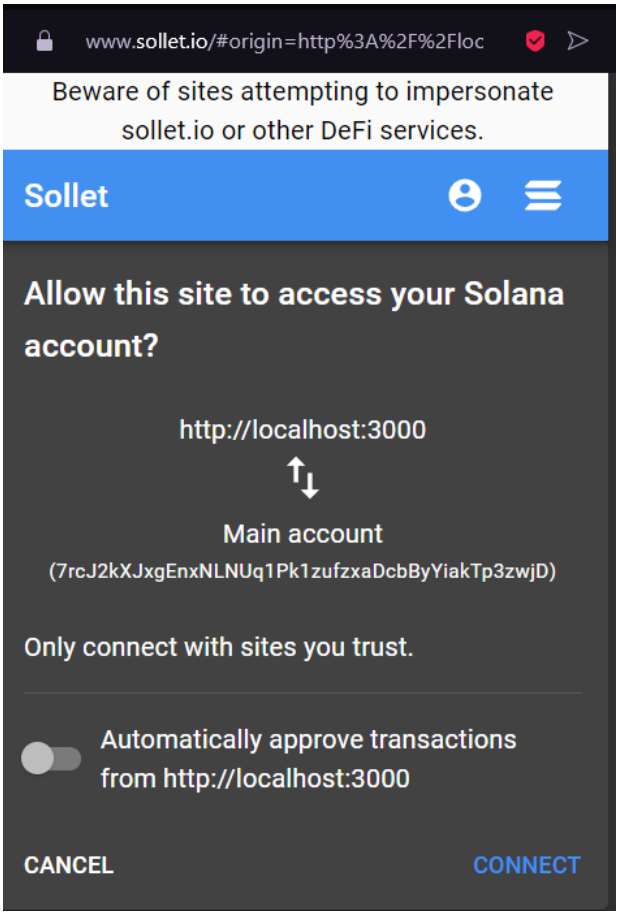

Figure 5.2: Promt of the wallet requierements

Chapter 6

Conclusions

There is no doubt that at the moment blockchain technology is in the initial phase of its development. However, this technological phenomenon deserves full attention and a thorough analysis, since, gradually transforming the structure finance of our institutions will begin to influence all spheres of life in our society, including economy and politics, both in Latin America and throughout the world.

On the blockchain, there are a large number of companies emerging, a fact that is corroborated by the exposure of some of them in this tesis. Both the factor of cryptocurrency, despite the crisis it is going through today, such as the own blockchain technology could improve and optimize notably the economic situation in the region, reviving the financial sphere through the inclusion in it of a increasing number of people without access to Bank System o this technology. The results of the development provide different tools to motivate the adoption of the blockchain technology and how can increase the emerging economy.

- The bibliographic research of this project was successfully carried out, obtaining contents such as repositories, bibliographic sources, and specific documentation that detailed the use of clusters, functions, and specifications that the Blockchain required.
- How to operate a specific blockchain was understood. In this case, it was Solana, where advantages make it suitable for developing the faucet and native tokens and testing.
- A native token called Eternum is created, the steps to follow to replicate it and verify

its existence in the Blockchain network are detailed.

- It was possible to unify the project through the realization of the web faucet, obtaining the development of transactions locally by using Devnet and Testnet of Solana.
- It was possible to perform the validation of transactions by local development. The transactions happened between the linked wallet on the web faucet and other external wallet.
- The faucet can be used on scientific research blockchain infrastructures like the Bloxberg.com version in Solana blockchain. Transactions in the development of the Devnet and Testnet are designed to run multiple executions. The web faucet can be used to realize control tests of transaction, quantity, and execution time. All those tests can be generated within the Solana blockchain network. Configurations can be performed changing a line within the wallet.js file.
- Blockchain faucet code is available as open source Solana smart contract written in JavaScript.
- A decentralized Faucet is contributed with low Solana blockchain gas cost, which enables an unlimited number of users to use it without running into block gas limit exceeding problems.
- An autonomous and fair blockchain faucet for its own crypto-currency distribution is contributed.

Chapter 7

Recommendations

Solana is a very new network that is constantly being built and enhanced. Because some configurations may change over time and the official documentation may not have all the latest information, it may take more time and investigation to uncover fixes and workarounds for problems many people are unaware of. Solana offers some very potent tools. New features were introduced by altering the application's base code, which improved my understanding of the programming underlying the basic UI included in the Metaplex repository.

Chapter 8

Future Works

- The actual output of the thesis is a website that is operational on the local server and deployed on the Devnet and Testnet. Thus the faucet cannot be moved or actually purchased. If the launch date has arrived, users can link their wallets to mint. A countdown shows how much time is remaining until the precise launch date. By moving the web to a distant server and utilizing the Mainnet, the project could be developed further, and the native tokens could be converted to genuine tokens.
- The objective is to enhance the application even more and add new features. Users can link their wallets, but they are unable to disconnect them, therefore that might be added as an example. A part called ”WalletMultiButton” is available in the Solana official repository and can be used for this.

Bibliography

[1] J. Abbate, *Inventing the internet*. MIT press, 2000.

[2] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A brief history of the internet," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 22–31, 2009.

[3] K. Hafner and M. Lyon, *Where wizards stay up late: The origins of the Internet*. Simon and Schuster, 1998.

[4] M. Campbell-Kelly and D. D. Garcia-Swartz, "The history of the internet: the missing narratives," *Journal of Information Technology*, vol. 28, no. 1, pp. 18–33, 2013.

[5] P. Ginsparg, "Kenneth g. wilson: Renormalized after-dinner anecdotes," *Journal of Statistical Physics*, vol. 157, no. 4, pp. 610–624, 2014.

[6] R. Wiggins, "Al gore and the creation of the internet," 2000.

[7] J. Abbate, *Inventing the internet*. MIT press, 2000.

[8] I. Siles González, "A la conquista del mundo en línea: internet como objeto de estudio (1990-2007)," *Comunicación y sociedad*, no. 10, pp. 55–79, 2008.

[9] J. Kurose and K. Ross, "Computer networks: A top down approach featuring the internet," 2010.

[10] J. Abbate, *Inventing the internet*. MIT press, 2000.

[11] B. Obama, "Remarks by the president on securing our nation's cyber infrastructure," *The White House (May 29, 2009), http://www. whitehouse.*

gov/the_press_office/Remarks-by-the-President-on-Securing-Our-Nations-Cyber-Infrastructure, 2009.

[12] C. Decarie, "Facebook: Challenges and opportunities for business communication students," *Business Communication Quarterly*, vol. 73, no. 4, pp. 449–452, 2010.

[13] B. F. Jackson, "Censorship and freedom of expression in the age of facebook," *NML Rev.*, vol. 44, p. 121, 2014.

[14] T. C. Bailey and I. Hsieh-Yee, "Combating the sharing of false information: History, framework, and literacy strategies," *Internet Reference Services Quarterly*, vol. 24, no. 1-2, pp. 9–30, 2020.

[15] C. E. H. Chua and J. Wareham, "Fighting internet auction fraud: An assessment and proposal," *Computer*, vol. 37, no. 10, pp. 31–37, 2004.

[16] J. E. Klobas and L. A. Clyde, "Social influence and internet use," *Library Management*, 2001.

[17] R. DARIGA, "History of development internet," 2000.

[18] M. Kenney, "Cyber-terrorism in a post-stuxnet world," *Orbis*, vol. 59, no. 1, pp. 111–128, 2015.

[19] V. Cerf and B. Aboba, "How the internet came to be," *The On-line User's Encyclopedia: Bulletin Boards and Beyond. Reading, Massachusetts: Addison-Wesley*, 1993.

[20] A. Powell, "Assessing the influence of online activism on internet policy-making: The case of sopa/pipa and acta," *PIPA and ACTA (March 30, 2012)*, 2012.

[21] D. McCullagh, "Vint cerf: Sopa means' unprecedented censorship'of the web," *CBNET News*, 2011.

[22] W. J. Drake, C. G. Vinton, and W. Kleinwächter, "Internet fragmentation: An overview." World Economic Forum, 2016.

[23] R. Kaur and A. Kaur, "Digital signature," in *2012 International Conference on Computing Sciences*. IEEE, 2012, pp. 295–301.

[24] P. Kitsos, N. Sklavos, and O. Koufopavlou, "An efficient implementation of the digital signature algorithm," in *9th International Conference on Electronics, Circuits and Systems*, vol. 3. IEEE, 2002, pp. 1151–1154.

[25] A. Ahmad, M. Saad, L. Njilla, C. Kamhoua, M. Bassiouni, and A. Mohaisen, "Blocktrail: A scalable multichain solution for blockchain-based audit trails," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[26] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. Leung, "Decentralized applications: The blockchain-empowered software system," *IEEE Access*, vol. 6, pp. 53 019–53 033, 2018.

[27] P. L. Fernández, P. L. López, M. A. F. Gámez, and S. M. F. Minguélez, "Identificación de factores de influencia en el precio de las criptomonedas: evidencia para bitcoin y ethereum," *Cuadernos económicos de ICE*, no. 100, pp. 215–233, 2020.

[28] A. Chepurnoy, V. Kharin, and D. Meshkov, "A systematic approach to cryptocurrency fees," in *International Conference on Financial Cryptography and Data Security*. Springer, 2018, pp. 19–30.

[29] M. F. . C. Lattuada, A., "Faucet: A user-level, modular technique for flow control in dataflow engines. proceedings of the 3rd acm sigmod workshop on algorithms and systems for mapreduce and beyond."

[30] J. Kwon, "Tendermint: Consensus without mining," *Draft v. 0.6, fall*, vol. 1, no. 11, 2014.

[31] A. Yakovenko, "Solana: A new architecture for a high performance blockchain v0. 8.13," *Whitepaper*, 2018.

[32] A. Wright and P. De Filippi, "Decentralized blockchain technology and the rise of lex cryptographia," *Available at SSRN 2580664*, 2015.

[33] N. Radziwill, "Blockchain revolution: How the technology behind bitcoin is changing money, business, and the world," *The Quality Management Journal*, vol. 25, no. 1, pp. 64–65, 2018.

[34] L. Baird, M. Harmon, and P. Madsen, "Hedera: A governing council & public hashgraph network," *The trust layer of the internet, whitepaper*, vol. 1, pp. 1–97, 2018.

[35] D. Wörner, T. Von Bomhard, Y.-P. Schreier, and D. Bilgeri, "The bitcoin ecosystem: disruption beyond financial services?" 2016.

[36] S. Nakamoto, "Bitcoin whitepaper," *URL: https://bitcoin. org/bitcoin. pdf-(: 17.07. 2019)*, 2008.

[37] satoshi, "Bitcoin.org is a community funded project, donations are appreciated and used to improve the website."

[38] R. Bdiwi, C. De Runz, S. Faiz, and A. A. Cherif, "Towards a new ubiquitous learning environment based on blockchain technology," in *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*. IEEE, 2017, pp. 101–102.

[39] A. Rejeb, J. G. Keogh, S. J. Simske, T. Stafford, and H. Treiblmaier, "Potentials of blockchain technologies for supply chain collaboration: a conceptual framework," *The International Journal of Logistics Management*, 2021.

[40] J. Li, J. Wu, and L. Chen, "Block-secure: Blockchain based scheme for secure p2p cloud storage," *Information Sciences*, vol. 465, pp. 219–231, 2018.

[41] D. Zheng, C. Jing, R. Guo, S. Gao, and L. Wang, "A traceable blockchain-based access authentication system with privacy preservation in vanets," *IEEE Access*, vol. 7, pp. 117 716–117 726, 2019.

[42] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–34, 2019.

[43] P. De Filippi, M. Mannan, and W. Reijers, "Blockchain as a confidence machine: The problem of trust & challenges of governance," *Technology in Society*, vol. 62, p. 101284, 2020.

[44] N. Alsalami and B. Zhang, "Utilizing public blockchains for censorship-circumvention and iot communication," in *2019 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 2019, pp. 1–7.

[45] M. Kaczorowska, "Blockchain-based land registration: Possibilities and challenges," *Masaryk UJL & Tech.*, vol. 13, p. 339, 2019.

[46] R. M. Díaz, L. Valdés Figueroa, and G. Pérez, "Blockchain implementation opportunities and challenges in the latin american and caribbean logistics sector," 2021.

[47] A. Osuna Garcia, "Bitcoin: claves de su posible adopción como moneda de curso legal en américa latina," 2022.

[48] N. Popper, "Can bitcoin conquer argentina," *The New York Times, April*, vol. 29, 2015.

[49] https://bitso.com, "https://bitso.com," 2023.

[50] https://www.bitinka.com/es/bitinka/home, "https://www.bitinka.com/es/bitinka/home," 2022.

[51] https://www.volabit.com, "www.volabit.com," 2022.

[52] M. Di Pierro, "What is the blockchain?" *Computing in Science & Engineering*, vol. 19, no. 5, pp. 92–95, 2017.

[53] A. Sunyaev, "Distributed ledger technology," in *Internet Computing*. Springer, 2020, pp. 265–299.

[54] H. Natarajan, S. Krause, and H. Gradstein, "Distributed ledger technology and blockchain," 2017.

[55] A. Bharadwaj, "Blockchain and the right to be forgotten," *Nirma ULJ*, vol. 11, p. 35, 2021.

[56] P. De Filippi and S. Hassan, "Blockchain technology as a regulatory technology: From code is law to law is code," *arXiv preprint arXiv:1801.02507*, 2018.

[57] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014.

[58] P. Tasca, "Token-based business models," in *Disrupting finance*. Palgrave Pivot, Cham, 2019, pp. 135–148.

[59] M. Kolvart, M. Poola, and A. Rull, "Smart contracts," in *The Future of Law and etechnologies*. Springer, 2016, pp. 133–147.

[60] A. Papantoniou, "smart contracts in the new era of contract law," *Papantoniou, A.(2020). Smart contracts in the new era of contract law. Digital Law Journal*, vol. 1, no. 4, pp. 8–24, 2020.

[61] D. E. Mik, "Smart contracts: A requiem," *Journal of Contract Law, Forthcoming*, 2019.

[62] S. N. Khan, F. Loukil, C. Ghedira-Guegan, E. Benkhelifa, and A. Bani-Hani, "Blockchain smart contracts: Applications, challenges, and future trends," *Peer-to-peer Networking and Applications*, vol. 14, no. 5, pp. 2901–2925, 2021.

[63] E. Mik, "Smart contracts: terminology, technical limitations and real world complexity," *Law, Innovation and Technology*, vol. 9, no. 2, pp. 269–300, 2017.

[64] F. Hinestrosa, "Codigo civil de bello en colombia, el," *Rev. Derecho Privado*, vol. 10, p. 5, 2006.

[65] F. D. V. Marín, "Panorama actual del bitcóin. una descripción práctica y jurídica de las criptomonedas en colombia y ecuador," *Foro, Revista de Derecho*, no. 36, pp. 49–71, 2021.

[66] M. J. Castro Cobo *et al.*, "Investigación de la legalidad y cumplimiento tributario en la compra y venta de vehículos usados en el ecuador," Master's thesis, Universidad Andina Simón Bolívar, Sede Ecuador, 2014.

[67] A. Egberts, "The oracle problem-an analysis of how blockchain oracles undermine the advantages of decentralized ledger systems," *Available at SSRN 3382343*, 2017.

[68] S. Sung, "A new key protocol design for cryptocurrency wallet," *ICT Express*, vol. 7, no. 3, pp. 316–321, 2021.

[69] M. J. Shayegan, H. R. Sabor, M. Uddin, and C.-L. Chen, "A collective anomaly detection technique to detect crypto wallet frauds on bitcoin network," *Symmetry*, vol. 14, no. 2, p. 328, 2022.

[70] Solana, "https://docs.nft4artists.io/guides/what-is-a-wallet," *Whitepaper*, 2022.

[71] H.-T. Kung and J. T. Robinson, "On optimistic methods for concurrency control," *ACM Transactions on Database Systems (TODS)*, vol. 6, no. 2, pp. 213–226, 1981.

[72] S. S. Chow, Z. Lai, C. Liu, E. Lo, and Y. Zhao, "Sharding blockchain," in *2018 IEEE international conference on internet of things (iThings) and IEEE Green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData)*. IEEE, 2018, pp. 1665–1665.

[73] N. Bodziony, P. Jemioło, K. Kluza, and M. R. Ogiela, "Blockchain-based address alias system," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 16, no. 5, pp. 1280–1296, 2021.

[74] (2022) Transactions. [Online]. Available: https://docs.solana.com/developing/programming-model/transactions

[75] G. A. Pierro and R. Tonelli, "Can solana be the solution to the blockchain scalability problem?" in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2022, pp. 1219–1226.

[76] (2022) Instructions. [Online]. Available: https://github.com/solana-labs/solana/tree/6606590b8132e56dab9e60b3f7d20ba7412a736c

[77] E. Peterfay, "Creating a website for a programmatically generated nft collection: solana network," 2022.