International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-2, December 2019

An Efficient Technique for Managing user Information using Ethereum Network

Koushik Akkinapalli, Sasibhushana Rao Pappu, Ramesh Babu Vankara, Manikanta Vella

Abstract*: Due to enormous increase in development of technology maintenance of user's information has become major concern. However, user's information has been effectively maintained by the third parties but the problems with the current system are cost has been increased for maintaining information, lack of transparency and fairness, overuse of power and nature of opaqueness in the current system. So distributed system/ledger i.e., Blockchain came into existence to solve the problems encountered with the current system. The advantages of the proposed distributed system are cost is negligible by eliminating the need of third parties, effectively enhancing the execution speed of transactions and facilitating its reconciliation, maintaining transparency thereby ensuring integrity of user's information. In this paper, we introduce Ethereum based blockchain network for maintaining user's information through smart contracts. Experimental results shows that how effectively user's information has been maintained through blockchain based networks.

***Keywords* : Blockchain, Distributed ledger, Ethereum, Smart contracts.**

I. INTRODUCTION

In traditional systems maintenance of user's information has been important criteria for many organizations but the main problems with the previous models are tremendous increase of transaction costs, lack of transparency and possible misuse of information. So, in order to overcome the problems with the traditional centralized systems, Blockchain a distributed system/ledger has been evolved. Blockchain is a distributed information repository system/ledger of records which prevents unauthorized users to access the ledger. However information will be stored in block by block manner where each block is identified by its secure cryptographic signature. Each record will be validated by the miners before storing any entry in the new block and every information stored in the ledger is immutable (not editable).In comparison with traditional databases, blockchain can perform only insertion operation,

Revised Manuscript Received on December 30, 2019.
***** Correspondence Author
Koushik Akkinapalli*, Department of CSE, Raghu Institute of Technology, Visakhapatnam, India,
Sasibhushana Rao Pappu, Department of CSE, Aditya Institute of Technology and Management, Tekkali, India**,**
Ramesh Babu Vankara, Department of CSE, Raghu Institute of Technology, Visakhapatnam, India,
Manikanta Vella, Department of CSE, Raghu Institute of Technology, Visakhapatnam, India,

blocks are replicated on every node in the network, nodes in the network must agree on the result of obtained transactions i.e., achieving consensus and everyone can also validate the transactions participating in the network In blockchain based networks two keys i.e, public & private keys are maintained by every node in the network where public key can be shared with everyone in the network but the private key is kept in secure manner. In these networks, in order to perform transactions with the other node first transaction is signed with the other node's public key and signature is applied on the signed block, then to obtain the given transaction it must be signed with nodes' private key. Here, signature is unique for each and every transaction and keys are stored in secure wallet where nodes from time to time can continuously monitor their wallet by checking their current balance. Here, miners verify transactions by solving complex mathematical puzzles ensuring proof of work and also holds the copy of the ledger where they confirm that current node of the transaction possess the funds which are being transferred across the network and they also get reward for solving the puzzle. There are three important components which achieve proof of work are nonce, hash and transaction. Transactions are ordered by the secure system by placing them in blocks and are linked to each other forming a chain. To summarize, major properties of transaction are it is secured by consensus algorithm, recorded in the blocks by the miners, encryption of transactions by asymmetric algorithms, propagation among several nodes in the network and storage of information in a secure ledger. The next section of paper deals with Ethereum and its characteristics, smart contracts and solidity scripting language. The next further section deals with implementation and results and finally the conclusion of the paper.

II. ETHEREUM

In present distributed blockchain platforms, it is used for building any decentralized applications in an effective manner. It mainly focused on key areas such as issuance of currency, building applications useful for the organizations and smart contracts. The main intention behind the introduction of Ethereum decentralized platform was the development of smart contracts for blockchain based networks. The main properties of Ethereum networks are Ethash used for achieving consensus in the network and on an average blocks will be mined for 12-15 seconds Smart contracts are the main building blocks of the programs written exclusively for ethereum networks. It is generally a protocol which automatically executes on blockchain platforms by ethereum virtual machine (EVM) when specific conditions are met.

The advantages of smart contracts are reducing service and administration costs, automation of applications in an effective manner and efficient generation of reports. EVM enables provision of developing the applications by using smart contract based languages which are generally compiled into bytecode. In ethereum, for every contract that is being executed on blockchain platform, several computations are internally performed and tokens called ethers are consumed inside the network which are referred to as gas. Here the cost for the given transaction is determined by the price of the gas times the amount of gas used. The main difference between ethereum networks and bitcoins are former maintains the state of the transactions

In these networks, the different steps performed by the miner for validating the transactions are

1) Determine the transactions processed in the given network
2) Form the blocks called Uncles or owners
3) Get the reward after successful mining of the block
4) Finally block is written to the chain and state is computed as valid
5) Here, reward is given to each block successfully mined which have been set to 3 ethers

The different technologies that supports ethereum networks are swarm (used for effective storage of data), Whisper (used for exchange of messages between the nodes) and EVM (for achieving consensus) They are mainly used for building decentralized applications by using several components such as browsers, frameworks, different libraries, various front end tools and ethereum clients. Generally, ethereum clients performs the following actions such as forming an interaction with the given network, writing the contracts which are converted into bytecode via compiler and deployed into blockchain, implementing different protocols, replicating the blocks across the networks, managing ethereum accounts which are used for sending/receiving the transactions using ethereum accounts. Generally Ethereum based applications are also executed in browser using popular metamask plugin for chrome which uses infura service. The different libraries used for building dapps are Web3.js(javascript), Web3(java) and Nethereum(c#.net).Solidity a popular scripting language used for writing smart contracts in ethereum networks

III. SOLIDITY

It is high level contract oriented scripting language which is similar to classes and supports several concepts like inheritance, various category of libraries etc which compiles the given set of instructions into bytecode which are being interpreted by EVM. Solidity compiler produces bytecode and ABI(application binary interface) after successful compilation of contract file. Generally solidity file (.sol file) consists of several sections such as

1) Version containing pragma and include directives
2) Contracts containing declarations of state variables, functions and its modifiers, events, struct and enum types.

State variables hold values which are stored in permanent contract storage. The different types of state variables are

1) Value types (strings, integers, booleans, fixed, address consists of 20-byte ethereum account address which provides the base for all the contracts, enum types i.e., user defined types which are converted explicitly to/from integer types)
2) Function types (internal/external functions where internal functions are invoked inside the contract file and external functions are invoked through API calls consisting of address and signature of the function). The different access specifiers for the function types are
   a) public (any contracts can invoke the function)
   b) private (only particular contract can invoke the function)
   c) view (function which doesn't change contract data)
   d) pure (function will not modify or access contract data)
   e) payable (ether will be propagated along the network when someone invokes the function)
3) Reference types (fixed and dynamic array, mapping used to map key to given value, struct which holds different types). The location of data may be memory(temporary) or storage(permanent)

Functions contains block of the code which are executed inside EVM and they also contain modifiers which modifies behavior of functions by checking the conditions before executing the body of the function. The different operators supported are arithmetic, bitwise, logical and increment operators.

The different global variables available across the network are data, sender, sig and value which are accessed using msg object. Error handling functions supported in contract file are

1) assert (test for internal errors if any inside the contract file)
2) require (testing for valid inputs or checking whether the given conditions are met or not) .

The structure of the contract file is shown in the figure below

```
//solidity version

contract contractname

{

  //state variables declaration;

  //function definition consisting of setters and getters for the given state variables/ modifiers

  //structure of function definition

  //function functionname(<parameter types>) {internal|external} [pure|payable|view] [returns <return
type>]

  {

    //body

  }

}
```

Fig. 1. Structure of Contract file

IV. IMPLEMENTATION AND RESULTS

The paper has been implemented on Windows 10, AMD processor and also Ganache blockchain is used for testing smart contracts deployed on the chain through specified endpoint. Ganache blockchain test accounts are shown in the figure below

Fig. 2. Blockchain Test Accounts

Now we have used Remix-EthereumIDE browser for writing smart contracts. Here we prepared contract for user consisting of following details such as name, email address, contact number and address shown in the figure below

```
pragma solidity >=0.4.0 <=0.6.0;

contract UserInformation{
   string public _name; //state variable
   string public _emailAddress;
   string public _address;
   address public owner;
   uint256 public _contactNo;

   /*Called with the contract is deployed and initializes the value*/
   constructor() public {
       _name="koushik";
       _emailAddress="koushik@gmail.com";
       _address="Mvp,Visakhapatnam";
       owner=msg.sender;
       _contactNo=9579579000;
   }
   // Get Function
   function get_name() public view returns(string memory){
       return _name;
   }
   // Set Function
   function set_name(string memory name) public {
       if(owner==msg.sender)
       _name=name;
       else
       {
           revert("owner only can update the name");
       }
   }
```

Fig. 3. User Contract

Compiling contract

After writing the contract, plugin manager must be used for enabling the plugins such as solidity compiler (compiling contracts) and deploy and run transactions (deploying contracts on the blockchain network). Now once contract is compiled successfully ABI and bytecode are generated shown below


```
[
    {
        "constant": true,
        "inputs": [],
        "name": "get_address",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "payable": false,
        "stateMutability": "view",
        "type": "function"
    },
    {
        "constant": true,
        "inputs": [],
        "name": "_address",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
```


Fig. 4. Application Binary Interface

Application Binary Interface

It mainly specifies the structure of the contract in the form of key-value pair

Bytecode

It converts specified contract into machine readable format to be interpreted by EVM.

Deploying the contract

Once contract is compiled successfully, then contract must be deployed on the ganache blockchain platform through Web3 provider environment selected and the end point must be specified for communication with platform. Once connection is established successfully, several accounts retrieved from ganache are shown in the figure below

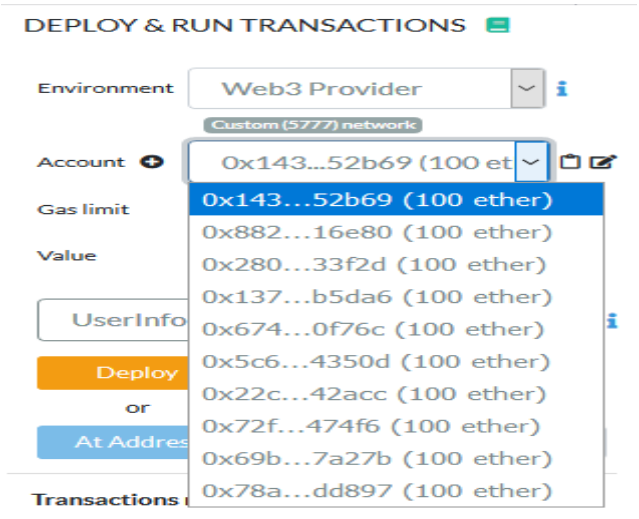


Fig. 5. Ganache Accounts displayed in Remix-IDE

Once Deploy button is clicked, contract address is generated for the selected account and blocks are created shown in the figure below

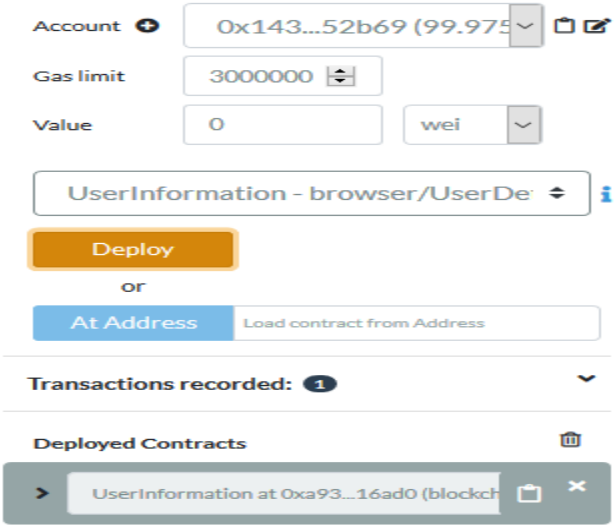


Fig. 6. Contract Address

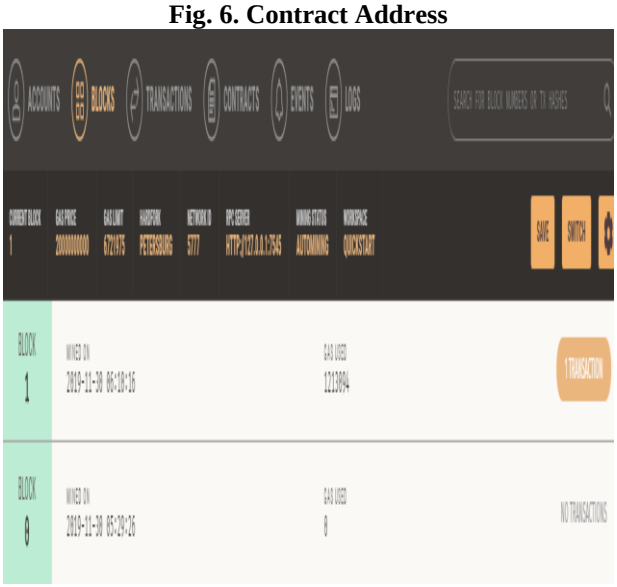


Fig. 7. Created Blocks after deployment

Now click on any one of the block to get its complete details

Fig. 8. Block information for Block1

Once contract is deployed ethers are deducted from current balance shown below

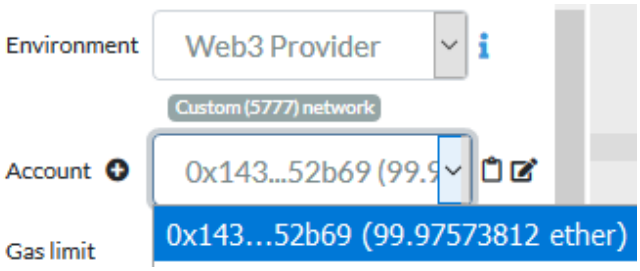


Fig. 9. Account Balance after contract creation

Now click on deployed contracts to get the information about the user details

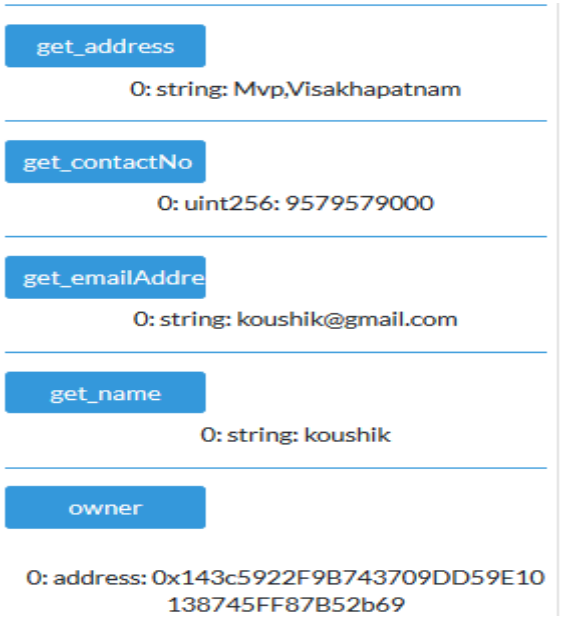


Fig. 10. User Contract Output

Now update data of the user for any of fields (for example contactNo and emailAddress) shown below

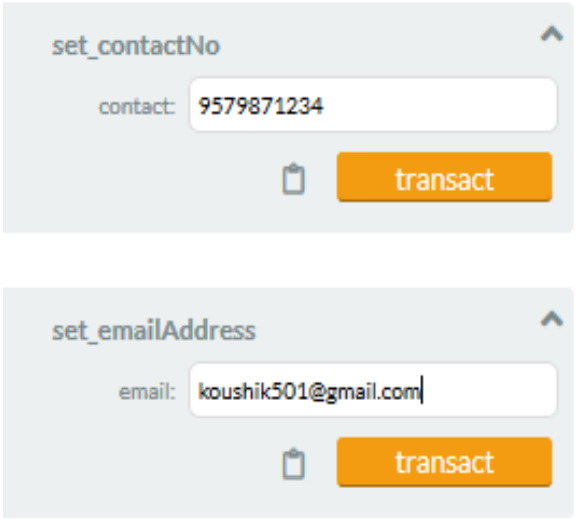


Fig. 11. Updation of User data

After clicking on transact button the user information will be updated and new blocks will be created after successful updating which is shown below

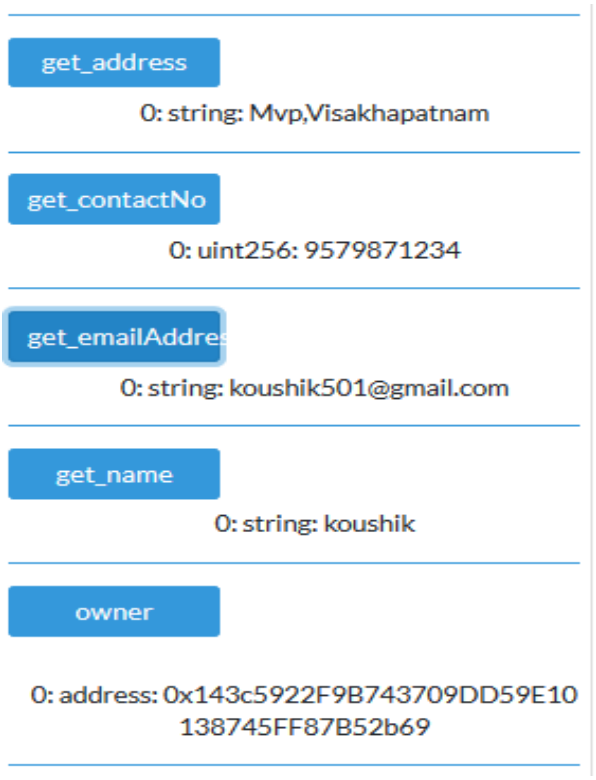
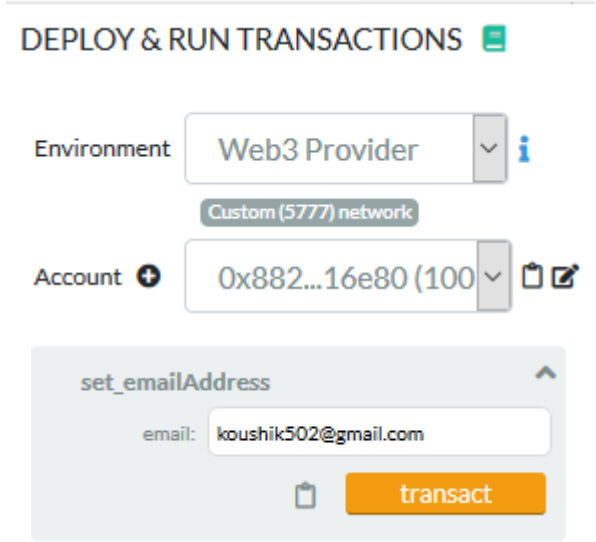


Fig. 12. User Contract Output after updation

Fig. 13. Formation of Blocks after updation

Now try to change the given account to some other account address and try to perform updation exception will be generated since other account can't perform updation in this way integrity of data can be assured shown below



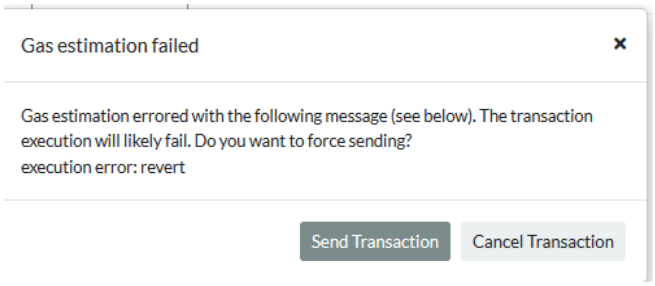


Fig. 15. Exception generated after updation of data

V. CONCLUSION

Blockchain has solved many of the problems encountered with traditional models by eliminating the need for third paties, making transaction costs negligible and maintained transparency by applying digital signatures and also ensuring their information will be maintained in secure ledger.

In this paper we introduced Ethereum a blockchain platform which experimentally proved that user information is managed in an effective manner.

REFERENCES

1. Zibin Zheng, Shaon Xie, Hong-Ning Dai, "An overview of Blockchain Technology: Architecture, Consensus and Future Trends," 6th IEEE International Congress on Big Data, June 2017
2. Alia Alammary, Samah Alhazmi, Marwah, Almasri and Saira Gillani, "Blockchain-Based Applications in Education: A Systematic Review", Appl.Sciences, June 2019.
3. https://etherscan.io
4. https://ethplorer.io
5. https://consensys.net/developers/
6. https://metamask.io
7. Shuai Wang, Liwei Ouyang, Yong Yuan, Xiaochun Ni, Xuan Han, Fei-Yue Wang, "Blokchain-Enabled Smart Contracts: Architecture, Applications and Future Trends", IEEE Transactions on Systems, Man and Cybernetics: Systems, February 2019.
8. Maximillan Wohrer, Uwe Zdun, "Smart contracts: security patterns in the ethereum ecosystem and solidity", IEEE, March 2018
9. https://ethgasstation.info
10. https://ethereum-magicians.org/t/standardize-url-format-for-web3-browsers/2422
11. Dejan Vujicic, Djana Jagodic, Sinisa Randic,"Blockchain technology, bitcoin and Ethereum: A brief overview" IEEE, 2018.
12. https://blockgeeks.com/guides/7-of-the-best-ethereum-wallets/
13. https://andersbrownworth.com/blockchain/
14. Sara Rouhani, Ralph Deters, "Performance Analysis of ethereum transactions in private blockchain",IEEE,2018.
15. https://ethereum.org/
16. Peter Hegedus, "Towards Analyzing the Complexity Landscape of Solidity Based Ethereum Smart Contracts", 2019.
17. https://solidity.readthedocs.io/
18. Debasis Bhattacharya, Mario Canul,Saxon knight, Mohammad Q.Azhar,"Programming Smart Contracts in Ethereum Blockchain using Solidity", ACM, February 2019.
19. https://www.trufflesuite.com/
20. Alper Alimoglu, Can Ozturan, "Design of a Smart Contract Based Autonomous Organiization for Sustainable Software",IEEE,2017.

AUTHORS PROFILE

Koushik Akkinapalli, is working as an Assistant Professor in Department of Computer Science & Engineering at Raghu Institute of Technology(Autonomous), Visakhapatnam. He has total of 7 years teaching experience and 1 year industry experience. He is currently pursuing PhD in Cloud Computing at JNTU Kakinada. He published 3 reputed International journals and having ISTE life time membership. He got outstanding performance award from TCS and qualified in UGC-NET.

Sasibhushana Rao Pappu, is working as an Assistant Professor in Department of Computer Science & Engineering at Aditya Institute of Technology and Management (Autonomous), Tekkali. He has total of 7 years teaching experience and currently pursuing PhD in Computer Networks at JNTU Kakinada. He has published 2 reputed International journals and having CSI membership. He has qualified in UGC-NET, GATE and AP-SET.

Ramesh Babu Vankara, is working as an Assistant Professor in Department of Computer Science & Engineering at Raghu Institute of Technology (Autonomous), Visakhapatnam. He has total of 3 years teaching experience and having two reputed memberships(IAENG and SDIWC). He has published 3 reputed International journals and attended 2 workshops.

Manikanta Vella, is working as an Assistant Professor in Department of Computer Science & Engineering at Raghu Institute of Technology(Autonomous), Visakhapatnam. He has total of 1.5 years teaching experience and having two reputed memberships(IAENG and theIRED). He has published 1 reputed International journals and attended 3 workshops