BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION COMPUTER SCIENCE

DIPLOMA THESIS

WorkShare: Empowering Compensation with Blockchain Technology

Supervisor
Conf, Dr. Craciun Mihai Florin Gabriel

Author
Benedek Robert George

2023

ABSTRACT

In the current business environment, companies and employees face the weighty burden of government taxes and contributions, ranging from 35% to 47.25% of the gross salary in Romania [Ana23a], alongside the additional transfer costs imposed by banks. This dual challenge places a substantial financial strain on businesses. Another significant challenge is finding hardworking software developers and scalability, especially when limited to local talent pools.

The application proposed in this thesis titled WorkShare, is a groundbreaking solution that leverages blockchain technology to address multiple challenges in project management and compensation. It offers a solution using the power of blockchain technology, smart contracts and Polygon network . WorkShare minimizes transaction fees to nearly zero, optimizes payment processes, and significantly reduces tax and contributions obligations on earnings by up to 78% taking advantage of the 10% tax in Romania for earnings that come from cryptocurrencies [Ana23b] and even by 100% in countries like Belarus, where the earnings from cryptocurrencies are not taxed. WorkShare offers an economical alternative that brings significant relief to companies, empowering them to optimize resource allocation and drive growth in the digital economy. Furthermore, it provides additional incentives for developers and introduces a unique NFT proof system that validates their skills and project accomplishments.

Additionally, the application addresses scalability concerns faced by companies by connecting companies with a pool of skilled developers from around the world, WorkShare ensures a consistent workflow and project availability, regardless of the company's current project load.

With WorkShare, companies can achieve greater efficiency in resource management, foster a thriving digital economy, and reward developers for their contributions through the innovative NFT proof mechanism.

This work is the result of my own activity. I have neither given nor received unauthorized assistance on this thesis.

Contents

Chapter 1

Introduction

Money has been an essential part of human life since the beginning of civilization, enabling the exchange of goods and services. Over time, money has evolved from tangible objects with intrinsic value, such as gold or silver coins, to intangible assets that can exist solely in digital form. As the world has become increasingly digitized, so has the concept of money. Throughout history, money has taken various forms 1.1, from shells and beads to coins made of precious metals. The introduction of paper money in China during the Tang Dynasty marked a significant milestone in the history of money. Paper money quickly spread across the world,and governments began to regulate its production and use.

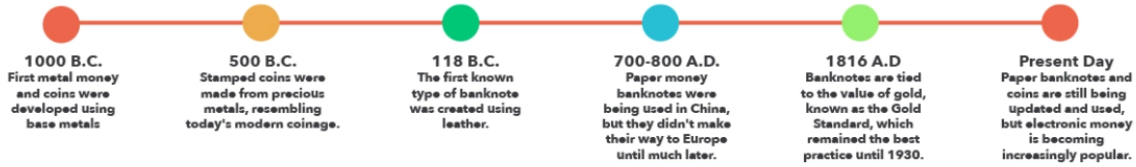

Figure 1.1: https://mint.intuit.com/blog/investments/the-history-of-money/

With the advent of modern technology, money has become increasingly intangible, and physical currency is being replaced by digital transactions. Today, money is often represented as numbers on a screen, easily controlled and multiplied by banks. Today, we might be witnessing a new shift in the way we perceive and use currency. This shift has led to the development of digital cryptocurrencies like Bitcoin. As the history tends to repeat it is possible that in the near future we are going to see digital cryptocurrencies that appear as a global currency and bitcoin as a store of value as it has all the characteristics of money: durability, portability, divisibility, fungibility, scarcity, acceptability. Bitcoin combines the hardness of gold with the portability and fungibility of fiat and comes built for the digital age. Its supply is regulated by its code and enforced by those who use it and can be sent anywhere in the world in seconds, anytime, without incurring the prohibitive costs so often

charged in the traditional financial system. Some countries around the globe are starting to adopt Bitcoin as legal tender, such as El Salvador and the Central African Republic. [Bro22]

Bitcoin have introduced to the world the blockchain technology which has been adopted now by some of the biggest IT companies around the world : IBM (IBM), NVIDIA (NVDA), PayPal (PYPL), Coinbase (COIN), Riot Blockchain (RIOT). In a few years a lot of the web2 platforms are going to be shifted toward web3, maybe even how people are going to be paid, the formal and legal contracts that we sign now might be replaced by smart contracts and the payments in fiat replaced by cryptocurrencies payments.

With the increasing adoption of blockchain technology, I have chosen to explore its potential in addressing a common challenge faced by software companies - scaling their workforce for large projects or during periods of decreased demand. One of the primary obstacles to scalability is the high cost of maintaining a permanent workforce, including the significant taxes that employers must pay for their employees, which can be as high as 48% in Romania. However, by using cryptocurrencies to pay employees, the tax burden shifts to the employees themselves, who are only required to pay taxes when they sell their cryptocurrencies. This could result in a significant reduction in taxes, with rates as low as 0% in some countries. However, in Romania, the tax rate is 10%. As such, my thesis aims to investigate the feasibility and potential benefits of using blockchain-based payment systems to enable more flexible and cost-effective employment practices.

In today's world, one of the biggest problems faced by companies is the high cost of payment transactions, especially when sending money across borders. Traditional payment systems such as banks charge high fees, which significantly affect the profitability of businesses. Moreover, sending payments during weekends or holidays is often impossible, further delaying the completion of projects and resulting in additional costs.

To solve these challenges, I have developed an innovative application that uses polygon blockchain technology to enable fast, secure, and cost-effective transactions. Our app connects companies and individuals looking for skilled developers from around the globe to complete their projects with experienced developers searching for work opportunities.

By using blockchain technology, we eliminate the need for intermediaries such as banks, thereby reducing the costs of transactions significantly. The app provides a seamless experience for companies and individuals to find developers with the necessary skills to complete their projects while ensuring the developers are paid promptly and transparently.

Furthermore, WorkShare app leverages the potential of blockchain technology to

enable payments through digital currencies, which are increasingly gaining traction worldwide. With China introducing its digital yuan [Mac22] and the US investigating CBDCs (central bank digital currency) [gov23], we believe that the future of payments lies in blockchain technology. The app is well-positioned to take advantage of this trend by enabling secure and fast transactions on the blockchain.

In summary, WorkShare is an innovative solution that aims to address the high costs and inefficiencies of traditional payment systems while leveraging the potential of blockchain technology to enable fast, secure, and transparent transactions. I believe that the app will transform the way payments between companies and someone that provides a se rvice are made, benefiting both companies and developers alike.

Chapter 2

Blockchain

A blockchain is a distributed, decentralized database that enables multiple parties to store and access data without the need for a centralized authority. It is essentially a growing ledger (chain) of digital records that is protected by cryptographic techniques. It contains a number of transactions that are verified and then added to the chain in a linear, chronological order. Once a block is added to the chain, it cannot be changed without affecting all subsequent blocks, making tampering with the information stored in the blockchain nearly impossible 2.

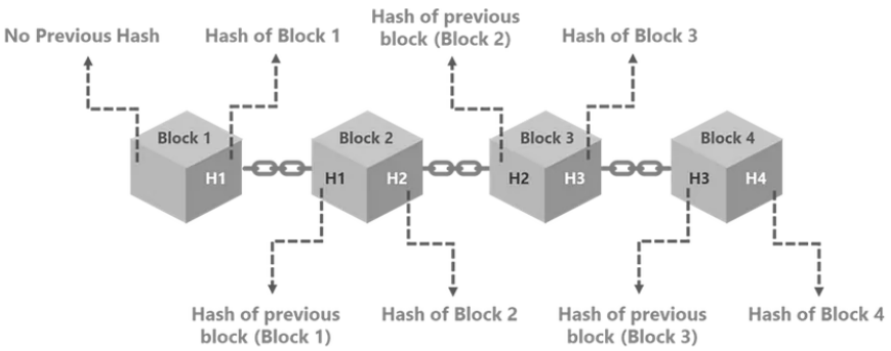

Figure 2.1: Hash of the previous block links the block to another block in the Blockchain [Bre22]

Each participant in a blockchain keeps a copy of the database, which is synchronized with the other participants via a consensus protocol. Consensus protocols ensure that all participants agree on the current state of the database and prevent malicious actors from altering the ledger without permission. [Bre22]

Most people associate blockchain technology with cryptocurrencies, which use blockchains to record and verify transactions. However, blockchain has many potential applications that are not limited to cryptocurrencies. It can, for example, be used to track asset ownership and transfer, manage supply chain logistics, and enable secure online voting systems, among other things.

2.1 Distributed ledger technology

Distributed ledger technology (DLT) is a method of storing data that allows multiple parties to share and synchronize information across a network of computers without the need for a central authority. In simpler terms, it's a way of maintaining an honest and open record of transactions or data.

The most widely recognized example of DLT is blockchain, a decentralized and distributed ledger that stores transaction records permanently and sequentially. Each node, or computer, in the network maintains a copy of the ledger, and each transaction is validated and recorded by a network of nodes through a consensus mechanism.

DLT has many advantages over traditional database systems, including increased transparency, improved security, and lower costs. By removing intermediaries or central authorities, DLT can lower transaction costs and improve the speed and efficiency of transactions. Furthermore, DLT can be used to create new business models and build trust in complex networks of stakeholders. [FRA23]

2.2 Cryptography

Cryptography is the practice of securing information by converting it into a code or cipher making it difficult for unauthorized parties to read or access. It involves two important functions - encryption and decryption.

Encryption is the process of transforming the original information or plaintext into a series of random numbers that appear meaningless (ciphertext). Decryption, on the other hand, is the reverse process of converting ciphertext back into plaintext.

There are several types of cryptography, including symmetric key cryptography, public key cryptography, and hash algorithms. Symmetric-key cryptography uses the same key for encryption and decryption, while public-key cryptography uses two different keys. Hashing algorithms, such as SHA-256, generate a fixed-length digital fingerprint of data that ensures its integrity and authenticity.

In the blockchain, cryptographic hashing is critical to the security and immutability of the ledger. Each block in the chain contains a hash of the previous block, making it difficult for anyone to alter or manipulate the data without being detected. In addition, cryptographic hashing is used in the generation of wallet addresses to ensure the privacy and security of users transactions 2.2.

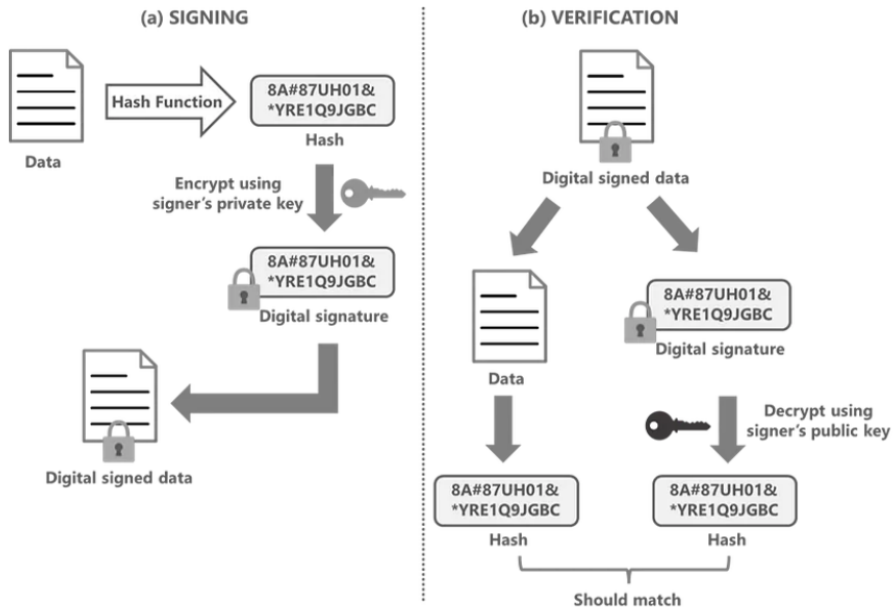

Figure 2.2: Encrypting information with (a) receiver B's public key and (b) sender A's private key [Bre22]

2.3 Network nodes

The role of Blockchain nodes is to maintain the distributed ledger and act as communication hubs for various network tasks. They are authorized stakeholders who confirm the legality of subsequent batches of transactions, known as blocks. Each node is allocated a unique identifier to distinguish it from other nodes. In a Proof-of-Work Blockchain like Bitcoin or Monero, miners are responsible for maintaining the network. Full nodes are required to store all Blockchain transactions and validate blocks and transactions, while lightweight nodes only need to download block headers to verify transactions. It's worth noting that not every version of a full node includes a block reward.[cou22]

2.4 Consensus protocol

When all members of a business network agree on a consensus protocol, it guarantees that only transactions verified by the network are recorded in the ledger. This consensus mechanism also ensures that all shared ledgers are precise duplicates, reducing the possibility of fraudulent activities, as any tampering attempt would have to occur simultaneously across multiple places. Additionally, the use of cryptographic hashes, such as the SHA256 algorithm, ensures that even a slight modification to the transaction input would generate a distinct hash value, signaling a potentially compromised transaction. Furthermore, digital signatures guarantee

that transactions are authentic and originated from legitimate senders who signed them with private keys, preventing impersonation. [SB18]

2.5 Blockchain types

There are four types of blockchain: public, permissioned or private, hybrid and federated or consortium.2.3

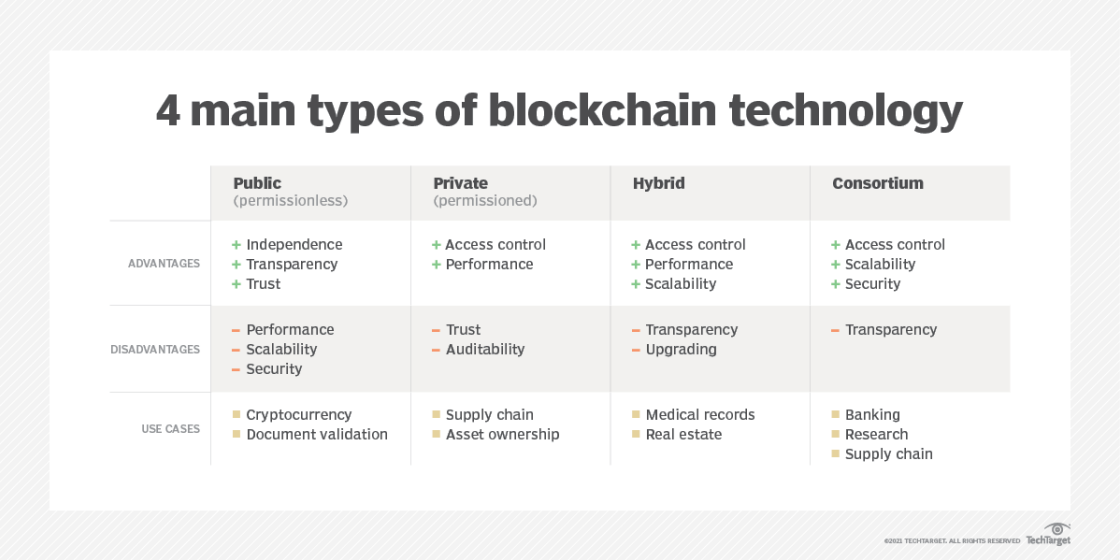

Figure 2.3: The differences between blockchain types [Cam23]

2.5.1 Public blockchain

A public blockchain is a form of distributed ledger system that works with no constraints or limitations on who can participate. Bitcoin and Ethereum are two of the most well-known examples of public blockchains. Transactions in these networks are validated and recorded by a decentralized network of nodes that conform to a set of agreed-upon rules or consensus algorithms. The most important advantage of public blockchains is their transparency and security. Because there is no centralized authority operating the network, all participants have equal access to the ledger and may check transaction legitimacy. As a result, public blockchains are resistant to censorship or manipulation by any single entity. Yet, the open nature of public blockchains makes them vulnerable to attacks and necessitate enormous computational resources to maintain if the consensus mechanism is proof of work (Pow) , lately there exists better options.

2.5.2 Private blockchain

Private blockchains, on the other hand, are restricted networks which can only be joined and used with permission. They are frequently used by enterprises that need to share sensitive information or control who has access to the network. Private blockchains are more secure than public blockchains due to access constraints and often have faster transaction processing times. Big IT companies like IBM have started to use this technology, they developed Hyperledger Fabric : is a permissioned blockchain designed for enterprise use cases.

2.5.3 Hybrid blockchain

Hybrid blockchains are a combination of public and private blockchains. They offer the benefits of both types, including decentralized consensus mechanisms and access controls for restricted users. Hybrid blockchains are often used by organizations that need to balance the benefits of openness with the need for privacy and security. Transactions and records in a hybrid blockchain are typically not made public but can be verified when necessary, for as by granting access via a smart contract. Secret data is retained within the network but can still be verified.Altering or changing transactions are impossible even if a private entity owns the hybrid blockchain.Two examples would be Binance Chain: A blockchain developed by cryptocurrency exchange Binance and "Food Trust" from IBM which was developed to improve efficiency throughout the whole food supply chain.

2.5.4 Consortium blockchain

A consortium blockchain is a type of blockchain where multiple organizations come together to form a shared network, which is jointly owned and operated by the members. In this model, the participating organizations have a say in the consensus process and can control who has access to the data on the blockchain. Consortium blockchains are often used in industries such as finance, supply chain management, and healthcare, where multiple parties need to collaborate and share data securely and efficiently. Compared to public blockchains, consortium blockchains offer greater privacy and scalability while still allowing for a degree of decentralization. Two notable examples: Ripple, which is a payment protocol and network that uses a consortium blockchain to facilitate cross-border payments between financial institutions. It is designed to be fast and efficient, with transaction settlement times of just a few seconds and VeChain, a supply chain management platform that uses a consortium blockchain to track and verify the authenticity of products as they move through the supply chain.

Chapter 3

Ethereum and Smart Contracts

3.1 Ethereum blockchain and its capabilities

Ethereum is a blockchain platform that allows the development of decentralized apps (dApps) and smart contracts. Because of its adaptability and robust features, it was established in 2015 and has since become one of the most popular blockchain platforms for developers.

The ability to construct and execute smart contracts is one of Ethereum's most important features. These are self-executing contracts in which the provisions of the agreement are encoded directly into code. They are used to automate transaction execution and eliminate the need for intermediaries, lowering transaction costs and increasing efficiency. Ethereum also has its cryptocurrency called Ether (ETH), which is used to pay for transaction fees and execute smart contracts. In addition, Ethereum provides developers with the ability to create their tokens and initial coin offerings (ICOs) on the platform, which has led to the creation of many new blockchain-based projects and applications. [tea19]

Ethereum's ability to design and deploy decentralized apps (dApps) on the blockchain is another distinguishing characteristic. These decentralized applications can range from simple games to major financial applications and are constructed with smart contracts and other blockchain technologies.

Furthermore, Ethereum is always evolving and strengthening its technology through upgrades and changes. For example, Ethereum is now shifting from a proof-of-work (PoW) consensus algorithm to a proof-of-stake (PoS) consensus algorithm in order to reduce energy consumption and boost efficiency. 3.1

Ultimately, Ethereum's capabilities as a blockchain platform have opened up a world of possibilities for developers, entrepreneurs, and enterprises to create innovative, secure, and trustless decentralized apps.

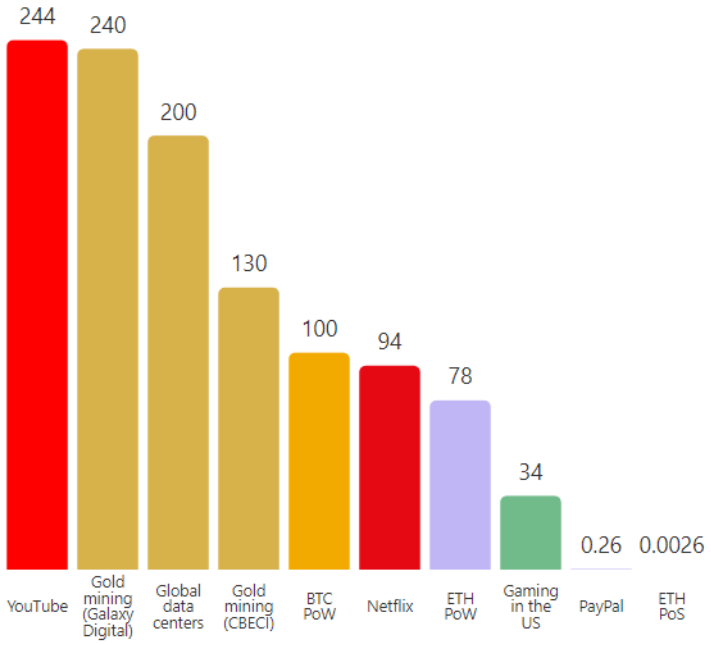

Figure 3.1: Annual Energy Consumption in TWh/yr [tea19]

3.1.1 Gas and its role in transactions

"Gas refers to the unit that measures the amount of computational effort required to execute specific operations on the Ethereum network." [Et23]

Gas is a fundamental component of Ethereum's blockchain network that supports transaction efficiency and security. It is the charge that users pay to execute a transaction or smart contract on the Ethereum network, each transaction and smart contract execution in Ethereum consumes a specific amount of computational resources, such as computing power and storage. Because these resources are limited and hence scarce, the network must allocate them efficiently to preserve the system's stability and security.

The amount of gas required for a transaction is determined by the operation's complexity as well as current network factors such as the number of pending transactions and available resources. Users must pay for gas in Ether, the Ethereum network's native coin. The price of gas is denominated in Ether, and users can indicate how much Ether they are ready to spend per unit of gas.

3.1.2 Scaling - layer 2

Most Layer 2 solutions revolve around a server or a cluster of servers, commonly referred to as nodes, validators, operators, sequencers, block producers or similar terms. The operation of these Layer 2 nodes may be carried out by individuals, businesses or entities that use them, a third-party operator or a large group of in-

dividuals akin to Mainnet. Typically, transactions are submitted to these Layer 2 nodes as opposed to directly to Layer 1 (Mainnet). For certain solutions, the Layer 2 instance batches transactions into groups before anchoring them to Layer 1, where they are secured and immutable. However, the mechanisms and procedures involved in this process vary widely across different Layer 2 technologies and implementations. [cor23]

Layer 2 is needed because it greatly improves the transactions per second, making the user experience smoother and reducing network congestion on Mainnet Ethereum. Transactions are bundled together into a single transaction to Mainnet Ethereum, which reduces gas fees for users and makes Ethereum more accessible. It is important to note that updates to scalability should not compromise decentralization or security, and that layer 2 networks build on top of Ethereum. Additionally, application specific layer 2 networks bring their own efficiencies when working with assets at scale.(Polygon 3.4, Arbitrum, Optimism)

3.2 Smart contracts

Smart contracts are computer programs that are activated when a user initiates a transaction on the Ethereum blockchain. They set Ethereum apart from other cryptocurrencies by allowing it to execute a wide range of operations, making it extremely versatile. Decentralized apps, or dapps, are another name for these programs.

Smart contracts have a number of advantages over standard contracts. A smart contract, unlike conventional goods that may change conditions or remove features, is permanently published on Ethereum and will stay operational as long as Ethereum exists. Even the author is unable to remove it. Furthermore, because smart contracts are automated, they treat all users equally and are always accessible. [tea19]

3.2.1 Solidity, the programming language used to write smart contracts

Solidity is an object-oriented, high-level language for implementing smart contracts being. It is a curly-bracket language designed to target the Ethereum Virtual Machine (EVM) being influenced by C++, Python and JavaScript. Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features. Breaking changes as well as new features are introduced regularly, it currently use a 0.y.z version number to indicate the fast pace of change.[eth16] Some

of the last changes introduced in Solidity version 0.8.0 are : arithmetic operations revert on underflow and overflow even if it comes at a slight increase of gas costs, ABI coder v2 is activated by default, address literals have the type address instead of address payable. [Aut16]

3.2.2 Upgradable contracts

The immutability of blockchains, including the data stored on the Ethereum blockchain and its smart contracts, is a key feature of blockchain technology. However, there are instances when upgrading smart contracts becomes necessary. This can be due to various reasons, such as fixing bugs, improving functionality, optimizing code to reduce Ethereum gas usage, adapting to technological, market, or societal changes, or avoiding the need to migrate users to a new application version. Upgrading smart contracts requires careful consideration and planning to ensure the security and integrity of the blockchain network.

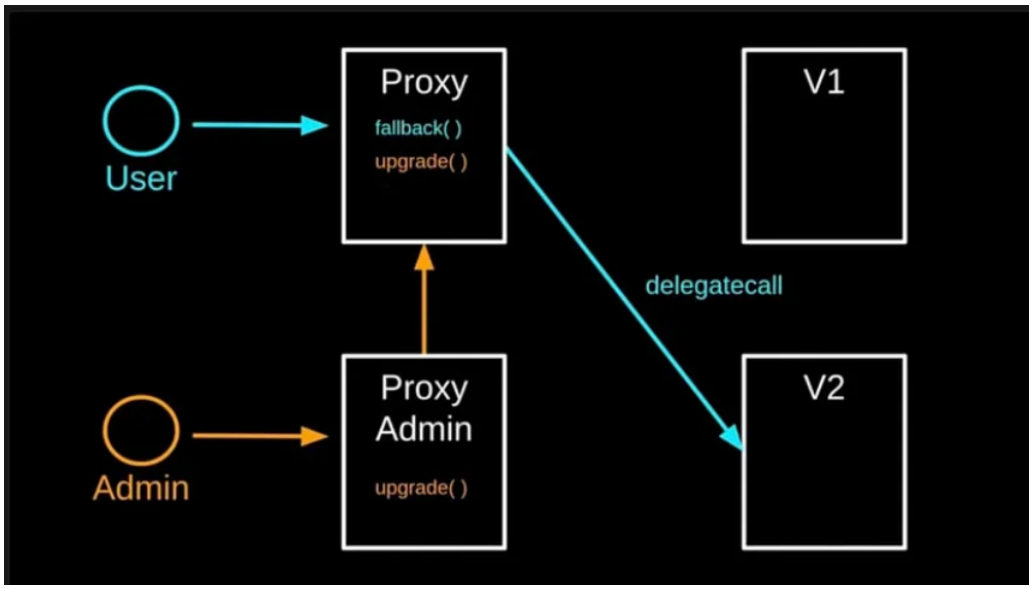

Figure 3.2: The most popular is called UUPS proxy. (or ERC1967 proxy) [Tru22]

In blockchain technology, smart contracts are considered immutable once they are deployed. However, a dApp can consist of multiple smart contracts that work together to create its ”backend”. This offers the opportunity to upgrade the interaction pattern between these smart contracts. Contrary to popular belief, upgrading a smart contract does not involve modifying its deployed code. Rather, it involves replacing the existing smart contract with a new one. This is usually done in a manner that ensures the end user does not need to alter their interaction with the dApp. Therefore, upgrading smart contracts is essentially a process of replacing old, immutable smart contracts with new ones, leaving the old ones abandoned on the

chain.[Pra22]

To upgrade smart contracts, the ”Proxy Pattern” 3.2 software architecture is commonly used. It functions as an intermediary, directing incoming traffic from the frontend to the appropriate smart contract on the backend. The proxy is itself a smart contract keeping all the data and possessing an unchanging Ethereum contract address, allowing for the swapping of other smart contracts in the system. When a new smart contract is deployed, the proxy contract is updated with its correct address. This approach allows end users to interact with the proxy directly, while accessing the other smart contracts indirectly through the proxy.

3.3 OpenZeppelin library

”OpenZeppelin is a “battle-tested” open source framework comprised of reusable Ethereum smart contracts.”[Cre18]

OpenZeppelin is a framework designed to help smart contract developers create distributed applications (dapps) with reduced vulnerability risk. It does this by providing a library of standard, tested, and community-reviewed code. With over 4.5 billion dollars worth of digital assets powered by its smart contracts, OpenZeppelin has proven to be a reliable and effective tool for developing secure dApps.

The principles and strategies that form the foundation of OpenZeppelin’s development approach are focused on security, simplicity, modularity, clarity, testing, consistency, and auditing. These principles are implemented in the form of security in depth, simple and modular code, clarity-driven naming conventions, comprehensive unit testing, pre-and-post-condition sanity checks, code consistency, and regular audits. [Cre18]

One of the benefits of using OpenZeppelin is its coherent architecture, which allows developers and teams to easily organize base contracts, libraries, and interfaces that can be extended and used for custom implementation. It provides developers with a powerful set of tools for building secure and reliable smart contracts, while also encouraging best practices.

3.3.1 ERC20 - Token

ERC-20 is a widely used smart contract standard on the Ethereum blockchain for creating fungible tokens. It defines a set of rules that all tokens should follow, allowing developers to accurately predict how new tokens will function within the system. This simplifies the development process, as new projects won’t need to be redone each time a new token is released, as long as the token adheres to the

standard. The ERC-20 interface outlines the required functions that a token must implement to be considered an ERC-20 token.[tea20b]

The methods of a ERC-20 interface: [tea20b]

```
function totalSupply() external view returns (uint256);
function balanceOf(address account)
    external view returns (uint256);
function allowance(address owner, address spender)
    external view returns (uint256);

function transfer(address recipient, uint256 amount)
    external returns (bool);
function approve(address spender, uint256 amount)
    external returns (bool);
function transferFrom(address sender, address recipient,
uint256 amount) external returns (bool);
```

3.3.2 ERC721 - Non Fungible Token

ERC-721 is a non-fungible token (NFT) standard that is used to uniquely identify something or someone. These tokens are suitable for platforms that sell collectibles, access keys, lottery tickets, numbered seats at concerts and sporting events, and other services. Unlike standard tokens, NFTs have unique features and prices that might vary depending on criteria such as age, rarity, or look.[tea20a]

It establishes a standard for non-fungible tokens, allowing developers to build tokens with distinct attributes and values. This standard assures that each token is unique and can be identified and traded separately. NFTs are a breakthrough in blockchain technology that has created new opportunities for inventors, collectors, and traders.

The methods of a ERC-721 interface [tea20a]

```
 function balanceOf(address _owner)
 external view returns (uint256);
function ownerOf(uint256 _tokenId)
external view returns (address);
function safeTransferFrom(address _from, address _to,
uint256 _tokenId, bytes data) external payable;
function safeTransferFrom(address _from, address _to,
uint256 _tokenId) external payable;
function transferFrom(address _from, address _to,
```

```
uint256 _tokenId) external payable;
function approve(address _approved, uint256 _tokenId)
external payable;
function setApprovalForAll(address _operator, bool _approved)
external;
function getApproved(uint256 _tokenId)
external view returns (address);
function isApprovedForAll(address _owner, address _operator)
external view returns (bool);
```

3.3.3 Ownable Contract: Secure Ownership and Access Control

"Access control—that is, "who is allowed to do this thing"—is incredibly important in the world of smart contracts." [doc23]

The Ownable contract is a standard contract implementation provided by the OpenZeppelin library in the Ethereum ecosystem. It provides a set of functions and modifiers that enable secure ownership management. It includes a constructor that sets the initial owner of the contract and grants them exclusive administrative privileges. The contract owner has the authority to perform certain actions, such as transferring ownership to another address or renouncing ownership altogether.

One of the key features of the Ownable contract is the "onlyOwner" modifier, which can be applied to functions to restrict access to the contract's owner. By using this modifier, developers can ensure that only the owner is allowed to execute specific functions, providing an additional layer of access control and security.

3.4 The Polygon Network: Fast Transactions with Almost Zero Fees

In the world of blockchain technology, scalability and transaction costs have always been significant challenges. As the popularity of decentralized applications (DApps) continues to grow, the Ethereum network, one of the most widely used blockchain platforms, has faced limitations in terms of transaction speed and high fees. However, the emergence of the Polygon network has brought a revolutionary solution to these challenges. This chapter will explore the Polygon network and its remarkable features, particularly its ability to facilitate fast transactions with almost zero fees.

3.4.1 Understanding the Polygon Network

The Polygon network, formerly known as the Matic Network, is a Layer 2 scaling solution built on top of the Ethereum blockchain. It aims to address the scalability issues of Ethereum by providing a framework for developing and deploying scalable DApps. Polygon achieves this by utilizing a combination of technologies, including sidechains, Plasma chains, and state channels. 3.3

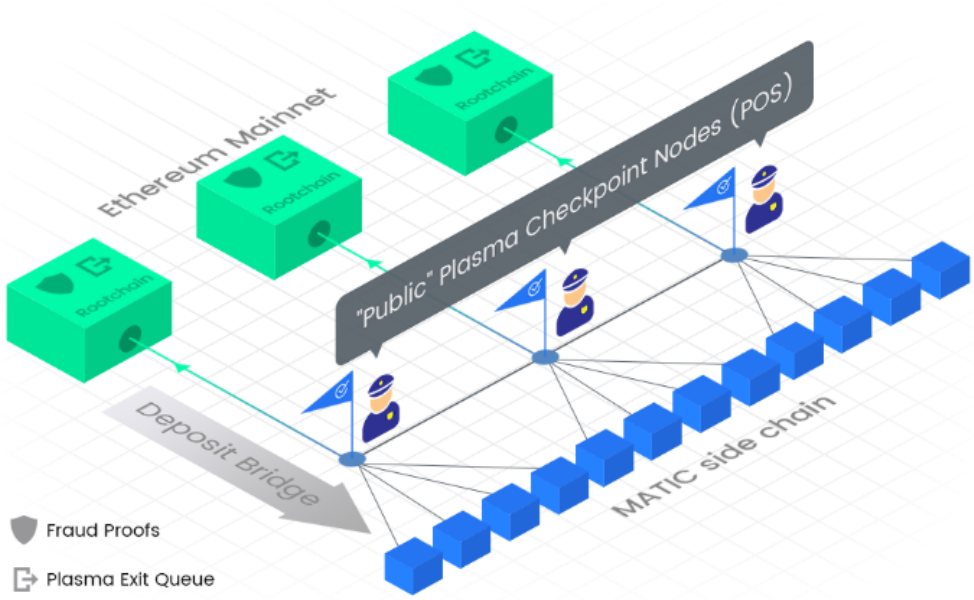

Figure 3.3: Polygon PoS Architecture [Sal23]

3.4.2 Fast Transactions

One of the standout features of the Polygon network is its ability to facilitate fast transactions. Unlike the Ethereum mainnet, which often experiences network congestion and slower transaction confirmations, Polygon offers significantly higher throughput and faster block confirmations. The network has an average block processing time of 2.3 seconds compared to Ethereum's 13.2. [Tea22] This means that users can enjoy near-instantaneous transaction processing, enabling a seamless and efficient user experience.

3.4.3 Almost Zero Fees

Another significant advantage of the Polygon network is its near-zero transaction fees. Ethereum's rising gas fees have become a barrier to entry for many users and developers, especially for applications that require frequent transactions or microtransactions.To cut gas costs, scaling solutions like Polygon process transactions on side chains. Polygon has the potential to handle up to 65,000 transactions

per second, whereas Ethereum can process only up to roughly 17 transactions per second.[Tea22] However, Polygon eliminates this hurdle by offering incredibly low fees and fast transactions, making it economically viable for a wide range of use cases.

Chapter 4

Proposed approach for Company management with freelancers

This chapter presents the WorkShare application's innovative strategy, leveraging smart contracts for efficient project management and collaboration. It outlines the user journey, benefits of smart contract integration, and key functionalities. This approach enables companies to streamline project selection, developer assessment, communication, and evaluation, optimizing resource allocation and accessing a global talent pool. WorkShare empowers companies and developers with transparent transactions, automated payments, and a fair compensation system, revolutionizing company-freelancer dynamics in the digital economy.

4.1 Regular company issues

Developing software projects can be complicated, with large teams and constantly evolving requirements. Managing this complexity poses a significant challenge for software development companies. Project managers must balance the project's needs, allocate tasks to developers according to their experience levels, and ensure timely completion within budget and quality standards. Furthermore, motivating team members to be productive can be difficult when they know they will receive a fixed salary regardless of their performance. This can lead to complacency and lack of motivation to exceed expectations.

Software development is a highly competitive field, and companies can face significant challenges when it comes to attracting and retaining talented developers. Given the high demand for their skills, developers have plenty of options available to them, meaning that companies must offer attractive salaries, benefits, and working conditions to attract and retain the best talent, even if there isn't a particularly important project to work on at the time. In addition, developers often prefer to

work on projects that are both innovative and challenging, and that align with their personal values and interests. This can make it tough for companies to keep their developers engaged when they have specific projects that need to be completed, which may not always match up with the developer's interests.

Companies must also be able to accurately forecast revenue and manage cash flow to ensure they have the resources to invest in new projects and technologies.

4.2 Current approach

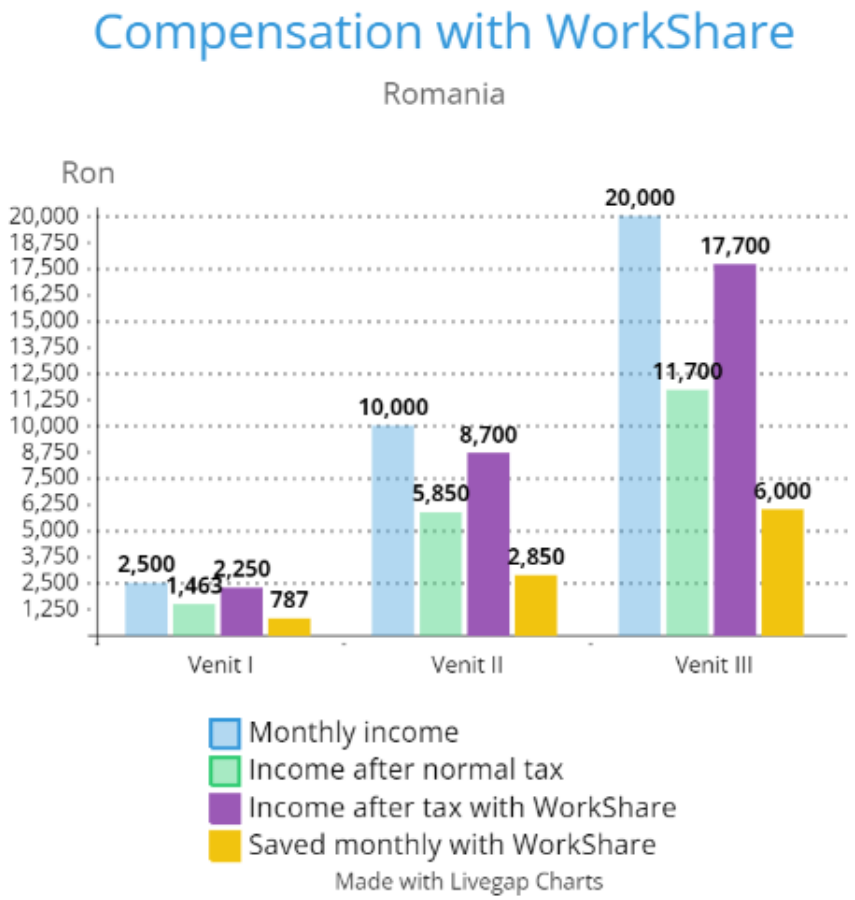

Figure 4.1: Tax Reduction Chart which illustrates the approximate earnings, without factoring in transfer costs or exchange fees.

The app's goal is to help companies become more scalable by providing access to talented developers who are confident in their skills and willing to take full responsibility for their projects and to significantly reduce taxes with the help of cryptocurrencies earnings 4.1. These developers are aware that they may not receive payment if their work is inadequate or may be penalized if they miss the deadline. This approach motivates the developers to work hard and complete their projects

efficiently. As a reward, developers who deliver high-quality work within the deadline receive a NFT that represents the proof of certain skills.

The app uses smart contracts to ensure that every transaction is transparent and recorded on the blockchain, enabling real-time monitoring for all involved parties, including developers and managers and it simplifies the hiring and onboarding process, making it easier for companies to scale their teams rapidly and efficiently based on the needs they have for projects. Developers can choose the projects that interest them, and managers can assign tasks based on experience levels that are easily verifiable through the blockchain. The developers are going to be paid through the smart contract after they finish the project and gets verified by the manager, if it has been done before deadline he will receive the full amount in WorkShareToken, which is the internal token of the application, otherwise a penalty that is set when the project is created is going to be applied.

4.3 Advantages of running the app on a smart contract

Smart contracts play a vital role in decentralized applications, serving as a fundamental component that enables the automated execution of predetermined tasks according to established business rules. By incorporating smart contracts into financial transactions, numerous benefits can be realized, primarily centered around increased efficiency and reduced reliance on intermediaries like asset servicers and custodians because the funds for the project are kept inside the contract, and released only after predetermined conditions have been met. Also as smart contracts are based on blockchain, they guarantee the immutability of data so it offers transparency and the progress of a developer can be checked by the managers that want to hire them.

However, it is crucial to highlight the positive impact that smart contracts can have on financial transactions. By embracing these innovative tools, businesses can unlock efficiency gains, less taxes, reduce reliance on intermediaries, and enhance the trustworthiness of their operations. Through continued advancements in technology and the establishment of clear legal and regulatory frameworks, the adoption of smart contracts in financial markets can be further accelerated, enabling organizations to harness their full potential for transformative change. [Sin22]

Additionally, I have taken proactive measures to optimize gas usage within the application. By carefully architecting smart contracts and implementing efficient transaction management strategies, they have minimized the gas fees required for various operations, further enhancing the user experience and reducing the financial burden on users.

These efforts to address the challenges associated with high gas fees demonstrate

my commitment to the WorkShare project to deliver a practical and user-friendly application on the Polygon network. By leveraging the advantages of blockchain technology while overcoming its limitations, WorkShare aims to provide a seamless and cost-effective experience for both companies and developers, facilitating efficient project management and encourage a vibrant community of talent exchange.

4.4 Challenges

Blockchain technology is still relatively new and may take a few more years to be widely adopted, finding developers that are willing to be paid in cryptocurrencies might be a challenge for now but in time, I believe that cryptocurrencies are going to be the main payment method. This presents an advantage to applications like WorkShare, as they can leverage this time to improve and optimize their features but also a disadvantage because there are not a lot of users that want to use, test and give feedback about the application.

During times of congestion on the Ethereum blockchain, gas fees can skyrocket, rendering numerous transactions economically unfeasible. This poses a significant obstacle to the widespread adoption of blockchain technology. To mitigate this challenge, I have explored solutions provided by Polygon. By utilizing the Polygon network 3.4, the application can benefit from its layer 2 scaling solutions, which offer faster transactions and significantly lower fees compared to the Ethereum mainnet. This integration ensures that users of the WorkShare application can enjoy seamless and cost-effective transactions, with the security of Ethereum network, even during periods of high network congestion.

Despite the immense potential of smart contracts, their adoption within financial markets has been relatively slow and limited. This can be attributed to various risks that need to be addressed. Technology-related issues, such as coding errors, pose challenges that must be carefully mitigated to ensure the proper functioning and security of smart contracts as they are immutable. Additionally, legal and regulatory uncertainties surrounding smart contracts create complexities in terms of accountability and compliance, further impeding their widespread adoption.

Companies may face difficulties in finding trustworthy managers to whom they can entrust the funds required to start a new project or task via the smart contract. Additionally, companies may encounter situations in which a certain task needs to be completed, but no developer is willing to take it on. However, this issue can often be resolved with a greater incentive or reward. Finally, there may be cases in which a developer is unable to complete a task due to reasons such as lack of knowledge, poor time management, or personal issues. In such cases, the developer would not receive payment for the work completed, and may also receive a negative review

(NFT) depending on the case.

During the early stages of the application, some developers may take advantage of the lack of feedback and blockchain data on a developer's previous work. They might accept a project and produce unsatisfactory work, then create a new account to start fresh after receiving a negative review. To address this issue, developers could be required to register and undergo verification by a specialized team. Moreover, developers could receive different NFTs based on their skills and experience, enabling clients to make better-informed decisions about whom they work with. This approach could improve the quality of work delivered through the application and help establish trust among users.

Another one of the significant challenges that the WorkShare app might encounter is effectively communicating project ideas to developers. A potential issue arises when the description of a project is ambiguous or difficult for developers to grasp, leading to possible misunderstandings regarding their expected deliverables. Ensuring clear and concise expression of project concepts becomes crucial for seamless collaboration and efficient development processes. Without precise communication, there is a risk of misinterpretation, which can hamper productivity and the overall success of the project. Therefore, the WorkShare app needs to address this challenge by implementing pattern that facilitate comprehensive and unambiguous project descriptions, ensuring developers have a clear understanding of their tasks and goals.

4.5 Application flow

The WorkShare application offers a streamlined and efficient workflow for developers and project managers to collaborate on projects. This chapter outlines the step-by-step application flow, ensuring a seamless experience for all users involved. 4.2

1. User Registration:

   Developers are required to register on the WorkShare platform. Registration is an easy process as the developers only have to provide the email.

2. Project Exploration and Application:

   Developers browse through the available projects listed on the WorkShare platform, each project includes a short description, requirements, compensation information, and the deadline, for more information if you press on the project card you are going to be redirected to a PDF stored of Pinata , where all

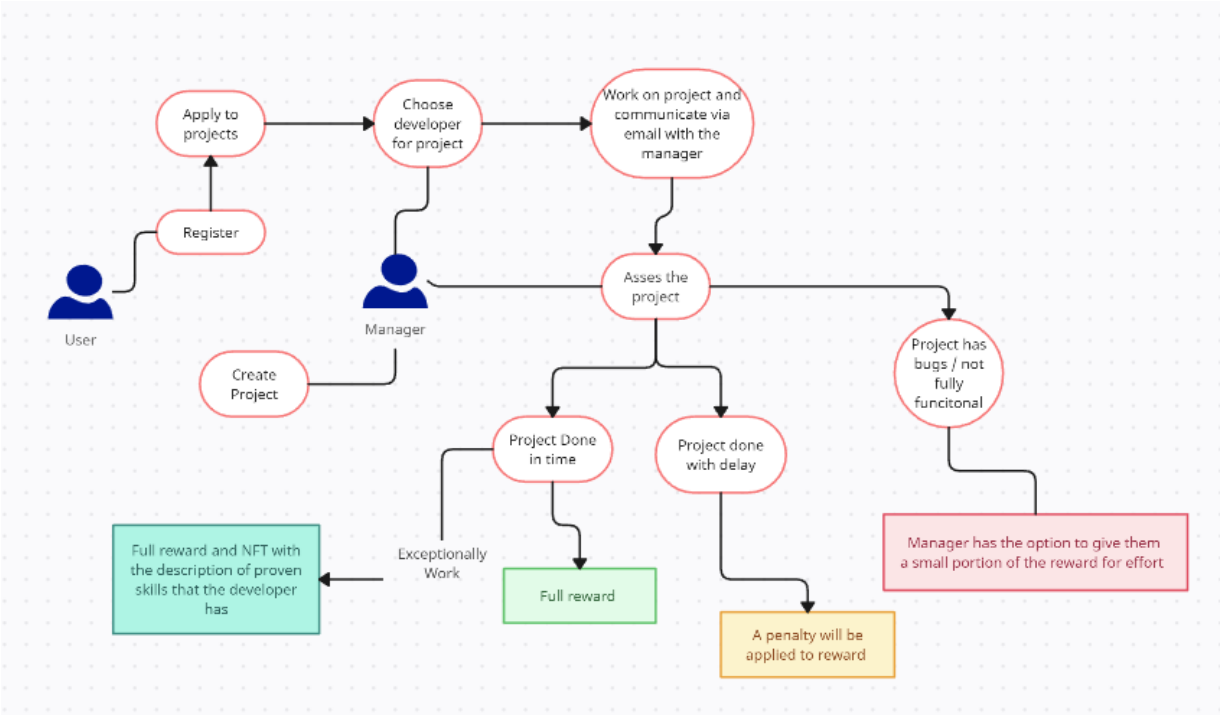

Figure 4.2: Application flow diagram

the immutable information about the project are stored. Using the platform's interface, developers apply for the chosen projects.

3. Project Manager Review and Selection:

   Project managers carefully review the applications received from developers. They assess each applicant's qualifications, experience, and suitability for the project based on MasteryMilestones NFTs that were earned by the developer and blockchain data about completed projects.

4. Communication and Collaboration:

   Once a developer is selected for a project, project managers and developers initiate communication via email discuss project details, timelines, deliverables, and any other relevant information.The email communication serves as a primary means for collaboration and updates throughout the project duration

5. Project Execution and Completion:

   Developers work diligently on their assigned projects, following the agreed-upon timelines and deliverables, utilize their skills and expertise to complete the project according to the requirements and quality standards. During the project execution, project managers may provide guidance, feedback, and support to developers as needed.

6. Project Assessment and Rewards :

   Upon project completion, project managers assess the quality and performance of the delivered work. They evaluate the developer's contribution, adherence to timelines, and overall project outcome. Based on the assessment, developers receive rewards and compensation including even NFTs which are a proof of skills, which may vary depending on the project's complexity and their level of performance.

4.6 Functionalities

The Functionalities chapter provides an overview of the key features and capabilities of the application. It highlights the various functionalities that have been implemented to address the needs and requirements of the users. This chapter explores the core functionalities of the application, such as user registration, project application, project assessment, and rewards distribution. By exploring the specific functionalities, readers gain a comprehensive understanding of how the application operates and how it enables efficient project management and collaboration.

4.6.1 Connect wallet

Connecting a wallet in a decentralized application (dApp) is an important feature that allows users to interact with the blockchain and conduct various operations including as transferring tokens, signing transactions, or executing smart contracts.

The user must first have a compatible wallet, such as MetaMask, Trust Wallet, or Coinbase Wallet, in order to link a wallet in a dApp. These wallets often communicate with the blockchain via a browser extension or a mobile app, allowing the user to securely manage their digital assets.

When a user has a compatible wallet, they can connect it to the WorkShare application by clicking on the "Connect Wallet" from the home page. This operation sends a request to the wallet to approve the dApp connection, by displaying a pop-up window or notification asking the user to confirm the connection.

After the user confirms the connection, the application can access the user's wallet and interact with the blockchain on their behalf, such as querying the balance of their tokens or executing transactions.

4.6.2 Developer register

Once the user is connected to their wallet, the application can access their public address to create and store a mapping from the user's address to the email address

they provide. It will be helpful to provide additional details of projects and for communication process. To register, the user will be presented with an input field for their email address and a "Register" button 5.2 . When the user presses the button, the application will first validate the email address to ensure it meets the necessary format requirements.

Once the email address is validated, the application will then initiate a transaction request to the user's connected wallet to sign and approve the transaction. The wallet will display the gas fee required to execute the transaction, and the user will need to approve the transaction using their wallet interface. After the user approves the transaction, it will be broadcasted to the network and have a pending status. It may take several seconds or more for the transaction to be processed and confirmed on the blockchain.

4.6.3 Apply to a project

This subchapter describes the process of applying to a project on the platform. Once a user has connected their wallet, the manager can verify their work experience and completed projects on the blockchain using Polygonscan, a platform that allows users to access any blockchain transaction as well as any earned NFTs. However, the user must also be registered with a valid email address in order for the project manager to contact them.

To apply for a project, the user must first review the project description and requirements, which are securely stored on IPFS 5.3.6 to prevent tampering, and then just press the sky colored button Apply 5.3. It's important for the user to carefully consider their own skills and availability before applying to ensure that they are capable of completing the project on time.

The smart contract includes a function called "applyForProject" which checks that the project is in an open state, that the user is registered with a valid email address, and that they have not already applied for the project. If all these conditions are met, the user's address is added to the list of project applicants and the number of applicants is incremented.

4.6.4 Create project

To initiate a new project in the WorkShare application, the manager can utilize the visual appealing form 5.4 which calls the createProject function of the smart contract. This function allows the manager to specify the project's details, including a brief description, a requirements document CID, an NFT CID, the project's reward, penalty, and deadline. Upon invoking this function, the manager's account is verified as the project creator. The project's information is recorded on the blockchain,

and the corresponding reward is transferred to the smart contract. A small commission fee is deducted from the reward, with the remaining amount allocated as the project's actual reward. The event of project creation is emitted, providing the necessary details for further tracking and management.

4.6.5 Accept application

The manager can use the acceptApplication method to allocate a project to a developer within the WorkShare application. The manager can choose the best match for the project after reviewing the list of applicants and examining their prior work on the platform. This guarantees that the chosen developer has the required abilities and expertise. The manager can formally accept the chosen candidate for the associated project by supplying the project number and their location. When the application is approved, the project's status is changed to "In Progress," indicating that development work has begun.

4.6.6 Project Communication, Assessment, and Finalization

This subchapter focuses on the communication process between the manager and the developer in the WorkShare application. It outlines how the manager can effectively communicate with the developer through email or other means to discuss project details and progress. Upon project completion, the manager assesses the outcome and determines the rewards based on the developer's performance. If the project is successfully completed, the developer receives the full reward and a NFT minted from the MasteryMilestone smart contract as recognition. However, if the project is not completed within the specified deadline, the manager can call the "projectNotCompleted" function to address the situation. This function allows the manager to evaluate the developer's effort and allocate a partial reward accordingly, while returning the remaining tokens to the manager.

In conclusion, this chapter explored the essential functionalities of the WorkShare application, including project communication, assessment, and finalization. These features, along with others like granting/revoking admin roles, commission withdrawal, and project manager changes,which assures the correct functioning of the application and can only be called by the owner of the project establish a robust platform for efficient company management with freelancers. With ongoing enhancements and additional functionalities like token handling, WorkShare sets the stage for continued efficiency and improvement in the digital workspace. In the next chapter, we will delve into the implementation details of these functionalities, providing a more in-depth exploration of the inner workings of the WorkShare application.

Chapter 5

Implementation Details: Building WorkShare as a Smart Contract-Based Application

This chapter provides a detailed overview of the technical aspects of building the WorkShare app. It covers topics such as the choice of programming languages and frameworks, the design of the app's architecture, the use of smart contracts and blockchain technology,the deployment process for the WorkShareToken, MasteryMilestone and WorkShare smart contracts 5.1 and the integration of various APIs and third-party services. The chapter aims to give the reader an in-depth understanding of the development process and the technologies used to build the app.

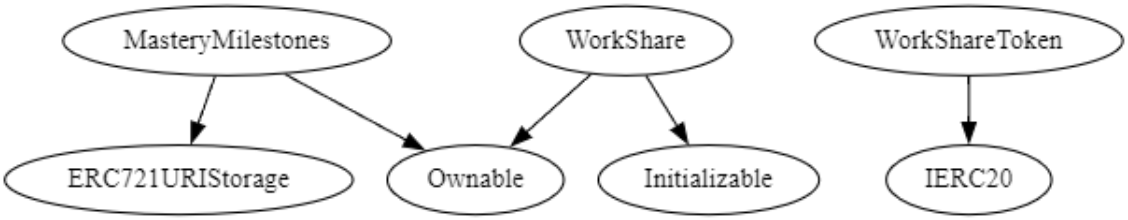

Figure 5.1: Inheritance graph

5.1 Deploying the smart contracts

This chapter describes the process of deploying the WorkShare, WorkShare Token and Mastery Milestone smart contracts using the Hardhat tool, MetaMask wallet, and Web3.js.

Before we dive into the process, let's briefly discuss what a smart contract is. A smart contract is a self-executing program that runs on a blockchain network. It

consists of a set of rules and regulations that are programmed into the contract and can be executed automatically once certain conditions are met 3.2.

Deploying a smart contract involves the following steps:

1. Setting up the development environment Firstly, the development environment for smart contract deployment need to be set. For this I used Hardhat, an open-source development environment for building and deploying smart contracts, MetaMask wallet, which allows us to interact with the Polygon network and Web3.js, a JavaScript library that provides a simple interface for interacting with smart contracts.

2. Writing the smart contract code I wrote the code using Solidity 3.2.1, which is a contract-oriented programming language specifically designed for Ethereum. Once the code is written, I used Hardhat tool for compiling it and check for any errors. Also I manual functional tested the smart contract using Remix, an integrated development environment which has a nice and easy to use interface for interacting with smart contracts.

3. Testing the smart contract Before deploying the smart contract, I tested the smart contracts thoroughly to ensure they work as expected. For this purpose, I used the Hardhat testing framework. I wrote tests cases to test various scenarios and edge cases as the rewards distribution, multiple registration, non-authorized request. This step is important as it helps to catch any bugs or issues in the contract before it is deployed on the mainnet.

4. Deploying the smart contract on the network Once the smart contract was tested and ready to be deployed, I needed to connect the MetaMask wallet to the Polygon network, where we want to deploy the contract using the following command with the help of the Hardhat tool:

   ```
   npx hardhat run --network polygon scripts/deploy.js
   ```

   This command will trigger the deployment script, which will compile the contract, will initialize the parameters needed by the WorkShare contract, create a deployment package, and deploy it on the Polygon network.

5. Interacting with the smart contract Once the smart contract is deployed, we can interact with it using Web3.js library. I used the contract's address and ABI (Application Binary Interface) to create an instance of the contract in our JavaScript code. Now we can then call various functions of the contract and interact with it.

In conclusion, deploying a smart contract involves setting up the development environment, writing the code, testing the contract, deploying it on the network,

and finally interacting with it using Web3.js or other libraries. By following these steps, we can deploy secure and reliable smart contracts that can be used for a variety of purposes on the blockchain network.

5.2 Design of the application

Design is a critical aspect of any software application, and it plays a significant role in shaping the user's experience. A well-designed application can make all the difference when it comes to user engagement, retention, and satisfaction. This chapter will describe the design of the WorkShare application, which has been implemented using React JS.

The WorkShare application has been designed to be intuitive, user-friendly, and visually appealing. The User Interface (UI) has been carefully crafted to ensure that users can easily navigate through the application, access the features they need, and complete their tasks efficiently. The design of the application has been guided by the principles of User Experience (UX) design, which focuses on enhancing the user's satisfaction and engagement by improving the usability, accessibility, and pleasure provided in the interaction with the product.

The WorkShare application uses a dark colour palette with a dark blue to grey gradient background, which provides a calming effect and a professional look. The logo 5.2 of the application is purple, which is a complementing colour to blue and helps to add contrast to the design. The images used in the application are visually interesting, and they help to add a sense of liveliness to the application.

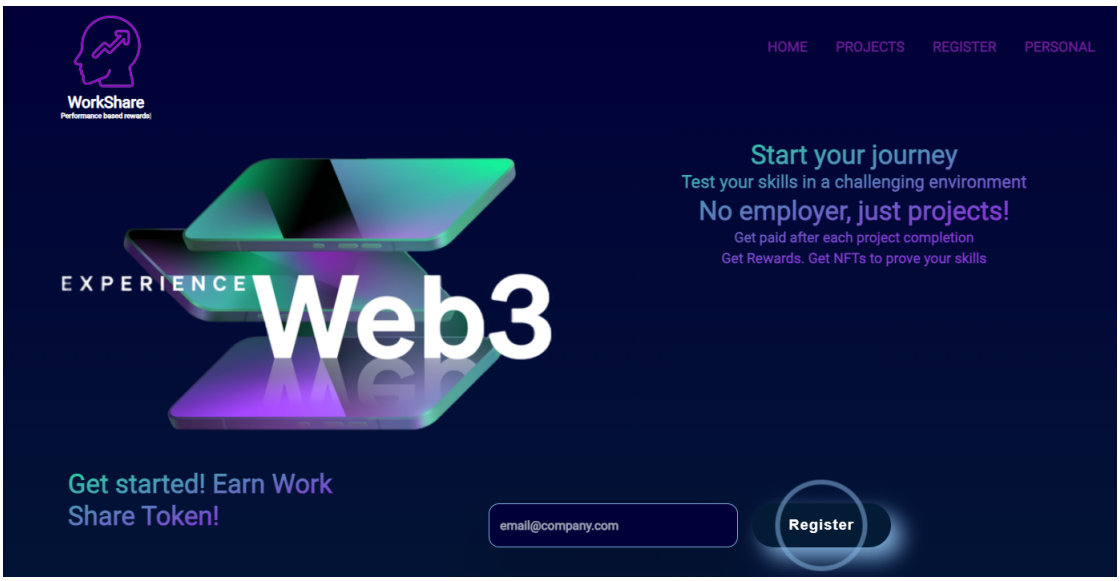

Figure 5.2: Register button

The buttons in the application have been given special effects, which makes them

stand out. This is an excellent technique to draw the user's attention to the most important parts of the application. For example, the "Register" button has a hover 5.2, a ring effect and a shadow, which makes it more visible, interactive and engaging for the user.

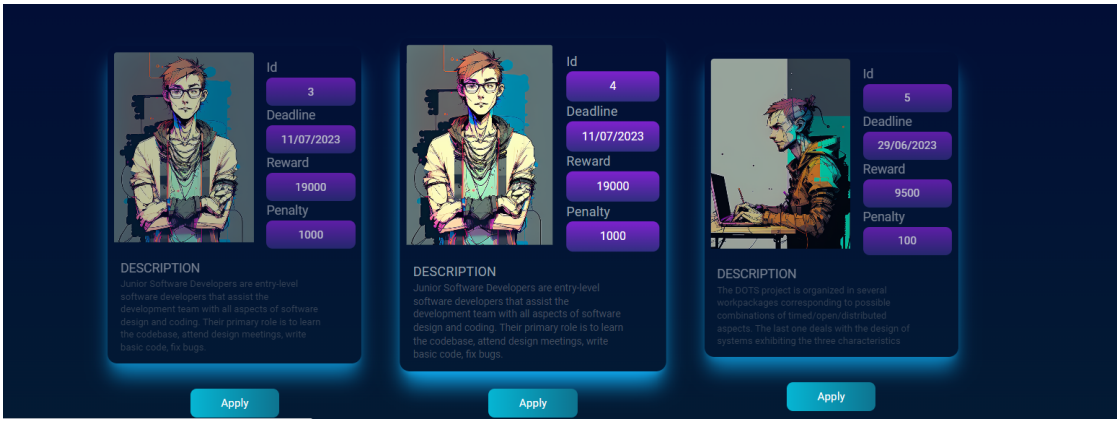

Figure 5.3: Projects list

The project card 5.3 in the application has been designed to be visually appealing and user-friendly. The card has a sky-coloured shadow, which adds depth and makes the card stand out. It includes all the relevant information about the project, such as the title, description, rewards, and time frame. Users can easily view all the relevant details of a project and decide whether to apply for it or not.

The form 5.4 used for introducing details about a project and input fields for different purposes have a shadow of purple, the color of the logo.

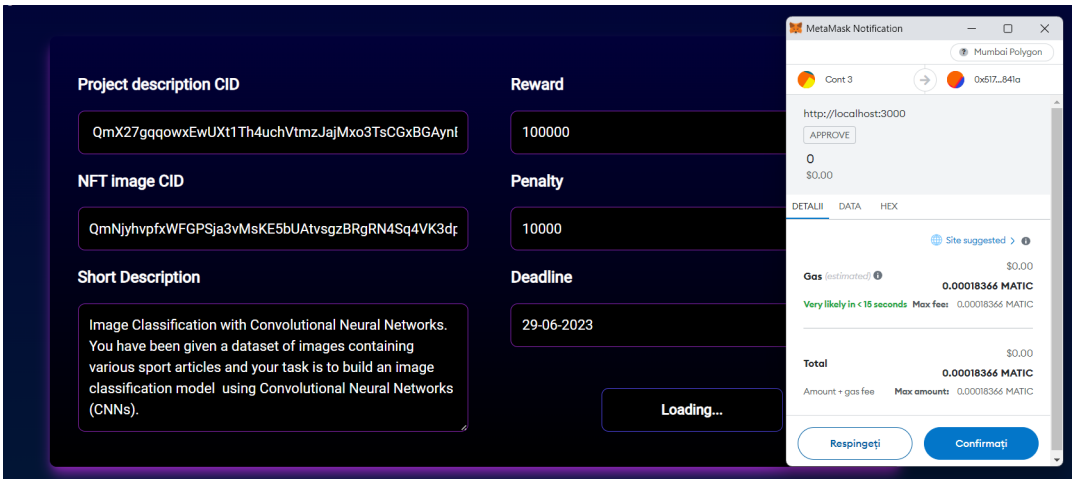

Figure 5.4: Create project form

In conclusion, the design of the WorkShare application has been carefully crafted to ensure that users have a positive and engaging experience while using the application. The UI has been designed using React JS and has been guided by the principles of UX design, with a focus on improving usability, accessibility, and pleasure.

The application uses a dark colour palette with a blue background, and the buttons have special effects to draw attention to important parts of the application. The project card has a sky-coloured shadow and provides all the relevant details about the project, making it easy for users to make informed decisions. The form is made with purple shadows so it offers an interesting visual contrast with the dark theme of the application.

5.3 Technical architecture and components

The technical architecture of WorkShare is designed to provide a robust and scalable platform for efficient project management and compensation. This chapter will outline the key components and their functionalities that make up the WorkShare application, ensuring a seamless and secure user experience. 5.5

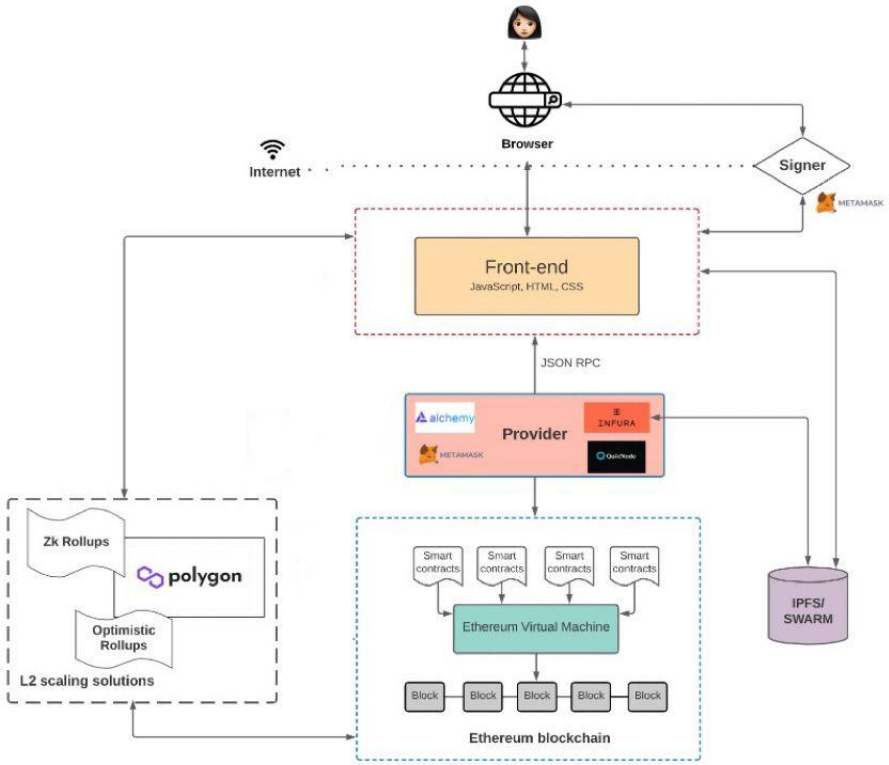

Figure 5.5: Technical architecture

5.3.1 Decentralized Application

The architecture of Web 3.0 applications (or "DApps") is completely different from Web 2.0 applications. Web 2.0 applications typically have a constantly updated database, backend code that defines application business logic, and frontend code

that defines UI logic. However, Web 3.0 applications eliminate the middle man by leveraging blockchain technology. There is no centralized database or web server for the backend logic. Instead, blockchain allows for the construction of decentralized state machines maintained by anonymous nodes on the internet.

At the core of WorkShare is a decentralized application built on blockchain technology, but has certain centralized elements for administrative purposes while maintaining decentralization in other aspects. The contract owner has the authority to grant or revoke administrative privileges and intervene in case of conflicts between project managers and developers. His role is to ensure a smooth and efficient process. The DApp leverages smart contracts to facilitate secure and transparent interactions between employers and developers. It provides a user-friendly interface for seamless navigation and interaction with the WorkShare ecosystem.

5.3.2 Polygon Network Integration

WorkShare integrates with the Polygon network 3.4 , a Layer 2 scaling solution for Ethereum. This integration enables fast and cost-effective transactions, with significantly reduced gas fees compared to the Ethereum mainnet. By leveraging Polygon's high throughput and low-latency infrastructure, WorkShare ensures smooth and efficient operations for users.

5.3.3 Signing Transaction - Metamask

Metamask is a tool that makes it easy for applications to handle key management and transaction signing. It's pretty simple: Metamask stores a user's private keys in the browser, and whenever the frontend needs the user to sign a transaction, it calls on Metamask.

It also provides a connection to the blockchain (as a provider) since it already has a connection to the nodes provided by Infura ("Infura is the service provider that MetaMask uses to get information on and off blockchains."[tea23]) since it needs it to sign transactions. In this way, Metamask is both a provider and a signer.

5.3.4 Decentralized Communication

The communication between the WorkShare frontend and smart contracts is facilitated through a web3 provider like Metamask or Alchemy, which implements a JSON-RPC specification. This ensures that there's a uniform set of methods when frontend applications want to interact with the blockchain. Instead of relying on a centralized backend server, WorkShare utilizes smart contracts 2.2 to define the

logic of the application. These contracts are deployed onto the decentralized state machine, shared by all participants in the network and visible for everyone.

5.3.5 Frontend Interface

WorkShare features a user-friendly frontend interface 5.2 that allows seamless interaction with the DApp. The frontend interface provides intuitive navigation, project browsing, application submission, and communication features. It is designed to enhance user experience and facilitate efficient project management.

5.3.6 IPFS Pinata - Inter planetary file system

IPFS, short for InterPlanetary File System, is a protocol and network designed to create a permanent and decentralized method of storing and sharing immutable files. It is a distributed system that allows users to access and share files without the need for a centralized server or storage location. [Las21]

In the project, I use Pinata IPFS 5.6 to store the project descriptions and requirements, ensuring that the documents are tamper-proof and cannot be changed after they are uploaded. Each project has its own unique IPFS hash, which serves as its identifier. By using IPFS, we can also ensure that the files are easily accessible to anyone who wants to view them, as they are distributed across the network, and can be accessed from any IPFS node.

One of the main advantages of using IPFS is that it provides a high level of redundancy and fault-tolerance. Since the files are distributed across the network, there is no single point of failure. If one node goes down, the file can still be accessed from another node. Additionally, IPFS uses content-addressing, which means that files are identified by their content, rather than their location. This ensures that the file can be accessed even if the location of the original uploader is unknown.

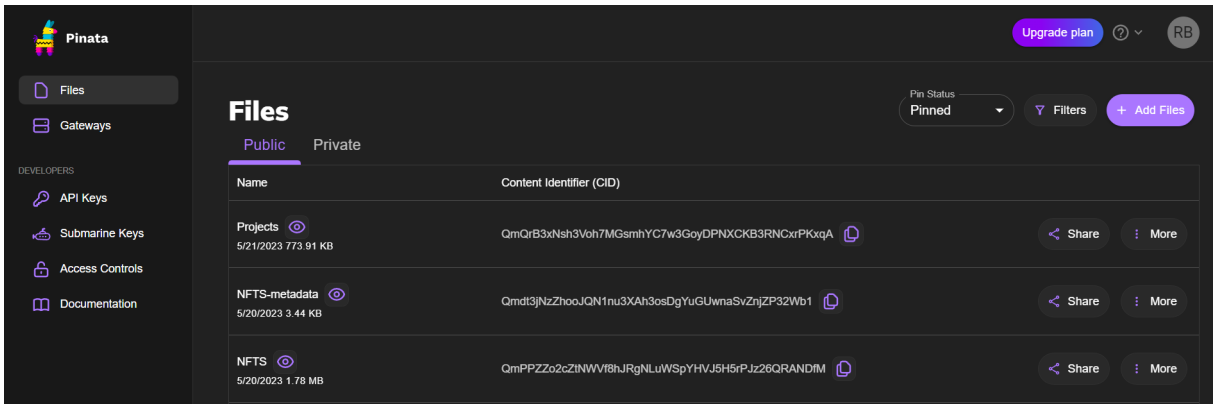

Figure 5.6: Pinata interface

To upload NFT data, you'll need to create a JSON file with the metadata for the NFT. This file should contain information like the name of the NFT, the description, image (a link where the image of the nft is stored) and a list of attributes of trait type, value pair. After you've selected your desired file to upload, Pinata will start uploading it to the IPFS network. This may take a few minutes, depending on the size of the file and the current state of the IPFS network. Once the file is uploaded, Pinata will provide a unique CID (content identifier) for your file. This CID is like a fingerprint for the data, and can be used to retrieve the data from the IPFS network. To upload a PDF with project details, you can follow a similar process. When someone wants to view the NFT or the project details, they can use the CIDs to retrieve the data from the IPFS network.

Overall, IPFS provides a secure and reliable way to store and share files in a decentralized manner, making it an ideal choice for the project.

5.4 Testing

Testing holds immense importance in the software development lifecycle, particularly when it comes to smart contracts, as they are both immutable and entrusted with users' funds. The criticality of testing lies in its ability to ensure the quality, reliability, and security of these applications. It involves a systematic approach to identifying, validating, and resolving defects, bugs, and errors throughout the development process. This includes conducting comprehensive unit testing, integration testing, and system testing to thoroughly evaluate the smart contract's behavior and minimize potential vulnerabilities. By prioritizing testing, we can establish trust, safeguard user funds, and uphold the integrity of smart contracts.

The WorkShare application, incorporates a range of features designed to facilitate seamless collaboration between developers and project managers. For a main feature of the application, such as the project application functionality, unit testing can be performed by creating test cases that cover different scenarios. These test cases can include verifying that the user registration process is working correctly, checking if the project details are displayed accurately, and validating that the application properly handles errors or edge cases. Unit testing is also applicable to smart contracts. In the context of smart contract development, unit tests are used to verify the functionality and behavior of individual contract functions or components. These tests are executed in a test environment, using Hardhat as development environment.

For integration testing several modules would need to be tested when integrated. These modules may include the user registration, application, and finalization of a project, so that the transfers of WorkShare tokens to take place and to verify they

are working properly as WorkShareToken is another contract. Another integration test would be the functionality of minting a NFT after the finalization of a project, from the MasteryMilestones smart contract. The integration testing strategy suitable for this application would involve testing the interactions and data flow between these modules to ensure they work seamlessly together.These tests are executed on mumbai network, the testnet of polygon.

System testing was performed to evaluate the overall behavior and performance of the WorkShare application as a whole. Several scenarios need to be tested to ensure the system meets the desired requirements and functions as intended. These scenarios include testing user registration, verifying the smooth flow of project creation after the requirements document and NFT were uploaded on Pinata IPFS, an user application and then the manager approval, evaluating the accuracy of reward calculations based on project assessment and performance, and simulating various user interactions like applying to a project, all within the application's user interface.

Chapter 6

Concluzion

Throughout this bachelor thesis, we have taken a step into the future of compensation, a step into the world of WorkShare, an innovative application that take advantage of blockchain technology to address the challenges faced in project management and compensation. We explained the concepts of blockchain and smart contracts, the key functionalities, technical architecture, and implementation details of WorkShare. We have discussed how this creative solution deployed on Polygon network, which offers the security of Ethereum network, is transparent, and an efficient platform for companies and developers to collaborate, optimize resource allocation, and drive growth in the digital economy.

The research question explored if the actual compensation system could be improved, and the taxes lowered. We looked into the efficacy of WorkShare in its impact on project management and compensation. By harnessing the potential of smart contracts and blockchain technology, it has effectively cut transaction fees, streamlined payment procedures, and lessened the burden of taxes and contributions, so that developers could improve their earning in relation with the work they provide. Moreover, its integration with a global talent pool has engendered scalability. The integration of communication, assessment, and finalization functionalities has enhanced collaboration and operational efficiency. While WorkShare showcases numerous strengths, it is important to acknowledge its limitations and weaknesses. One criticism is the potential dependency on blockchain technology, which may introduce complexities and risks associated with its adoption and implementation in the legal terms of a country. Moreover, the reliance on smart contracts necessitates thorough auditing and testing to mitigate potential issues arising from incorrect coding or vulnerabilities. WorkShare's effectiveness may also be subject to regulatory frameworks and legal ambiguities surrounding cryptocurrencies and blockchain technology, hindering its widespread adoption.

The results obtained from the analysis of WorkShare hold significant implications for project management and compensation practices. The application's suc-

cess in optimizing resource allocation, reducing transaction costs, and expanding the talent pool can be generalized to various industries and sectors. Its impact extends beyond individual companies, as the efficient allocation of resources and improved compensation systems contribute to the overall growth and productivity of the digital economy. By streamlining processes and enhancing transparency, WorkShare sets a precedent for future decentralized applications in project management and compensation.

I recognize the importance of incentivizing developers to actively contribute their code to the WorkShare platform. In the future I plan to introduce reward mechanisms, such as token incentives or as a reputational advantage a special NFT , to recognize and appreciate developers contributions. These incentives will further motivate developers to participate, collaborate, and share their expertise.

In conclusion, the exploration of WorkShare has demonstrated its effectiveness in revolutionizing project management and compensation. The application's ability to leverage blockchain technology, smart contracts, and a global talent pool has addressed critical challenges, enabling companies to allocate resources efficiently and the posibility of stimulating growth in the digital economy. However, it is essential to acknowledge the limitations and potential risks associated with its implementation. The generalizability of WorkShare's results highlights its impact on transforming traditional project management practices. As we conclude this bachelor degree, the possibilities for further enhancements and advancements in WorkShare and similar decentralized applications are boundless.

Bibliography

[Ana23a] Anaf. Ghid privind declararea şi plata contribuŢiilor sociale. https://static.anaf.ro/static/10/Anaf/AsistentaContribuabili_r/en_Ghid_contributii.pdf, 2023. Online. Accessed on: 05-13-2023.

[Ana23b] Anaf. Tratamentul fiscal al veniturilor realizate de persoanele fizice din transferul de monedă virtuală. https://static.anaf.ro/static/10/Anaf/AsistentaContribuabili_r/Brosusa_Criptomonede_2023.pdf, 2023. Online. Accessed on: 13-05-2023.

[Aut16] The Solidity Authors. Solidity v0.8.0 breaking changes. https://docs.soliditylang.org/en/v0.8.19/080-breaking-changes.html, 2016. Online. Accessed on: 04-05-2023.

[Bre22] Techskill Brew. Cryptography in blockchain. https://medium.com/techskill-brew/cryptography-in-blockchain-part-6-blockchain-basics-129ec058c57 2022. Online. Accessed on:03-29-2023.

[Bro22] Ryan Browne. Central african republic becomes second country to adopt bitcoin as legal tender. https://www.cnbc.com/2022/04/28/central-african-republic-adopts-bitcoin-as-legal-tender.html, 2022. Online. Accessed on: 06-07-2023.

[Cam23] Christine Campbell. What are the 4 different types of blockchain technology? https://www.techtarget.com/searchcio/feature/What-are-the-4-different-types-of-blockchain-technology, 2023. Online. Accessed on: 03-31-2023.

[cor23] Ethereum team corwintines. Scaling. https://ethereum.org/en/developers/docs/scaling/, 2023. Online. Accessed on: 06-08-2023.

[cou22] Blockchain council. What are blockchain nodes? detailed guide. https://www.blockchain-council.org/blockchain/blockchain-nodes/, 2022. 03-29-2023.

[Cre18] Doug Crescenzi. How to use openzeppelin's smart contracts. https://medium.com/upstate-interactive/how-to-use-openzeppelins-smart-contracts-d136f25c777, 2018. Online. Accessed on: 04-05-2023.

[doc23] OpenZeppelin docs. Access control. https://docs.openzeppelin.com/contracts/2.x/access-control, 2023. Online. Accessed on: 05-14-2023.

[Et23] corwintines Ethereum team. Gas and fees. https://ethereum.org/en/developers/docs/gas/, 2023. Online. Accessed on: 04-01-2023.

[eth16] Solidity. https://docs.soliditylang.org/en/v0.8.19/, 2016. Online. Accessed on: 04-05-2023.

[FRA23] JAKE FRANKENFIELD. Distributed ledger technology (dlt): Definition and how it works. https://www.investopedia.com/terms/d/distributed-ledger-technology-dlt.asp, 2023. Online. Accessed on: 06-08-2023.

[gov23] American government. Central bank digital currency (cbdc). https://www.federalreserve.gov/cbdc-faqs.htm, 2023. Online. Accessed on: 05-13-2023.

[Las21] Eric LastName. What is ipfs? the inter-planetary file system explained! https://medium.com/geekculture/what-is-ipfs-the-inter-planetary-file-system-explained-40744a7a 2021. Online. Accessed on: 05-13-2023.

[Mac22] Mikk Raud Eli MacKinnon. China's digital currency and blockchain network: Disparate projects or two sides of the same coin? https://digichina.stanford.edu/work/chinas-digital-currency-and-blockchain-network-disparate-projec 2022. Online. Accessed on: 05-13-2023.

[Pra22] Zubin Pratap. Upgradable smart contracts: What they are and how to deploy your own. https://blog.chain.link/upgradable-smart-contracts/, 2022. Online. Accessed on: 04-05-2023.

[Sal23] Danny Salman. Polygon pos architecture. https://wiki.polygon.technology/docs/home/architecture/polygon-architecture/, 2023. Online. Accessed on: 05-19-2023.

[SB18] Bhargav Perepa Sloane Brakeville. Blockchain basics: Introduction to distributed ledgers. https://developer.ibm.com/tutorials/cl-blockchain-basics-intro-bluemix-trs/, 2018. Online. Accessed on: 03-29-2023.

[Sin22] Ashish Sinha. Smart contracts could improve efficiency and transparency in financial transactions. https://www.spglobal.com/en/research-insights/featured/special-editorial/smart-contracts-could-improve-efficiency-and-transparency-in-fi 2022. Online. Accessed on: 05-17-2023.

[tea19] Ethereum team. What is ethereum ? https://ethereum.org/en/what-is-ethereum/, 2019. Online. Accessed on: 03-31-2023.

[tea20a] Ethereum team. Erc-721 non-fungible token standard. https://ethereum.org/en/developers/docs/standards/tokens/erc-721/, 2020. Online. Accessed on: 04-05-2023.

[tea20b] Ethereum team. Understand the erc-20 token smart contract. https://ethereum.org/en/developers/tutorials/understand-the-erc-20-token-smart-contract/, 2020. Online. Accessed on: 04-05-2023.

[Tea22] FXCM Research Team. Polygon — how it works, price predictions, matic token. https://www.fxcm.com/markets/insights/polygon-matic-guide/, 2022. Online. Accessed on: 05-18-2023.

[tea23] Metamask team. What is infura, and why does metamask use it? https://support.metamask.io/hc/en-us/articles/4417315392795-What-is-Infura-and-why-does-MetaMask-use-it-, 2023. Online. Accessed on: 06-10-2023.

[Tru22] TrustChain. All you need to know about upgradable smart contracts. https://trustchain.medium.com/all-you-need-to-know-about-upgradable-proxies-865704a28bc7, 2022. Online. Accessed on: 04-05-2023.