Dissertation on

“Tender Management System using Blockchain”

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE19CS390B – Capstone Project Phase - 2

Submitted by:

Garuda Yochna Sheetal	**PES2UG19CS126**
Ram Shabarish	**PES2UG19CS135**
Jinka Sagar	**PES2UG19CS168**
Sathvik Allundi	**PES2UG19CS368**

Under the guidance of

Prof. Animesh Giri
Assistant Professor
PES University

June - Nov 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

‘Tender Management System using Blockchain’

is a bonafide work carried out by

Garuda Yochna Sheetal	**PES2UG19CS126**
Ram Shabarish	**PES2UG19CS135**
Jinka Sagar	**PES2UG19CS168**
Sathvik Allundi	**PES2UG19CS368**

In partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE19CS390B) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period June 2022 – Nov. 2022. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature	Signature	Signature
Animesh Giri	Dr. Sandesh B J	Dr. B K Keshavan
Assistant Professor	Chairperson	Dean of Faculty

External Viva

Name of the Examiners **Signature with Date**

1. ____________________________ ____________________________

2. ____________________________ ____________________________

DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **"Tender Management System using Blockchain"** has been carried out by us under the guidance of Prof.Animesh Giri, Assistant Professor and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester June – Nov. 2022. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

Garuda Yochna Sheetal	PES2UG19CS126
Ram Shabarish	PES2UG19CS135
Jinka Sagar	PES2UG19CS168
Sathvik Allundi	PES2UG19CS368

ACKNOWLEDGEMENT

We would like to express our gratitude to Prof. Animesh Giri Department of Computer Science and Engineering, PES University, for his continuous guidance, assistance, and encouragement throughout the development of this UE19CS390B -Capstone Project Phase – 2.

We are grateful to the Capstone Project Coordinator, Dr. Annapurna , Professor and Prof. Gururaj, Assistant Professor, for organizing, managing, and helping with the entire process.

We take this opportunity to thank Dr. Sandesh B J, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support We have received from the department. We would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

We are deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing us various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement we have received from our family and friends.

ABSTRACT

For many years, Tender management has been a offline service and were handled by humans. There will be many expenses to operate an offline tender office, there needs to be proper infrastructure, office staff etc. Tender office means the location of the office of tender agent where the proposers will submit their bid for a tender. All these years people have to go to the office to submit their tenders which is a tedious process. Then came the online tender management system which will make the process easier and less expensive when compared to an offline tender office. As these tenders are very important and contains very confidential information storing them online is very risky. So we are proposing a tender management system using blockchain technology. Blockchain technology is shared immutable ledger that facilitates the process of recording transactions and recording assets in a blockchain network. Using this technology we aim to decentralize the tendering process that is excluding the intermediate or a third party who is responsible in organising the tendering process.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

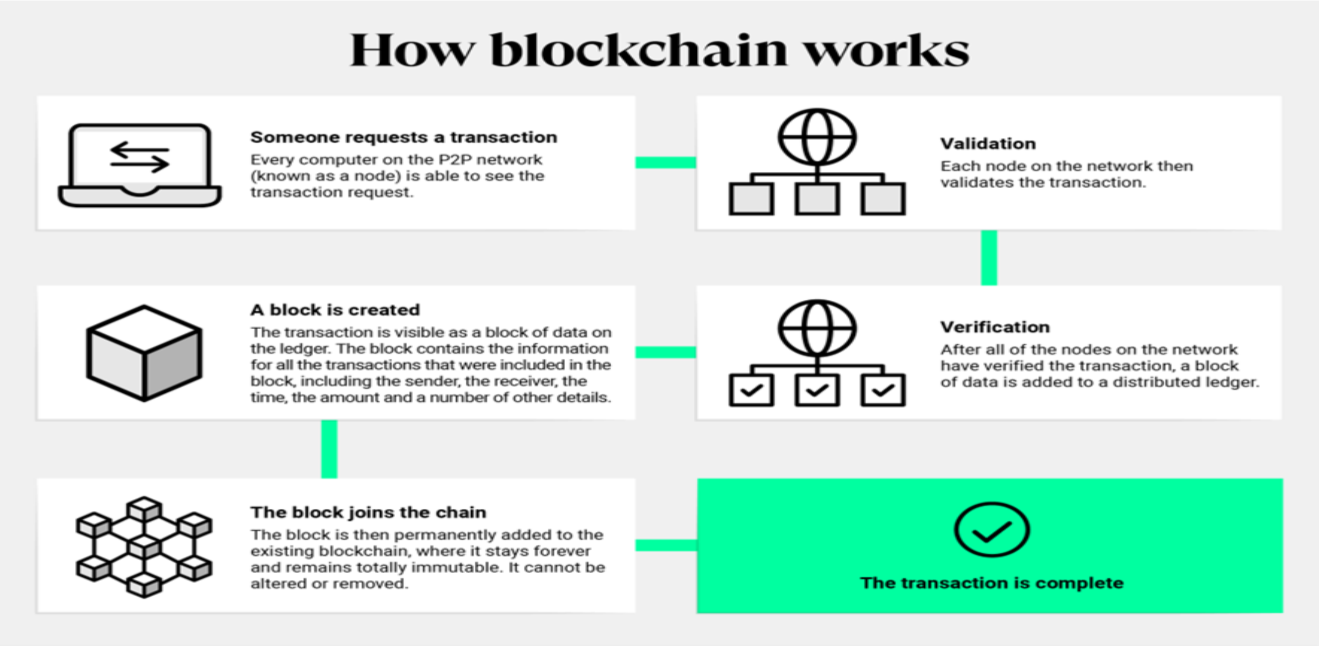
1.1 Blockchain

Blockchain is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a business network. An asset can be tangible (a house, car, cash, land) or intangible (intellectual property, patents, copyrights, branding). Virtually anything of value can be tracked and traded on a blockchain network, reducing risk and cutting costs for all involved.

Distributed ledger technology: All network participants have access to the distributed ledger and its immutable record of transactions. With this shared ledger, transactions are recorded only once, eliminating the duplication of effort that's typical of traditional business networks.

Immutable records: No participant can change or tamper with a transaction after it's been recorded to the shared ledger. If a transaction record includes an error, a new transaction must be added to reverse the error, and both transactions are then visible.

Smart contracts: To speed transactions, a set of rules called a smart contract is stored on the blockchain and executed automatically. A smart contract can define conditions for corporate bond transfers, include terms for travel insurance to be paid and much more.



How blockchain works

Someone requests a transaction
Every computer on the P2P network (known as a node) is able to see the transaction request.

Validation
Each node on the network then validates the transaction.

A block is created
The transaction is visible as a block of data on the ledger. The block contains the information for all the transactions that were included in the block, including the sender, the receiver, the time, the amount and a number of other details.

Verification
After all of the nodes on the network have verified the transaction, a block of data is added to a distributed ledger.

The block joins the chain
The block is then permanently added to the existing blockchain, where it stays forever and remains totally immutable. It cannot be altered or removed.

The transaction is complete

Figure 1.1.1 How blockchain works

How Blockchain works: As each transaction occurs, it is recorded as a “block” of data. Those transactions show the movement of an asset that can be tangible (a product) or intangible (intellectual). The data block can record the information of your choice: Who, what, when, where, how much and even the condition such as the temperature of a food shipment. Each block is connected to the ones before and after it. These blocks form a chain of data as an asset moves from place to place or ownership changes hands. The blocks confirm the exact time and sequence of transactions, and the blocks link securely together to prevent any block from being altered or a block being inserted between two existing blocks.

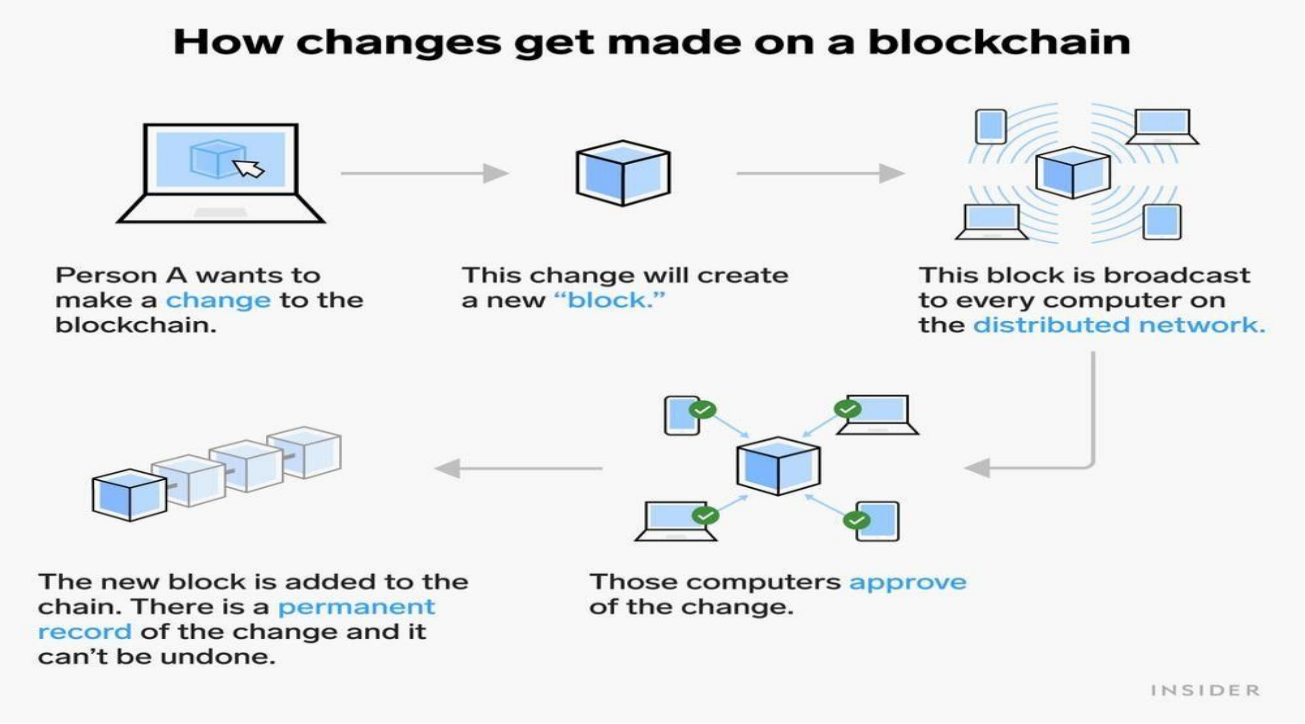

How changes get made on a blockchain
Person A wants to make a change to the blockchain.
This change will create a new "block."
This block is broadcast to every computer on the distributed network.
Those computers approve of the change.
The new block is added to the chain. There is a permanent record of the change and it can't be undone.
INSIDER

Figure 1.1.2 How changes get made on blockchain

Benefits of blockchain:

Trust: With blockchain, as a member of a members-only network, you can rest assured that you are receiving accurate and timely data, and that your confidential blockchain records will be shared only with network members to whom you have specifically granted access.

Security: Consensus on data accuracy is required from all network members, and all validated transactions are immutable because they are recorded permanently. No one, not even a system administrator, can delete a transaction.

Efficiency: With a distributed ledger that is shared among members of a network, time-wasting record reconciliations are eliminated. And to speed transactions, a set of rules called a smart contract can be stored on the blockchain and executed automatically.

1.2 Polygon

Polygon, formerly known as the Matic Network, is a scaling solution that aims to provide multiple tools to improve the speed and reduce the cost and complexities of transactions on blockchain networks. At the centre of Polygon's vision is Ethereum, a platform that is home to a range of decentralized applications, ones where you can join virtual worlds, play games, buy art, and participate in a range of financial services. However, this much activity on its blockchain has rendered Ethereum almost unusable, as the cost of transmission is rising and traffic is becoming clogged.

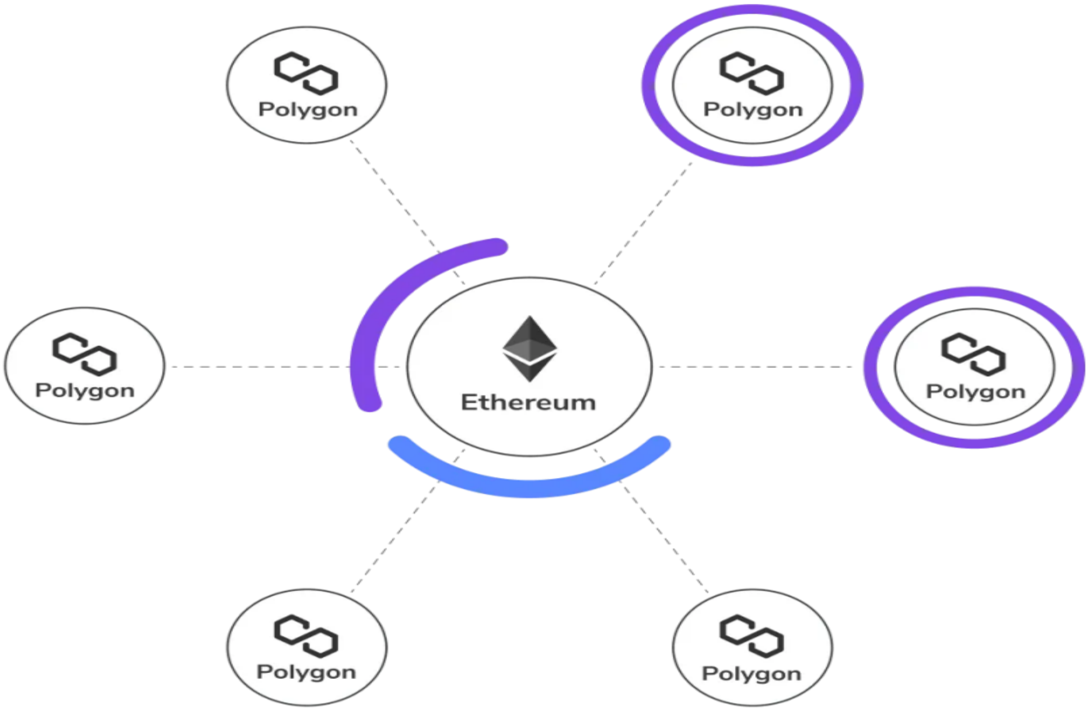


Figure 1.2.1 Polygon

In a nutshell, Polygon bills itself as a layer-2 network, meaning it acts as an add-on layer to Ethereum that does not seek to change the original blockchain layer. Like its geometric namesake, Polygon has many sides, shapes, and uses and promises a simpler framework for building interconnected networks. Polygon wants to help Ethereum expand in size, security, efficiency, and usefulness and seeks to spur developers to bring enticing products to market all the quicker.

Polygon PoS is a solution that achieves unprecedented transaction speed and cost savings by utilizing side-chains for transaction processing. At the same time, POS ensures asset security using the robust Plasma bridging framework and a decentralized network of Proof-of-Stake (PoS) validators.

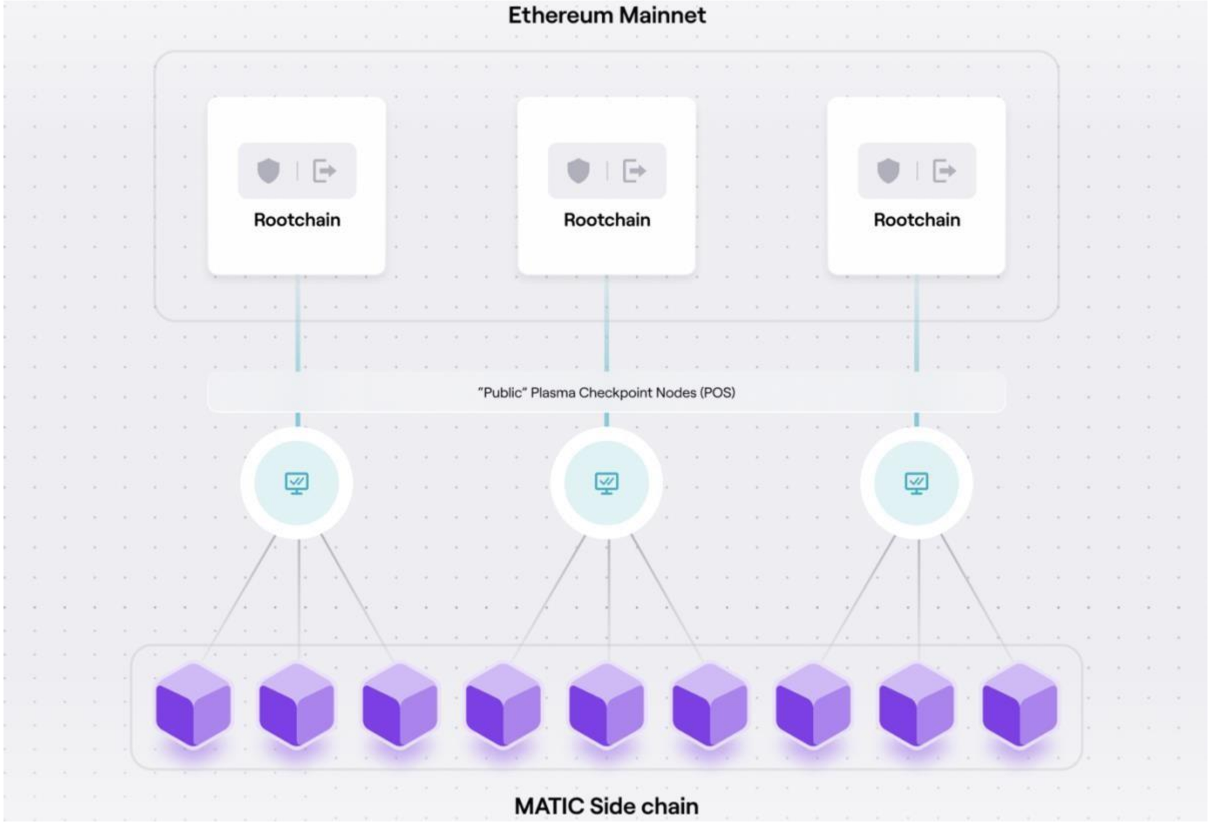


Figure 1.2.2 Polygon as sidechain

Key features of Polygon:

Speed: The Polygon Network uses a high-throughput blockchain with consensus provided by a group of Block Producers selected by stakeholders at each checkpoint. A Proof of Stake layer is used to validate blocks and periodically post proofs of Block Producers to the Ethereum mainnet. This enables rapid block confirmation rates of about 2 seconds while preserving a high amount of decentralization, resulting in excellent throughput for the network.

Scalability: Polygon Network achieves a hypothetical transaction speed of fewer than 2 seconds on a single sidechain. Using multiple sidechains helps the network to handle millions of transactions per second. This mechanism (already demonstrated in the first Matic sidechain) allows the Polygon network to scale easily.

Security: Polygon's smart contracts rely on Ethereum's security. To safeguard the network, it employs three critical security models. It uses Ethereum's staking management contracts and a group of incentivized validators running Heimdall and Bor nodes. Developers can also implement both models (Hybrid) into their dApp. Polygon supports many services you can use to test, compile, debug, and deploy decentralized applications onto the Polygon Network.

1.3 Smart contract

Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met.

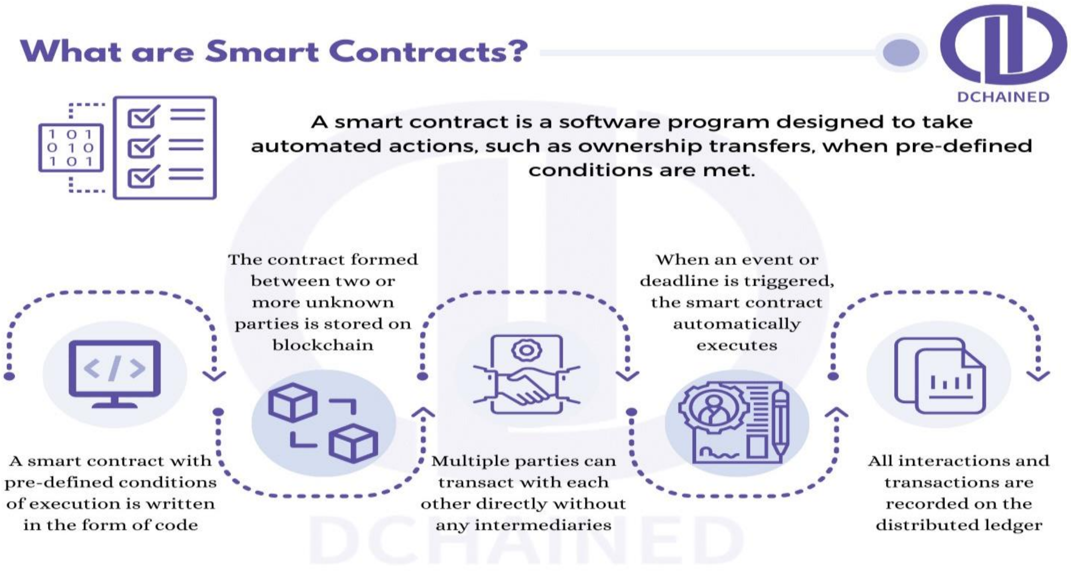

What are Smart Contracts?
DCHAINED
A smart contract is a software program designed to take automated actions, such as ownership transfers, when pre-defined conditions are met.
A smart contract with pre-defined conditions of execution is written in the form of code
The contract formed between two or more unknown parties is stored on blockchain
Multiple parties can transact with each other directly without any intermediaries
When an event or deadline is triggered, the smart contract automatically executes
All interactions and transactions are recorded on the distributed ledger
DCHAINED

Figure 1.3.1 What are smart contracts

How smart contracts work: Smart contracts work by following simple "if/when…then…" statements that are written into code on a blockchain. A network of computers executes the actions when predetermined conditions have been met and verified. These actions could include releasing funds to the appropriate parties, registering a vehicle, sending notifications, or issuing a ticket. The blockchain is then updated when the transaction is completed. That means the transaction cannot be changed, and only parties who have been granted permission can see the results.

Within a smart contract, there can be as many stipulations as needed to satisfy the participants that the task will be completed satisfactorily. To establish the terms, participants must determine how transactions and their data are represented on the blockchain, agree on the "if/when...then…" rules that govern those transactions, explore all possible exceptions, and define a framework for resolving disputes.

Then the smart contract can be programmed by a developer – although increasingly, organizations that use blockchain for business provide templates, web interfaces, and other online tools to simplify structuring smart contracts.

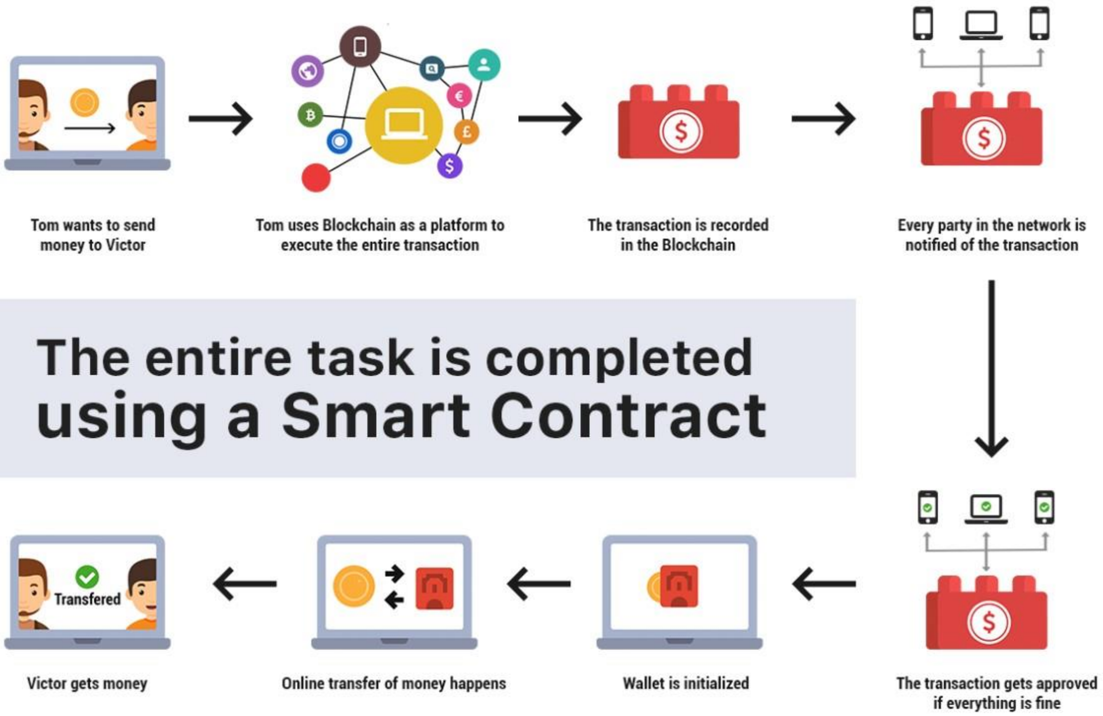

Tom wants to send money to Victor
Tom uses Blockchain as a platform to execute the entire transaction
The transaction is recorded in the Blockchain
Every party in the network is notified of the transaction
The entire task is completed using a Smart Contract
The transaction gets approved if everything is fine
Wallet is initialized
Online transfer of money happens
Transfered
Victor gets money

Figure 1.3.2 Smart contract flow

Benefits of smart contract:

Speed, efficiency and accuracy: Once a condition is met, the contract is executed immediately. Because smart contracts are digital and automated, there’s no paperwork to process and no time spent reconciling errors that often result from manually filling in documents.

Trust and transparency: Because there’s no third party involved, and because encrypted records of transactions are shared across participants, there’s no need to question whether information has been altered for personal benefit.

Security: Blockchain transaction records are encrypted, which makes them very hard to hack. Moreover, because each record is connected to the previous and subsequent records on a distributed ledger, hackers would have to alter the entire chain to change a single record.

Savings: Smart contracts remove the need for intermediaries to handle transactions and, by extension, their associated time delays and fees.

1.4 Hardhat

Hardhat is a development environment for Ethereum software. It consists of different components for editing, compiling, debugging and deploying your smart contracts and dApps, all of which work together to create a complete development environment.

Hardhat Runner is the main component you interact with when using Hardhat. It's a flexible and extensible task runner that helps you manage and automate the recurring tasks inherent to developing smart contracts and dApps.

Hardhat Runner is designed around the concepts of tasks and plugins. Every time you're running Hardhat from the command-line, you're running a task. For example, npx hardhat compile runs the built-in compile task. Tasks can call other tasks, allowing complex workflows to be defined. Users and plugins can override existing tasks, making those workflows customizable and extendable.

CHAPTER 2

PROBLEM STATEMENT

Tender refers to an invitation to bid for a project. Tendering usually refers to the process where governments and financial institutions invite bids for large projects that must be submitted within a finite deadline. It can also refer to the acceptance of a formal offer. So tenders are very important and have to be properly handled.

A tender management system tries to solve the problem of offline tender offices where they are requested to submit their bids. Where the information can be tampered and data manipulation can be done. To overcome this online tender management system are developed. We cannot say that online tender management system are safe anymore as there are lot of cyber-attacks going on. To solve this problem we are building tender management system using blockchain technology. Blockchain technology is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a blockchain network. Using this technology we can make sure that the transactions are not duplicated and the data is not tampered. This tender management system uses blockchain technology to send their bids to the contractors through blockchain.

CHAPTER 3

LITERATURE SURVEY

[1] Wang, D.; Zhao, J.; Mu, C. Research on Blockchain-Based E-Bidding System *Appl. Sci.* 2021, *11*, 4011.

Introduction: In the field of modern bidding, e-bidding leads a new trend of development, convenience and efficiency. Most systems need a strong and trusted third party to guarantee the security and integrity of the system, with the development of the blockchain technology and the rise in privacy protection, researchers have begun to focus on the core concept of decentralization. This paper introduces a decentralized electronic bidding system based on blockchain and smart contract. This system uses blockchain to replace the traditional database and uses chain to process the business logic. The decentralization, traceability and immutability of on-chain information storage makes blockchain a trusted machine with high reliability and security. Tendering and bidding is a kind of commodity trading behaviour, in simple words, it is an organized selection of excellent transaction ways by the tenderee. Compared with traditional offline bidding, it has an obvious difference in efficiency, information collection and other aspects, and is better in the identity authentication of the bidding object, confidentiality of the bidding content, fairness of the bidding process and other aspects.

Characteristics and Implementation: This paper presents a system framework that uses blockchain technology and smart contract to solve the privacy and security problems of E-bidding systems. The overall architecture of E-bidding systems based on blockchain technology is described. and provides a new solution for the reform of the E-bidding system based on blockchain technology.

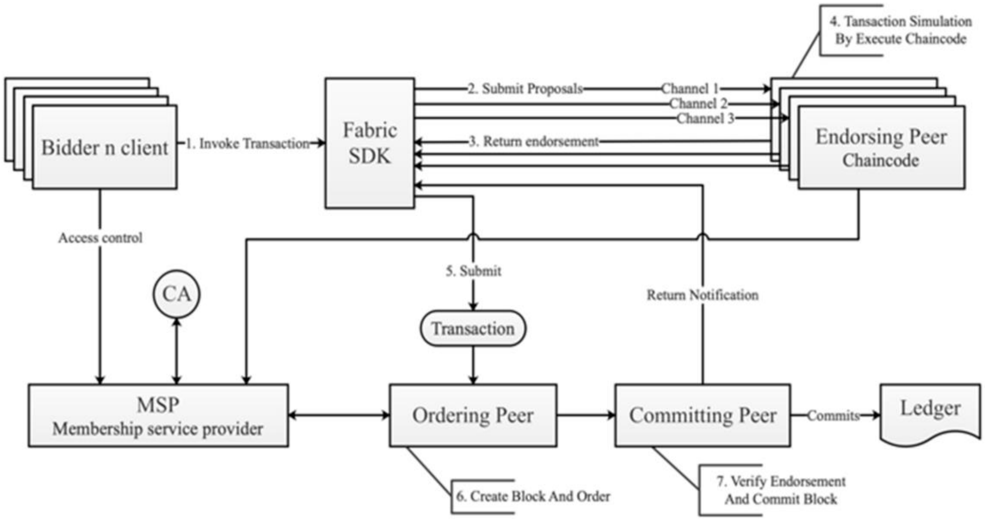


Figure 3.1.1 Framework and process of e-bidding system with HLF

Conclusion: This paper proposes the design and implementation method of a distributed E-bidding system based on blockchain. Its core idea is to combine blockchain technology with privacy protection, so that all participants using this system can safely participate in the opening stage. This paper uses a “Permissioned Blockchain” such as Hyperledger to develop the bidding process. Among them, chaincode and MSP can control the access of a single user together and can also set a time limit for it. Therefore, the risk of distributed denial of service (DDoS) attacks entering the network is greatly reduced, thereby ensuring the extensive security, confidentiality and non-repudiation of the entire process.

Limitations: HLF is a very new technology with rapidly changing functions. Establishing such a new network and getting other stakeholders to adapt to it is a huge challenge.
The lack of skilled blockchain technical personnel to operate and maintain the system is also an obstacle.

[2] Andrew, B.; Christopher, B.; Thomas, P. Digital Voting with the Use of Blockchain Technology.

Introduction: Democratic voting is serious and crucial event in any country. The most common way in which a country votes is through a paper based system. Digital voting is the use of electronic devices, such as voting machines or internet browsers, to cast the votes. They are referred as e-voting when using a voting machine in a polling station and i-voting when voting is done using internet browser. Security of digital voting is always the biggest concern when considering to implement a digital voting system. With such monumental decisions at stake, there can be no doubt about the system's ability to secure data and defend against potential attacks. One way the security issues can be potentially solved is through the technology of blockchains.

Characteristics and Implementation: The first aspect of our design is the registration process, verifying a voter is essential in establishing security within the system. Making sure that someone's identity isn't being misused for fraudulent purposes is important, especially when voting is considered, where every vote matters. Firstly, a transaction is created when user is registered and the next transaction is created when the government authorizes the users right to vote. As part of our design we have an encryption method based on public and private keys and have implemented a structure where the data is segregated within the blockchain. The public key is used to encrypt any vote made to that polling station. The data is then stored in an encrypted format within the blockchain and propagates out to the entire network.

Conclusion: The service proposal consists of a geographically distributed network comprising of both public and government infrastructure. This infrastructure houses two distinctly separate blockchains, one for voter information such as who has voted and the other for vote information such as what has been voted. The encryption mechanism that is being used would be close to impossible for any person to gain access to all votes without first taking control of the entire service network. The publication method of the private keys allows anyone to read the blockchain of votes and decrypt them with the newly available constituency private keys to verify the result of the election.

[3] F. S. Hardwick, R. N. Akram and K. Markantonakis, "Fair and Transparent Blockchain Based Tendering Framework - A Step Towards Open Governance"

Introduction: At a time when society is in constant transition to keep up with technological advancement, we are seeing traditional paradigms increasingly challenged open governance to mean a concept, which encourages and facilitates openness, accountability and responsiveness to citizens. For the success of open governance technologies like internet are crucial, other technologies like blockchain and smart contracts, which could be utilized to assist open governance. The phrase fair and open has different meaning based on the context of the situation. The paper proposes that the blockchain and smart contracts can enable an open governance framework that can facilitate citizens oversight on government functions that is easy to carry out with no associated financial costs. Therefore as a proof-of-concept, we explore the government tendering process, a set of activities that have three distinct phases: government tender opening, bidding period, and tender closing and selection of the best bid.

Characteristics and Implementation: In the implementation of the tender framework, Ethereum blockchain API has been used. Ethereum has been selected because it is open-source platform which is publicly available and a well known choice for distributed application. Each transaction has a gas usage and a gas cost. The gas usage is determined by how computationally expensive the transaction is. Truffle is used as framework to create contract and written in solidity language. For development and unit testing process a private blockchain is used running on testRPC Ethereum client.

Conclusion: Openness and transparency are frequently discussed in public service domain. Traditionally it has been difficult to build a transparent governance model. With increasing adoption of e-government and open government initiatives. Public opinion is in favour of developing innovative solutions that can increase openness and transparency in the government activities with minimum cost to citizens. To build such an environment, blockchain and smart contracts show great potential. In this paper blockchain and smart contract is used to implement the government tendering process to provide an open and fair tendering scheme. The main objective of the paper was to show that the tendering scheme can be made fair, open transparent and autonomous using smart contracts.

[4] Y. Goswami, A. Agrawal and A. Bhatia, "E-Governance: A Tendering Framework Using Blockchain With Active Participation of Citizens"

Introduction: The use of information technology for government activities to provide services, exchange information, etc. are becoming increasingly popular. One such area of e-governance is e-tendering. While e-tendering makes the tendering process more efficient a trust deficit remains between the government and citizens, due to centralized management of the whole tendering process. Therefore in this paper, a permissioned blockchain based approach to provide transparent and fair tendering system with the citizens active participation and tracking of funds is presented. In the proposed scheme the citizens can see the tendering activities transparently and give suggestions to solve governments posted problems and receive due credit of that contribution in a fair manner.

Characteristics and Implementation:

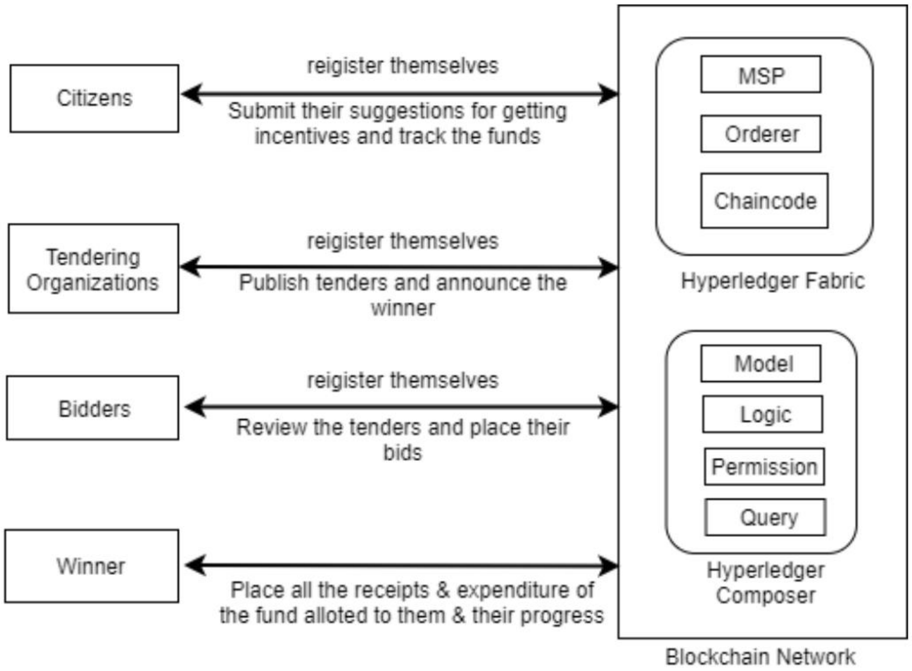

Citizens
reigister themselves
Submit their suggestions for getting incentives and track the funds
Tendering Organizations
reigister themselves
Publish tenders and announce the winner
Bidders
reigister themselves
Review the tenders and place their bids
Winner
Place all the receipts & expenditure of the fund alloted to them & their progress
MSP
Orderer
Chaincode
Hyperledger Fabric
Model
Logic
Permission
Query
Hyperledger Composer
Blockchain Network

Figure 3.4.1 proposed e-tendering framework using HLF

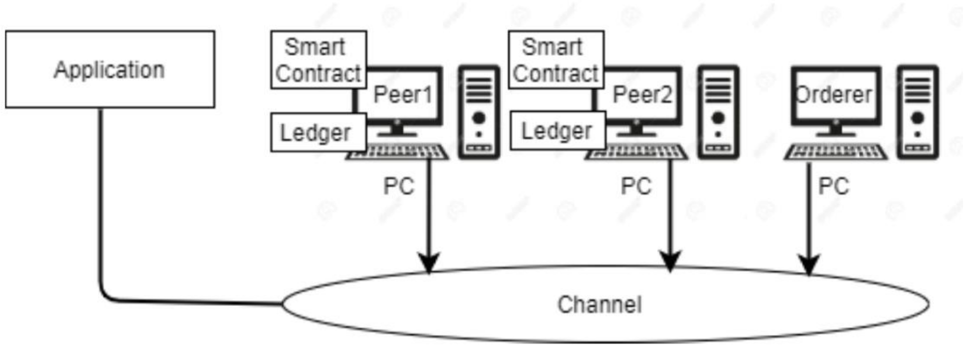


Figure 3.4.2 Blockchain framework

A blockchain network consisting of two peers and a orderer node has been developed as shown in Figure.3.4.2. The peers are connected via a channel on which the chaincode is installed and instantiated. The ordering service used is solo ordering service in which a single orderer node is responsible for providing ordering service. We have used docker containers to run our peers and other participants. The Logic for managing citizen ideas which allows citizens to post their suggestions, for tendering mechanism, incentives to be given to citizen based on their suggestions, fund tracking has been developed.

Conclusion: e-governance using blockchain has helped in achieving a transparent and fair solution to current issue of corruption and biases. Blockchain has allowed this of its property of immutability of data. The framework proposed in this paper allows citizen participation in the government process thus creating a more interactive governance process. The tendering mechanism becomes more transparent as the evaluation is done on the basis of the business logic deployed and not manually, thus decreasing biases. It also helps to develop a trust between the bidding organization and tendering organization as the winning organization cannot change the bid proposal once submitted, thus giving confidence to the tendering organization and helps in reducing fraud. The solution proposed in this paper is just a generic framework focused on tendering mechanism using blockchain.

[5] D. Mali, D. Mogaveera, P. Kitawat and M. Jawwad, "Blockchain-based e-Tendering System"

Introduction: The tendering process is generally used by governments and companies to produce goods or services from manufacturing companies or service providers. E-tendering being the mostly used procurement method, there are various security implications present. Blockchain technology can be used to solve this security implications as its heavily focuses on decentralization if information and secured by encryption and is integrated with undeniable block based architecture for transaction management. Current e-tendering system is not fair and open as information is not shared with all stakeholders. Apart from not being transparent security is also a major issue for these portals leading to fraud and manipulation of data stored int the centralized database. Therefore blockchain and smart contract can be used as a transparent, decentralized and secure tendering framework that will facilitate bidder oversight on portal functions and observe all the activities carried out by the tender portal.

Characteristics and Implementation: The tender smart contract is create using a factory contract that takes all the required information for creating a tender. These documents are linked to encrypted documents on the server. The process includes creating a tender, bidding on tender, bid evaluation and negotiation and finally winner selection publishing bids to the bidders.

Conclusion: When it comes to applications such as tender portals, where transparency and security are of foremost importance, traditional technologies and design patterns cannot be used as they put a threat to these requirements. As discussed earlier, there are many security requirements for a tendering framework that cannot be solved just by using a centralized tender portal for creating and bidding on the contracts. The security requirements and openness required from this type of application can only be solved by using fair, open, decentralized technology such as Blockchain and Smart Contracts.

[6] P. Manimaran and R. Dhanalakshmi, "Blockchain-Based Smart Contract for E-Bidding System"

Introduction: e-bidding systems have become widespread since the advent of the internet and mobile phones. In the electronic bidding systems, the seller will sell and item and many buyers will bid for that item and the highest bidder will get the item. One of the main issue with this e-bidding system is introduction of the third party mainly a company or set of companies which will develop and host either the website or smartphone app. The buyers and sellers have to trust this company because all bidding process will be handled by the company. The company can manipulate the bidding process if it wants. So to avoid the trust issues blockchain based electronic bidding system is introduced in this paper, where there is no need for third party. Smart contracts will handle all the bidding transactions, and blockchain is known for integrity this system makes sure that the integrity of the bidding process is preserved.

Characteristics and Implementation: For the bidding purpose only the public address of the bidder and sellers will be used throughout the process. No one will know the real identity of the buyer and the seller. The smart contract will contain the following functions blindAuction, bid, reveal, withdraw and auctionEnd. The system is implemented using Hyperledger fabric which is a permissioned blockchain. Hyperledger fabric can be used for open biddings where anyone can participate and confidential bidding where the bidding process should be known only to the participants.

Conclusion: The proposed blockchain based smart e-bidding system is secure and powerful compared to normal web based e-auction system. The blockchain technology brings in integrity and confidentiality to auction process which makes it possible to perform the auction even though some of the buyers are not trustworthy. This approach can be extended to other electronic-based systems such as electronic voting. Thus the blockchain technology is used to eliminate the middleman in the electronic bidding system. It eliminates the transaction fees given to the middleman because smart contract performs the functions of the middleman.

[7] Y. -H. Chen, S. -H. Chen and I. -C. Lin, "Blockchain based smart contract for bidding system"

Introduction: The E-auction, one of the popular e-commerce activities, allows bidders to directly bid the products over the Internet. As for sealed bid, the extra transaction cost is required for the intermediaries because the third-party is the important role between the buyers and the sellers help to trade both during the auction. To resolve the problems, the blockchain technology with low transaction cost is used to develop the smart contract of public bid and sealed bid. The smart contract, proposed in 1990 and implements via Ethereum platform, can ensure the bill secure, private, non-reputability and inalterability owing to all the transactions are recorded in the same but decentralized ledgers.

Characteristics and Implementation: For the testing of the contract they have created two blockchain accounts using Ethereum Wallet for testing and bidding transactions. In the miner, we adopt command-line and MinerGate to execute the data miner to get the coin for paying the transaction fee as shown. We can use the command-line to check the transaction status for the details of blocks in blockchain. In smart contract creation, three stages, namely writing, compiling, and announcing by using Solidity programming. The bytecode is generated by Solidity real-time compiler. The Solidity runtime is used to generate the Interface.

Conclusion: Therefore the paper provides e-auction mechanism based on blockchain to ensure electronic seals confidentiality, non-repudiation and immutability. In smart contracts for sealed orders, due to the complexity of the contract, the bidders and bidders come, say may call the wrong contract function. For example, the bidder inadvertently calls Reveal() to open all bids, so that the bidding must be terminated and re-arranged. For this purpose, we will set the authority judgment for different functions and will perform the function before first determine if the caller can perform this function.

CHAPTER 4

PROJECT REQUIREMENT SPECIFICATION

4.1 Purpose

A Tender management system is essential and ensures an efficient and effective process for the buyers who are wishing to source the contract which is nothing but a project either hosted by government or any other organization and the contractor wishing to handover the project to a legitimate buyer for the completion of the work. Blockchain based tender management system is proposed to eliminate the dependency on a third party and it also promotes decentralization.

4.2 Intended Audience

Government sectors (NIT, NMC, ZP, PWC), Private Industries, Software Industries, Construction Industries, Surveying, Educational institutions.

4.3 Intended Use

Tender management system using blockchain can be used as secure, efficient, and reliable tender management system because there is no involvement of third party therefore there is no dependency or a single point of failure as the third party will have a centralized server. Hence this system build using blockchain can be used without any dependency and runs on a public blockchain therefore all stakeholders have a clear view of the transaction done on the chain.

4.4 Scope

Project scope is simply a way to describe the work you are agreeing to deliver. It describes the project constrains or limitations and project assumptions.

Tender management system developed with blockchain integration is secure, efficient, reliable, and decentralized there the user can use the system without any trust on other parties in the system. As trustlessness is the core element of blockchain, crypto transactions and smart contracts. Trustless means the users don’t have to trust any intermediary or a third party that could operate between the bidders and the contractors in the tendering process. As a result the tendering process is done smoothly between the bidder and the contractor using smart contract. The smart contract is responsible for the proper automation of the tendering process.

The projects provides basic functionalities like adding a tender, bidding for a tender.

Tender will be added only by the contractor or admin.

The bidders will be able to submit or apply for a tender.

The transaction will be available on the public blockchain.

The bidders will get a list of available tenders.

4.5 PRODUCT FEATURES

The entire system is divided into 3 parts:

1. Add tender: The add tender function is responsible for adding new tenders to the list of all available tenders which the bidder can use to bid for the tender they want to. This function can clearly differentiate the users and the contractor. The contractor here is the one who is hosting the system intending to get bids for the proposed tenders. The function will only allow the contractor to add new tenders.

2. Bid for tender: This function enables the users of the bidders to bid for the tender they want to. The tenders available are in a list which contains all the tenders that are available for bidding. This function takes necessary information for the process of bidding it takes, name of the organization, organization email, organizations phone, proposed bid in string, and proposed bid in number. The function warns to fill all the fields mentioned above.

3. Smart contract: This is responsible for the validation of the two functions add tender and bid for tender. The smart contract makes sure only contractors will be able to add new tenders and it makes sure that a single organization cannot bid for a tender more than one, to prevent duplicate bids from the same organization.

4.6 USER CLASSES AND CHARACTERITICS

1. Normal Bidder: These are the non-technical users who are willing to bid for tenders. These users will interact with the user interface to select their desired tender to bid for. User has to fill out the details and submit the form for a successful bid.

4.7GENERAL CONSTRAINTS, ASSUMPTIONS AND DEPENDENCIES

1. Constraints: As of now the application works with only a good internet connection. It is limited to use only in a desktop with a browser compatible with any Ethereum wallet. The performance of the application depends on the chain that the smart contract is deployed or the token that will be used to interact with the system.

2. Assumptions: For the projects purpose it is being assumed that user knows English for providing text input while filling the form while bidding for a tender. It is also assumed that the user is familiar with basic usage of decentralized application, that is setting up the wallet, accepting the transaction and the gas info.

3. Dependencies:
   - reactjs is used to develop the front end.
   - Smart contract is developed using solidity programming language
   - Hardhat is used to compile, test and deploy the smart contract
   - Polygon Mumbai testnet is used for the smart contract testing
   - Test MATIC is required to interact with the smart contract
   - Browser compatible with Ethereum wallet
   - Smart contract and the front end is connected using libraries like ethers, walletconnect, and openzeppelin

4.8 FUNCTIONAL REQUIREMENTS

The functional requirement is a description of the service that the software must offer. It describes a software system or its components. A function is nothing but input to the software system, its behaviour and outputs. It can be specific functionality which defines what function a system is likely to perform. They are also called function specification.

Below are some of the functionalities which are expected to be performed by the system.

- The system should be able to add new tenders to the tenders list
- The system should be able to differentiate between the user and the contractor
- Add tender function should be allowed only to the contractor or the tender proposer
- The system must be able to list the available tenders in real time
- The system should be able to bid for a tender
- The system should make sure that all details are filled in the bidding form
- The system should be able to reject the duplicate bids

4.9 EXTERNAL INTERFACE REQUIREMENTS

External interface requirements define the software, hardware or database elements with which a system or a component must interact.

4.9.1 Hardware Requirements

- Minimum CPU or processor speed 1.3 GHz
- Minimum system memory or RAM 4GB

4.9.2 Software Requirements

- Solidity programming language for the development of smart contract
- Reactjs for the development of frontend
- Hardhat to compile, test, and deploy the smart contract
- Ethers library to connect frontend with the contract
- Browser with web3 compatibility
- Browser with an Ethereum wallet
- Test MATIC for interacting with the smart contract

4.10 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define the system attributes such as reliability, security, maintainability, useability, scalability, and security. They serve as restrictions or constraints on the design of the system.

Reliability: Software reliability can be defined as the probability of failure-free operation of a computer program in a specified environment for a specified time. The system must be able to operate without any failure or error.

Security: The system must be able to restrict any attacks as its important functionality uses blockchain technology the system must withstand data attacks.

Maintainability: The degree of effectiveness and efficiency with which product or system can be modified to improve it, correct it or adapt to changes in the environment. The current system can be improved to make it better or production ready by adding features.

Useability: It refers to the quality of the users experience when interacting with the system. It is about effectiveness, efficiency and overall satisfaction of the user. The system is very user friendly with minimum functions. So that there is no much complexity in using the system.

Scalability: The ability of the system to be able to grow or shrink to the changing demands. The system is dependent on the polygon network and is scalable up to the transaction capacity of the polygon network. Assuming only our application is running on the network.

CHAPTER 5

SYSTEM DESIGN

System design is the process of designing the elements of the system such as the architecture, modules and components, the different interfaces of that component and the data that passes through that system.

5.1 HIGH LEVEL DESIGN

The high level design is the general system of design. It refers to the overall design of the system and describes the overall architecture/ description of any application. It is also called system/macro-level design. It explains the architecture that would be used to develop a system. The architecture diagram provides an overview of the entire system, identifying the main components. This chapter covers the architecture and design decisions of the project.

5.1.1 Design goals

- The system should be able to handle multiple users at any point of time
- The systems interface should be simple and easy to use even for first time users
- The system should be responsive and efficient without any lag while navigating
- The system should display the available tenders

5.1.2 Use case diagram

The use case diagrams describe the high level functions and scope of a system. This diagram also identifies the interaction of the system and its actors. The use cases and actors in the use case diagrams describe what the system does and how the actors use it but not the way system operates internally.

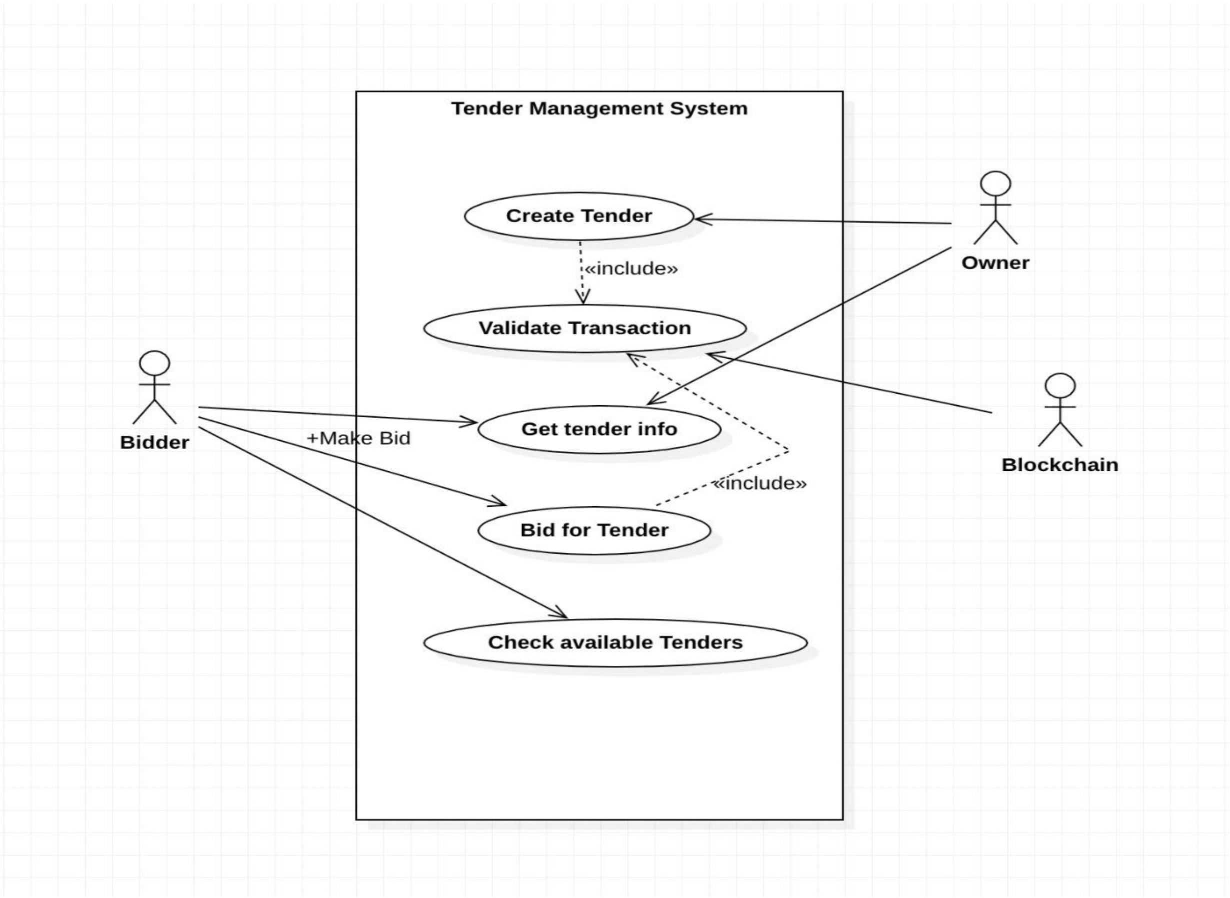


Figure 5.1.2 Use case diagram

- The bidder can use the function check available tenders to get the list of all available tenders.
- Bid for tender function is used to bid for a particular tender providing the required details

- Owner interacts with the system to add new tenders
- Owner can use the get tender info function to check the number of bids for a particular tender

5.1.3 Data flow diagram

Data flow diagram is a traditional visual representation of the information flow within a system. The DFD describes the operations through data movement. It helps in understanding the functioning of a system.

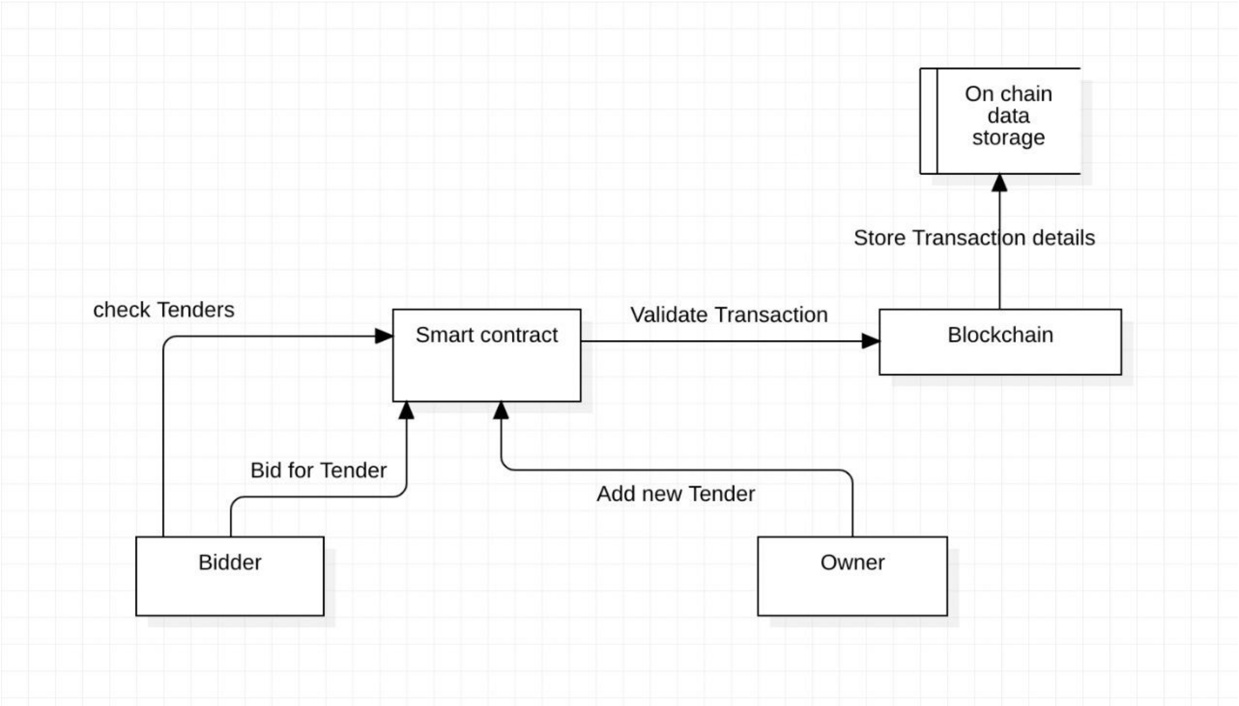


Figure 5.1.3 Data flow diagram

Owner must add tender so that bidder will be able to interact with the system.

When owner will add a new tender the data is sent to the smart contract after the verification of the data and after the transaction is successful, the transaction details are added to the blockchain.

When the bidder will bid for a tender the data is sent to the smart contract after the verification of the data and after the transaction is successful, the transaction details are added to the blockchain.

5.2 LOW LEVEL DESIGN

Low level design is a component level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture.

5.2.1 Architecture

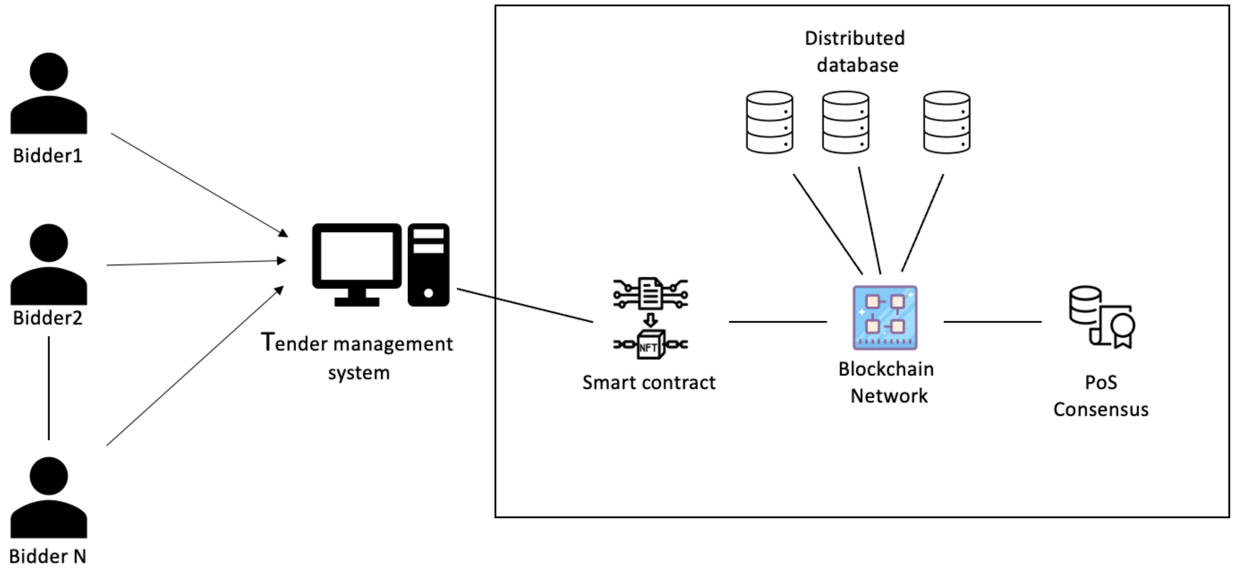


Figure 5.2.1 system architecture

CHAPTER 6

PROPOSED METHODOLOGY

The proposed system uses blockchain technology for the tender management system, which promotes decentralization which means all the stake holders can see the transactions that are happening through the system. Blockchain is used to eliminate the third party which acts the tender operator and has a centralized data base which is completely controlled by the third party. Therefore use of blockchain eliminates the third party to make the system more secure and efficient. The systems smart contract is deployed on the polygon Mumbai testnet for the development process, once the system is ready for production it can be later deployed on the polygon mainnet. The system has basic function which will be easy to use to new users as well. The system main functionality is to bid for a tender which will be added by the contractor(owner). The system is trustless as there is no involvement of the third party. To make sure that there are no duplicate bids the system will not allow a user to bid for a tender twice, therefor making sure that there are no duplicates. On the successful bidding for a tender the details about the bid are stored on the blockchain, the transaction record will be added on the blockchain as its public blockchain based therefore any one can see the transaction taking place on the system making the system completely decentralized.

CHAPTER 7

IMPLEMENTATION AND PSEUDOCODE

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;   //Must be included in every solidity code to inform about the version
address public owner;
string[] public tendersArray;   // array to store all the tenders which are available
constructor(){
owner = msg.sender;  // owner is one who deploys the smart contract to a network
}
struct ticker {
bool exists;
uint256 up;
string name;
string email;
string quotation;
uint256 price;
mapping(address => bool) Voters;
}
// a special data type which will store the information which the bidder has to fill
```

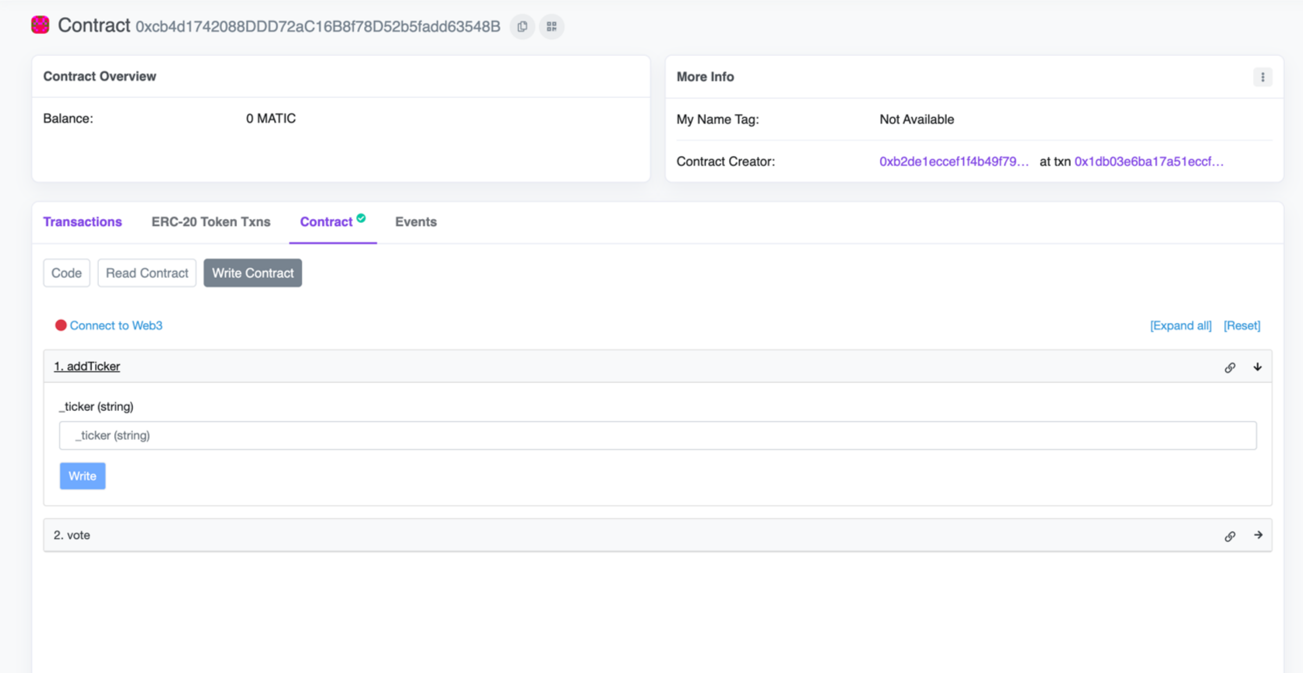
```
event tickerUpdated (
uint256 up,
string name,
string email,
string quotation,
uint256 price,
address voter,
string ticker
);
// whenever the contract is signed by the bidder it is a event so this data is set to the data which is filled by the bidder
mapping(string => ticker) private Tickers; // a mapping is created when a new tender is added.
function addTicker(string memory _ticker) public {
require(msg.sender == owner, "Only owner can create tenders");
ticker storage newTicker = Tickers[_ticker];
newTicker.exists = true;
tendersArray.push(_ticker);
}
// This function can be accessed only by the owner to add new tender which will be available for the bidder to bid.
```

```
function vote(string memory _ticker, string memory _name, string memory _email,
string memory _quotation, uint256 _price) public {
require(Tickers[_ticker].exists, "Can't vote on this tender");
require(!Tickers[_ticker].Voters[msg.sender], "You have already voted for this
project");
ticker storage t = Tickers[_ticker];
t.Voters[msg.sender] = true;
t.up++;
t.name = _name;
t.email = _email;
t.quotation = _quotation;
t.price = _price;
emit tickerUpdated(t.up, t.name, t.email, t.quotation, t.price, msg.sender, _ticker);
}
// The vote function takes all the inputs which are needed to be filled by the bidder
// It makes sure that the vote function is available for a bidder only once
// All the parameters in the struct are set in this function and then the function
// emit will store the data that is set in the function which will be stored in the
blockchain
```

```
function getVotes(string memory _ticker) public view returns(
uint256 up
){
require(Tickers[_ticker].exists, "No such project exists");
ticker storage t = Tickers[_ticker];
return(t.up);
}
// The getvotes function takes the tender name as input and then gives back the number
of bids that are made to that particular tender.
```

CHAPTER 8

RESULTS AND DISCUSSION



Contract 0xcb4d1742088DDD72aC16B8f78D52b5fadd63548B

Contract Overview

Balance: 0 MATIC

More Info

My Name Tag: Not Available

Contract Creator: 0xb2de1eccef1f4b49f79… at txn 0x1db03e6ba17a51eccf…

1. addTicker

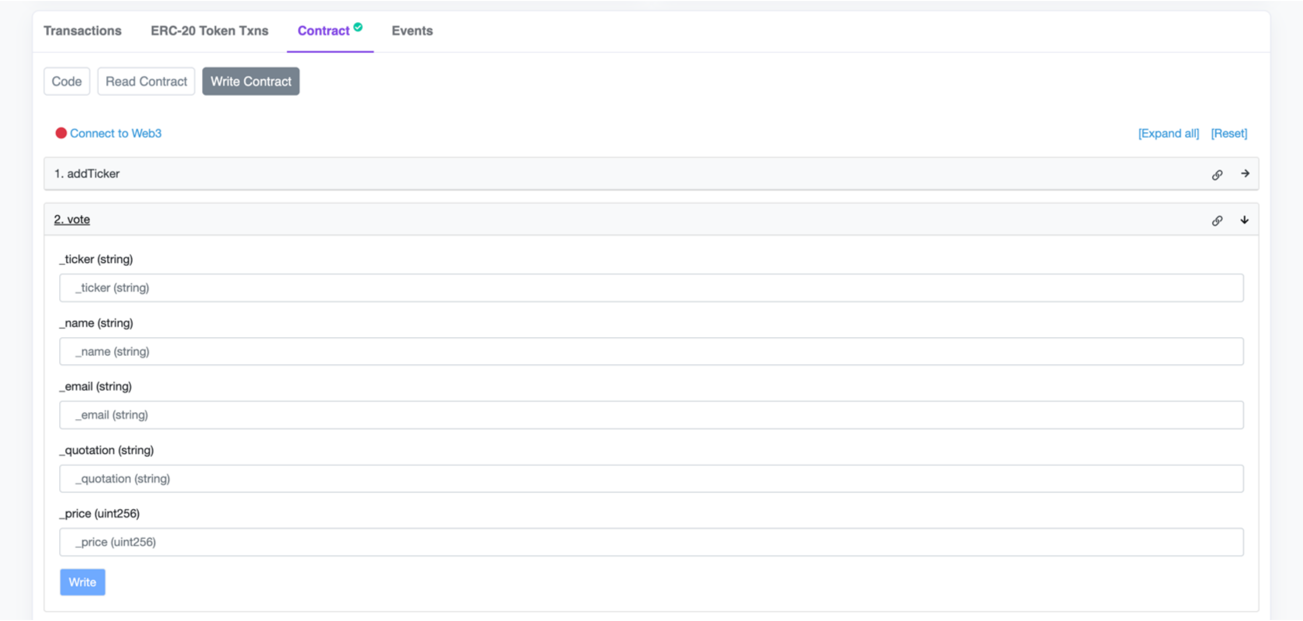
_ticker (string)

_ticker (string)

2. vote

Figure 8.1 addTicker function

This function is responsible for adding new tender to the tenders list. It takes a string as input that is name of the new tender.



1. addTicker

2. vote

_ticker (string)

_name (string)

_email (string)

_quotation (string)

_price (uint256)

Figure 8.2 Vote function

This is user function to add bid for a tender. It takes tender name, and bidding info and all the fields are required.

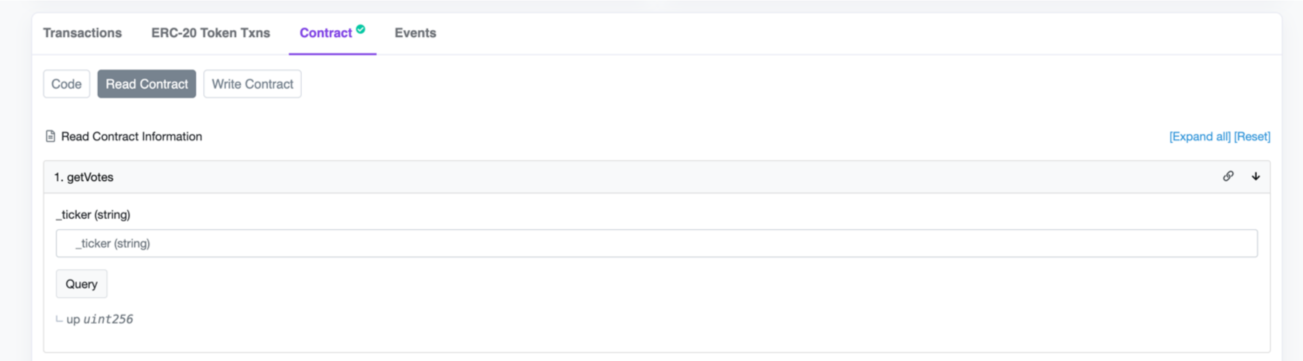


Figure 8.3 getVotes function

This function is responsible for giving out the total number of bids made for a particular tender. It takes a string that is tender name as input.

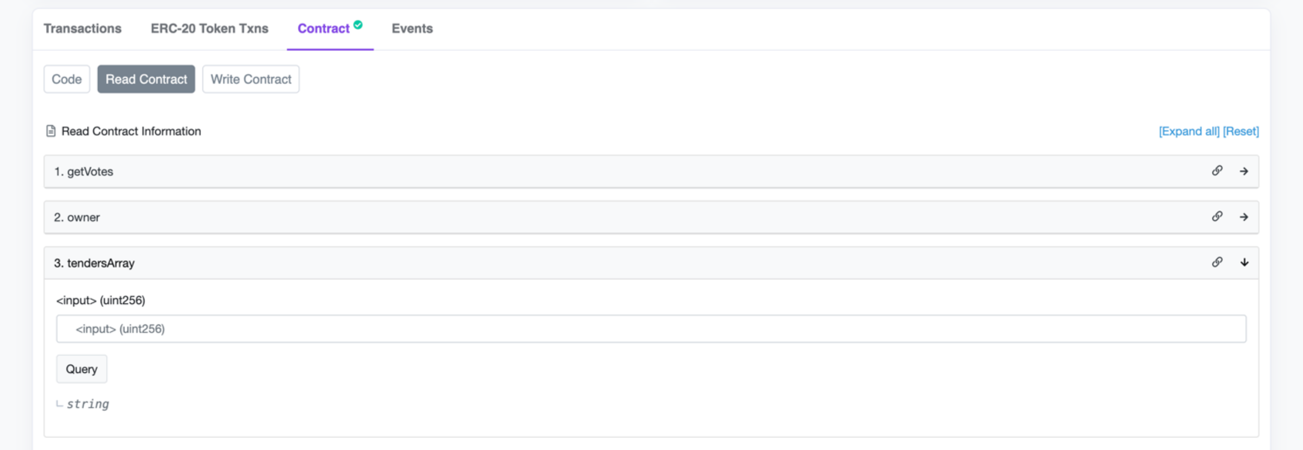
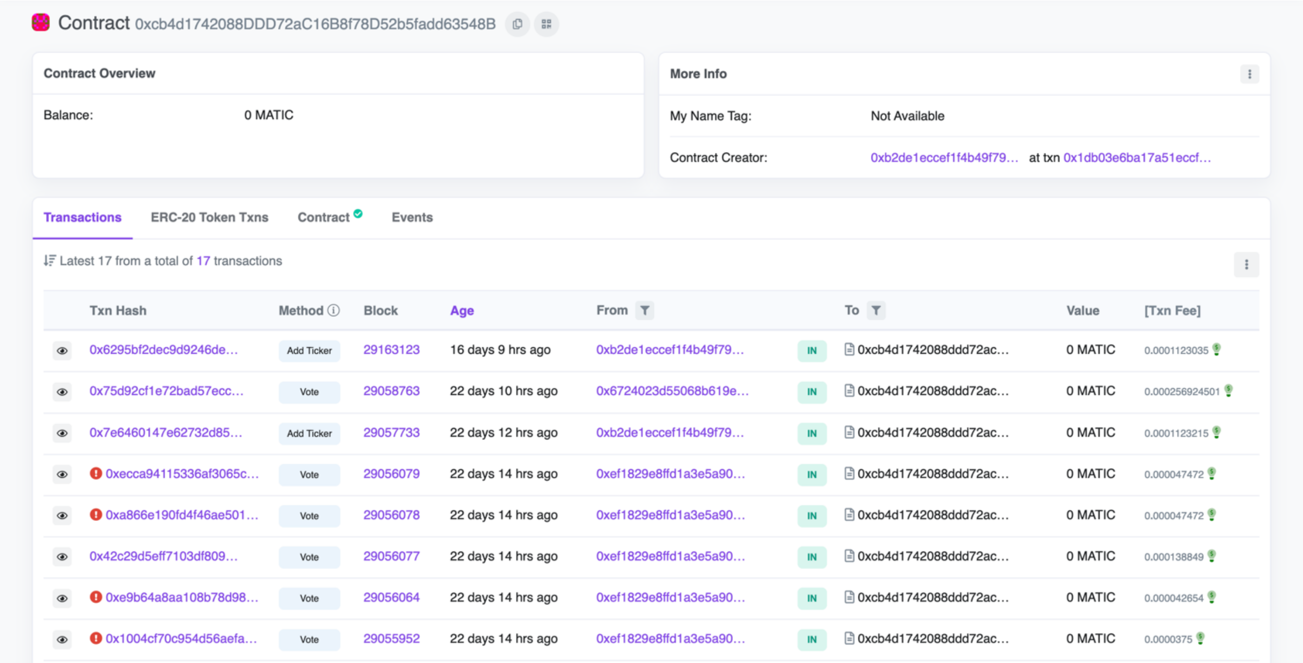


Figure 8.4 tendersList

This function acts as a list and the tenders can be accessed by providing number as an index in the given field.



Contract 0xcb4d1742088DDD72aC16B8f78D52b5fadd63548B

Contract Overview

Balance: 0 MATIC

More Info

My Name Tag: Not Available

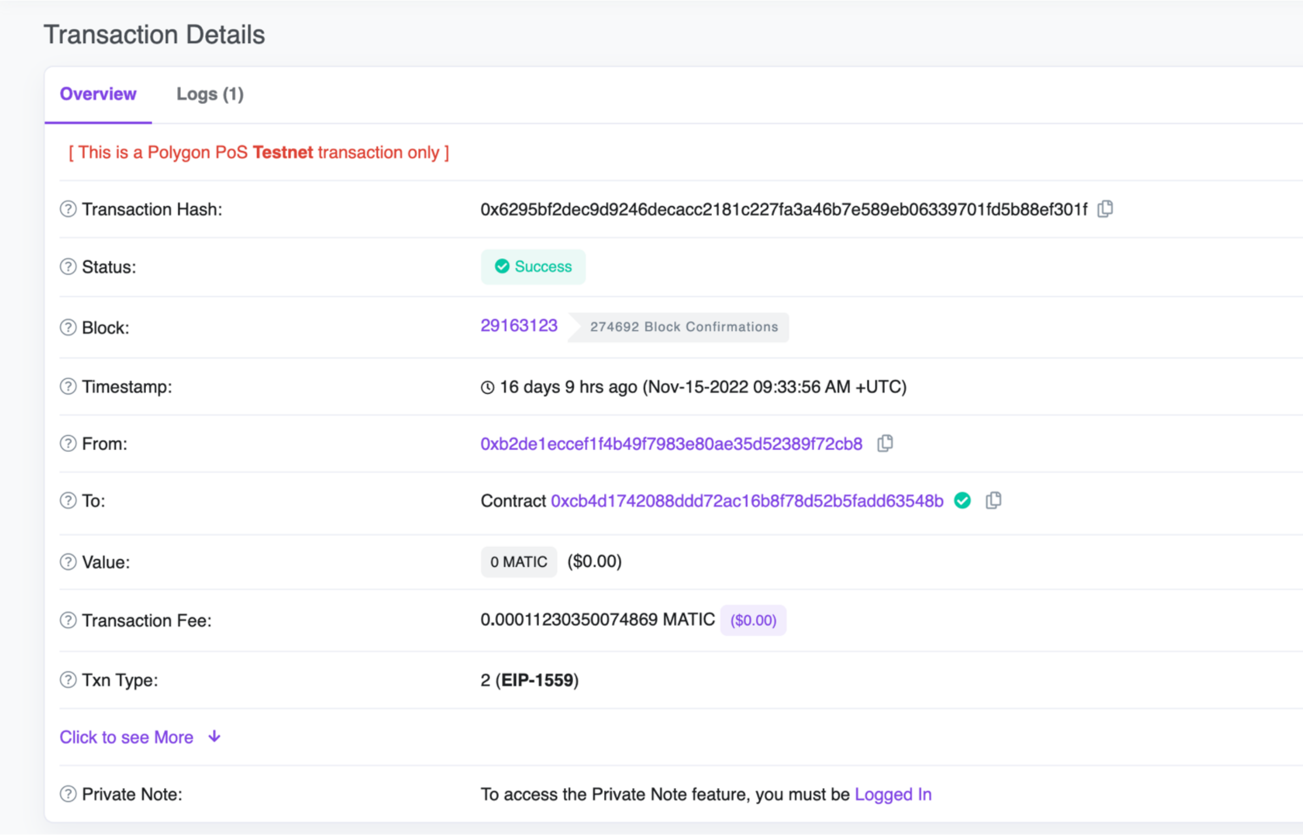
Contract Creator: 0xb2de1eccef1f4b49f79… at txn 0x1db03e6ba17a51eccf…

Latest 17 from a total of 17 transactions

Txn Hash	Method	Block	Age	From		To	Value	[Txn Fee]
0x6295bf2dec9d9246de…	Add Ticker	29163123	16 days 9 hrs ago	0xb2de1eccef1f4b49f79…	IN	0xcb4d1742088ddd72ac…	0 MATIC	0.0001123035
0x75d92cf1e72bad57ecc…	Vote	29058763	22 days 10 hrs ago	0x6724023d55068b619e…	IN	0xcb4d1742088ddd72ac…	0 MATIC	0.000256924501
0x7e6460147e62732d85…	Add Ticker	29057733	22 days 12 hrs ago	0xb2de1eccef1f4b49f79…	IN	0xcb4d1742088ddd72ac…	0 MATIC	0.0001123215
! 0xecca94115336af3065c…	Vote	29056079	22 days 14 hrs ago	0xef1829e8ffd1a3e5a90…	IN	0xcb4d1742088ddd72ac…	0 MATIC	0.000047472
! 0xa866e190fd4f46ae501…	Vote	29056078	22 days 14 hrs ago	0xef1829e8ffd1a3e5a90…	IN	0xcb4d1742088ddd72ac…	0 MATIC	0.000047472
0x42c29d5eff7103df809…	Vote	29056077	22 days 14 hrs ago	0xef1829e8ffd1a3e5a90…	IN	0xcb4d1742088ddd72ac…	0 MATIC	0.000138849
! 0xe9b64a8aa108b78d98…	Vote	29056064	22 days 14 hrs ago	0xef1829e8ffd1a3e5a90…	IN	0xcb4d1742088ddd72ac…	0 MATIC	0.000042654
! 0x1004cf70c954d56aefa…	Vote	29055952	22 days 14 hrs ago	0xef1829e8ffd1a3e5a90…	IN	0xcb4d1742088ddd72ac…	0 MATIC	0.0000375

Figure 8.5 all transactions

All activity that is done in the system that is the all the transactions done are stored on the blockchain.



Transaction Details

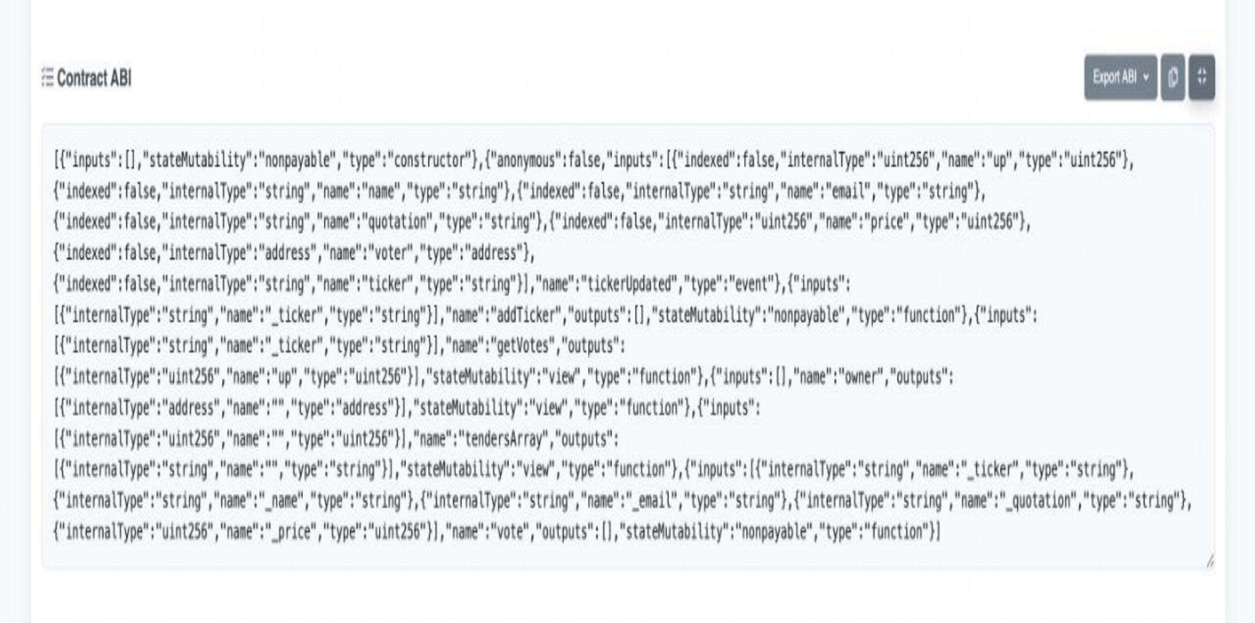
[This is a Polygon PoS **Testnet** transaction only]

ⓘ Transaction Hash:	0x6295bf2dec9d9246decacc2181c227fa3a46b7e589eb06339701fd5b88ef301f
ⓘ Status:	✓ Success
ⓘ Block:	29163123 274692 Block Confirmations
ⓘ Timestamp:	16 days 9 hrs ago (Nov-15-2022 09:33:56 AM +UTC)
ⓘ From:	0xb2de1eccef1f4b49f7983e80ae35d52389f72cb8
ⓘ To:	Contract 0xcb4d1742088ddd72ac16b8f78d52b5fadd63548b ✓
ⓘ Value:	0 MATIC ($0.00)
ⓘ Transaction Fee:	0.00011230350074869 MATIC ($0.00)
ⓘ Txn Type:	2 (**EIP-1559**)

ⓘ Private Note:	To access the Private Note feature, you must be Logged In

Figure 8.6 Transaction details

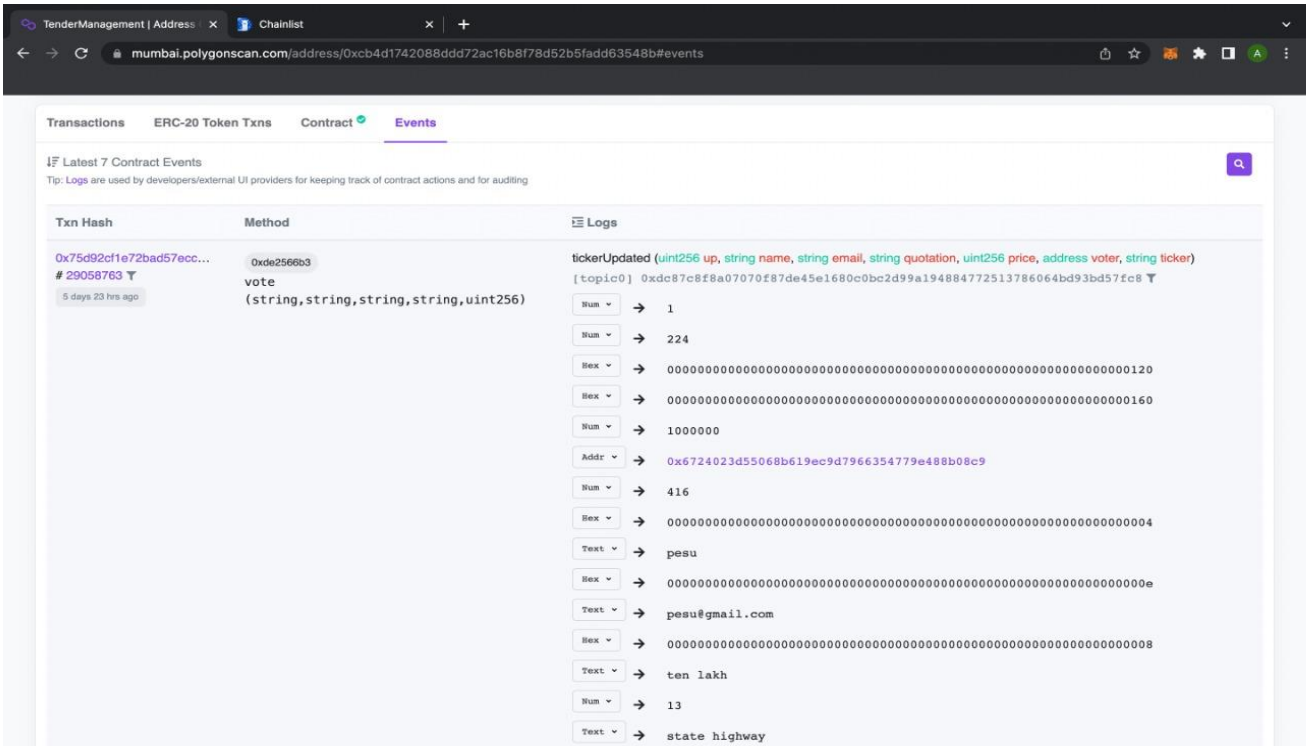
All the details of a particular transaction can be accessed in the network



```
[{"inputs":[],"stateMutability":"nonpayable","type":"constructor"},{"anonymous":false,"inputs":[{"indexed":false,"internalType":"uint256","name":"up","type":"uint256"},
{"indexed":false,"internalType":"string","name":"name","type":"string"},{"indexed":false,"internalType":"string","name":"email","type":"string"},
{"indexed":false,"internalType":"string","name":"quotation","type":"string"},{"indexed":false,"internalType":"uint256","name":"price","type":"uint256"},
{"indexed":false,"internalType":"address","name":"voter","type":"address"},
{"indexed":false,"internalType":"string","name":"ticker","type":"string"}],"name":"tickerUpdated","type":"event"},{"inputs":
[{"internalType":"string","name":"_ticker","type":"string"}],"name":"addTicker","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":
[{"internalType":"string","name":"_ticker","type":"string"}],"name":"getVotes","outputs":
[{"internalType":"uint256","name":"up","type":"uint256"}],"stateMutability":"view","type":"function"},{"inputs":[],"name":"owner","outputs":
[{"internalType":"address","name":"","type":"address"}],"stateMutability":"view","type":"function"},{"inputs":
[{"internalType":"uint256","name":"","type":"uint256"}],"name":"tendersArray","outputs":
[{"internalType":"string","name":"","type":"string"}],"stateMutability":"view","type":"function"},{"inputs":[{"internalType":"string","name":"_ticker","type":"string"},
{"internalType":"string","name":"_name","type":"string"},{"internalType":"string","name":"_email","type":"string"},{"internalType":"string","name":"_quotation","type":"string"},
{"internalType":"uint256","name":"_price","type":"uint256"}],"name":"vote","outputs":[],"stateMutability":"nonpayable","type":"function"}]
```

Figure 8.7 contract ABI

Contract ABI is used to get the information from the network to the frontend


TenderManagement | Address
Chainlist
mumbai.polygonscan.com/address/0xcb4d1742088ddd72ac16b8f78d52b5fadd63548b#events
Transactions
ERC-20 Token Txns
Contract
Events

Latest 7 Contract Events

Tip: Logs are used by developers/external UI providers for keeping track of contract actions and for auditing

Txn Hash	Method	Logs
0x75d92cf1e72bad57ecc... # 29058763 5 days 23 hrs ago	0xde2566b3 vote (string,string,string,string,uint256)	tickerUpdated (uint256 up, string name, string email, string quotation, uint256 price, address voter, string ticker) [topic0] 0xdc87c8f8a07070f87de45e1680c0bc2d99a194884772513786064bd93bd57fc8 Num → 1 Num → 224 Hex → 0000000000000000000000000000000000000000000000000000000000000120 Hex → 0000000000000000000000000000000000000000000000000000000000000160 Num → 1000000 Addr → 0x6724023d55068b619ec9d7966354779e488b08c9 Num → 416 Hex → 0000000000000000000000000000000000000000000000000000000000000004 Text → pesu Hex → 000000000000000000000000000000000000000000000000000000000000000e Text → pesu@gmail.com Hex → 0000000000000000000000000000000000000000000000000000000000000008 Text → ten lakh Num → 13 Text → state highway

Figure 8.8 Emitted details

After every successful bidding the details are emitted on the network to view.

CHAPTER 9

CONCLUSION AND FUTURE WORK

Therefore, the tender management system using blockchain provides decentralization and tamper proof data. As a tender management system needs it the most. It eliminates third party therefore only the contractor and the bidder take part in the process. The bidders can bid on the tenders they wish to bid and they need not worry about the data being manipulated. This Tender management system using blockchain can be developed further by separating the data storage area and using the blockchain only for the purpose of verification and other security reasons. By developing more functionalities the system will be able to replace the current tender management system.

REFERENCES

[1] Wang, D.; Zhao, J.; Mu, C. Research on Blockchain-Based E-Bidding System *Appl. Sci.* **2021**, *11*, 4011.

[2] Andrew, B.; Christopher, B.; Thomas, P. Digital Voting with the Use of Blockchain Technology.

[3] F. S. Hardwick, R. N. Akram and K. Markantonakis, "Fair and Transparent Blockchain Based Tendering Framework - A Step Towards Open Governance"

[4] Y. Goswami, A. Agrawal and A. Bhatia, "E-Governance: A Tendering Framework Using Blockchain With Active Participation of Citizens"

[5] D. Mali, D. Mogaveera, P. Kitawat and M. Jawwad, "Blockchain-based e-Tendering System"

[6] P. Manimaran and R. Dhanalakshmi, "Blockchain-Based Smart Contract for E-Bidding System"

[7] Y. -H. Chen, S. -H. Chen and I. -C. Lin, "Blockchain based smart contract for bidding system"

[8] V. B. Khanapuri, S. Nayak, P. Soni, S. Sharma, and M. Soni, “Framework to overcome challenges of implementation of e-procurement in indian context,” in International Conference on Technology and Business Management, pp. 28–30, 2011.

[9] D. Panda, G. Sahu, P. Gupta, et al., “Promoting transparency and efficiency in public procurement: E-procurement initiatives by government of india,” in 7th International Conference on E-government (ICEG), 2010.

[10] “Blockchain.” https://en.wikipedia.org/wiki/Blockchain. Accessed: 2020-7-19

[11] “hyperledger-fabric.” https://hyperledger-fabric.readthedocs.io. Accessed: 2020-7-22.

[12] O. Pal and S. Singh, “Blockchain technology and it’s applications in e-governance services,”

[13] M. K. Mustafa and S. Waheed, “A governance framework with permissioned blockchain for the transparency in e-tendering process,” International Journal of Advanced Technology and Engineering Exploration, vol. 6, no. 61, pp. 274–280, 2019.

[14] “Members of Parliament Local Area Development Scheme.” https://en.wikipedia.org/wiki/Members of Parliament Local Area Development Scheme. Accessed: 2020-7-20.

[15] Y. Chen, S. Chen and I. Lin, "Blockchain based smart contract for bidding system," 2018 IEEE International Conference on Applied System Invention (ICASI), Chiba, 2018, pp. 208-211.

[16] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008

[17] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, 2017, pp. 557-564.

[18] G. Cao and J. Chen, "Practical Electronic Auction Scheme Based on Untrusted Third-Party," 2013 International Conference on Computational and Information Sciences, Shiyang, 2013, pp. 493-496.

[19] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger", Ethereum Project Yellow Paper, vol. 151, pp. 1-32, 2014.

APPENDIX: DEFINITIONS AND ACRONYMS

Ethereum : It is a **decentralized open source blockchain** featuring smart contract functionality.

Polygon blockchain: It is an Indian blockchain platform. It aims to create a multi-chain blockchain ecosystem compatible with Ethereum. As with Ethereum ,it uses consensus model. Polygon’s native token is MATIC.

Ethereum Virtual machine : The Ethereum Virtual Machine or EVM is a piece of software that executes smart contracts and computes the state of the Ethereum network after each new block is added to the chain. The EVM sits on top of Ethereum's hardware and node network layer.

Smart contract : A smart contract is a **computer program** or a transaction protocol that is intended to automatically execute, control or document legally-relevant events and actions according to the terms of a contract or an agreement.

Meta mask : Meta Mask is an extension for accessing Ethereum enabled distributed applications, or "Dapps" in your normal Chrome browser. The extension injects the Ethereum web3 API into every website's JavaScript context, so that dapps can read from the blockchain.

HLC: Hyper ledger fabric.