Smart Contracts: Automated Stipulations on Blockchain

Ms. Vruddhi Mehta
Computer engineering
SVKM's Shri Bhagubhai Mafatlal Polytechnic
Mumbai, India
vruddhimehta12@gmail.com

Ms. Sakshi More
Computer engineering
SVKM's Shri Bhagubhai Mafatlal Polytechnic
Mumbai, India
sakshimore1188@gmail.com

Abstract-**Contracts, a set of legally negotiated rules between the transacting parties are often the prime cause of legal as well as business disputes. Due to this discord, they are generally viewed with contempt. The need to revolutionize contracts has been much felt by the attorneys and the business professionals, so as to avoid these daedal contract conflicts. This resulted in the advent of smart contracts that was led by Blockchain technology. It is a blend of legalese from lawyers and computer code. A smart contract is a versatile system capable of facilitating, automating and enforcing an agreement (i.e. contract).In this paper, we preview as well as analyze smart contract based on blockchain technology for a decentralized system.**

Keywords: Blockchain, Ethereum, Proof-Of-Work Algorithm.

I. INTRODUCTION

Smart contracts are virtual agreements which are neither physical nor computer-based documents. These computer programs, also called blockchain contracts, are technology-derived solutions; they are the product of computer programming and legalese. The agenda is to enable formality, trust and cost associated with the traditional route. Nick Szabo, the originator of the smart contracts concept, coined the concept of incorporating contract terms into computer hardware and software by describing a car lien. Without the conventional legal system, if the owner fails to make payments on the loan secured by the car, the lender must go through the process of repossessing the car. By using a smart contract to facilitate a hardware and software function in the car if the owner fails to make payments, the lender can make it impossible for the owner to start the car. Once the loan has been completely paid off, a new function is automatically added that disables the previous function in smart contracts [1]. What makes smart contracts unique is that they involve the automation of contract formation as well as the execution of the contract's terms, running on the blockchain. Blockchain technology can revolutionize the vision of an "any-to-any" marketplace into reality. The idea of smart contracts along with the terms and conditions both parties can specify assures trust in the enforceability of the contract and the identity of the counter party [2].

The statistic shows that 22 million-plus cases are pending in the Indian courts out of which 6 million cases have dragged on for more than 5 years. Blockchain has garnered recognition from various sectors. As the World Bank accepted blockchain, by launching its very own blockchain development laboratory. On the other hand, Blockchain is foreseen as a supposed solution by the Telecom Company to revolutionize the economy of African nations. Blockchain has also ventured into the Indian market by announcing its partnership with Unocoin which is one of the largest crypto-currency platforms. When we look at smart contracts, these can be executed within minutes. People shall always opt for the later. This paper will give an overall view of the technology [3]. In this paper, Section II gives a brief idea about blockchain and its functionality, Section III discusses smart contract and Section IV ideates Solidity – a smart contract language for Ethereum Virtual Machine.

II. BLOCKCHAIN

Blockchain a distributed system in terms of ledger of facts, that vary on term basis from monetary transactions to technical codes aped across several computers in a peer-to-peer network. Transactions of facts on network among the nodes take place under scrutiny and the communication within network integrates the principles of security and integrity, taking the advantage of cryptography and Proof-of-Work Algorithm. When a node is transacting a fact i.e. it wants to add a fact to the ledger, a consensus is formed in the network to determine where this fact should emerge in the ledger; this consensus is called a block [6]. Blocks containing metadata chained by hash values are used for reference to the previous block that is like page numbers which can be used to refer to the previous pages. The blockchain is particularly valuable at increasing the level of trust among network participants. Because every transaction builds on every other transaction, any corruption is readily apparent, and everyone is made aware of it. This self-policing can mitigate the need to depend on the current level of legal or government safeguards and sanctions to monitor and control the flow of transactions [7].

The Figure 3.1 shows the working diagram of a blockchain system. We have explained the detailed working of blockchain to enhance the understanding of smart contracts using fact as a money transaction.

978-1-5386-2238-4/18/$31.00 ©2018 IEEE

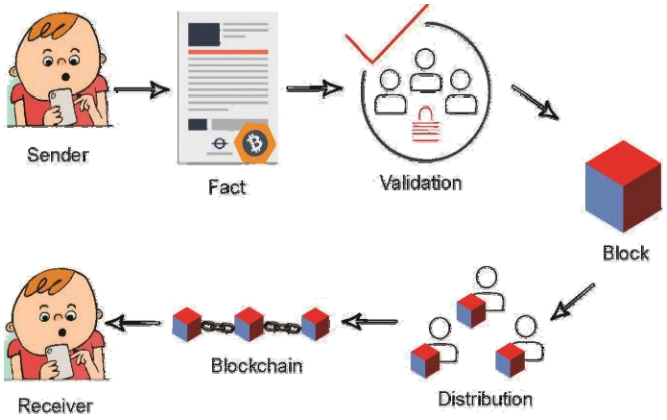

Figure 3.1 Working of Blockchain.

A. Sender-The transaction is initiated using a fact and for this purpose, the sender must have a wallet. This wallet is software installed on the participants' computer or mobile. To begin the process, the transaction is requested and by this process user A conducts a cross-border money transfer.

B. Validation-All the computers attached to the network simultaneously validates the transaction request as well as the user's status. Transactions are accepted only if they are valid.

C. Block-After the verification, the transaction is added to the block. So these transactions are collected and combined in a block.

D. Distribution-Once the node finds the Proof- Of –Work, the transaction is propagated and advertised to other nodes (computers) via the peer to peer network.

E. Blockchain-Nodes create the next block in the chain simultaneously as they accept the block, the previous block hash links the blocks together. These blocks that form the chain, provides a transparent and indelible record of transactions.

F. Receiver-Ownership of blocks that comprises of transactions gets transferred to the targeted account (digital address) of person B. Person B receives the funds [14].

A blockchain smart contract is in the form of transaction or block content on the blockchain network. The immutability of blockchain inhibits a trust on the contracts which enables a person to transact freely with an unknown party. Blockchain contracts contain the same amount of details as the conventional ones, but they are a notch up. They can perform tasks such as negotiating, monitoring and dynamic tracking of supply chains without manual effort. Smart contracts using blockchain technology eradicates the extra efforts, cost and the mediatory delay [4] [12].

E Karafiloski and A Mishev, the author of *"Blockchain Solutions for Big Data Challenges* "describes the concept of blockchain in a similar way. They state that blockchain acts as an opposite solution to every system based on client-server architecture. In their paper they mentioned blockchain as a database to store all transactions. They describe the process as; the nodes receiving the transactions store them onto their transactional pools. Predefined checks are run to validate the structure and actions in the transaction. The concept of mining which is finding the proof of work is a constant and continuous calculation of hash that fits the target. The first node to mine the block is the winner. His corresponding block gets added to the blockchain as a new block. They further explain the protection of personal, digital property and lastly IOT [4].

III. SMART CONTRACT

Smart contracts are the computerized automated protocols that execute the stipulations of a contract. In other terms, these are clauses embedded into property in the form of self - enforcing code. This minimizes the need for trusted intermediaries. Within blockchain context, they are scripts stored on the blockchain. They are assigned unique address. We can trigger the contracts by addressing to them. It then executes independently on every node in the network, according to the data in triggering transaction. Smart contracts allow us to have general purpose computations occur on the blockchain. Where they excel however, is when they are tasked with managing data-driven interactions between entities on the network.[12] The Figure 4.1 illustrates the concept of smart contracts. If we say that blockchain gives us trustworthy storage, then smart contracts provides us distributed trustworthy calculations. [2].

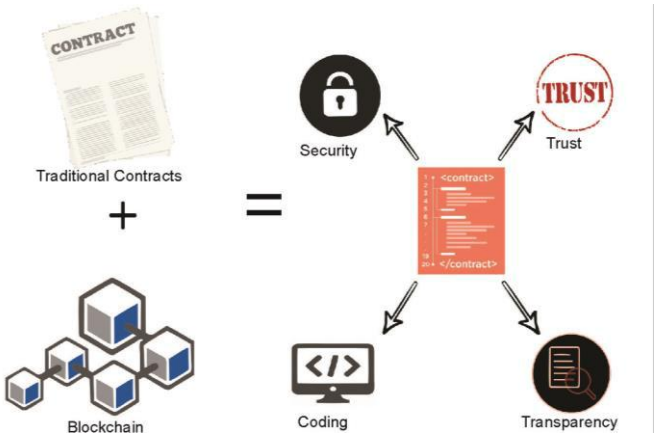

Figure 4.1 Graphical representation of Smart Contract.

Smart contracts are made simpler by the emergence of blockchain as a revolutionary networking paradigm. With the decentralized network, the smart contract has proved to be efficient in overcoming limitations of traditional contracts. A smart contract is a mixture of all the pros and cons of contracts and blockchain. Some of the features of the smart contract have been listed as follows:

A. *Security*: For a transaction to be valid, all transacting participants must agree on its validity.

B. *Code*: The contract is in the form of computer code that can be easily coded by transacting participants. It provides the user the autonomy to code according to their terms.

C. *Transparency*: Participants know where the asset came from and how its ownership has changed over time. A single, shared ledger provides one place to go to

determine the ownership of an asset or the completion of a transaction.

D. *Trust*: No participant can tamper with a transaction after it's been recorded in the ledger. If a transaction is in error, a new transaction must be used to reverse the error, and both transactions are then visible [7].

Smart contract enables contracting on easier terms, cheaper cost and at faster rate. The usage of core underlying technology of blockchain is expanding to support a broad range of collaborative activities amongst businesses, organizations, and individuals. The smart contracts are programs that are fundamentally event driven and data-centric [15]. One of the most prominent frameworks for smart contracts is Ethereum where they are written in Turing-complete language. This system can be majorly used in business negotiations where the statements are event driven assuring the execution without any failure proving reliability while transacting. Here, we preview the blockchain technology to understand the fundamentals of smart contract. Some terms that would facilitate easy understanding of a smart contract code using solidity are also being explained. We analyze the parameters on which smart contracts can be replaceable with the legal system [10]. These characteristics inhibit the usage of smart contracts in various domains where terms and conditions are to be verified before transacting. In the category of finance, contracts verify the ownership of the user and terms of the contract. It records the finance-related transactions over the blockchain network as a contract. For example, Trade affairs. In notary, it is used to store data persistently hence serves as a proof of integrity, for example, Legal contracts. Some games are worth a certain price. The terms and conditions of the game can be agreed and verified by smart contracts. In the domain of Wallet, contracts include hash keys, transactions and money transfer sand watch contracts. Here, it shall be managed by one or many owners. For example, Bitcoin [8] [9].

Smart contracts features	Blockchain	Traditional contracts
Lesser complexity of contract	✓	✓
Third party requirement		✓
Validation	✓	✓
Time effective	✓	
Accuracy	✓	
Service charge		✓
Administration cost		✓
Economic products	✓	
Security	✓	
Automation	✓	
Escrow		✓
Anonymity	✓	
Physical presence		✓
Trust	✓	
Secure backup	✓	
Checkpoints	10	7

Table 4.1 Analysis of Smart contracts on the basis of parameters. [8]

Table 4.1 gives an analysis of the features of smart contracts derived from blockchain and traditional contracts. We can infer from this table that smart contracts can be a revolutionary idea to eradicate the use of conventional methods. We tabulate certain features of smart contracts that have been adapted from the conventional system of contract and the enhanced by the addition of blockchain technology to the system. Writing a contract using the blockchain technology is as simple as writing a contract in the conventional system. The method of validation varies in both the systems. In the conventional system, jury verifies the contracts. On the contrary, a smart contract is validated using consensus algorithm, Proof-Of-Work. So, the fundamental features of a contract remain the same but the way it is executed is different. The blockchain smart contract proves to be an enhanced version the conventional system. Here, no third party involvement is needed. Therefore, except the transacting party, no one is aware of hash key. Though the terms of contracts are open to all, people transacting over it are anonymous. Hence this tamper-proof smart contract provides better security than the documents in the conventional system. Also, a backup is maintained at various nodes thereby ensuring no loss of data, which is not guaranteed in the conventional system. The overall cost of producing a contract is lower. This is because the contracts are executed within minutes so its equivalent to a one-time investment as compared to the conventional contracts which might be dragged for years together and one needs to produce money at every hearing. The significant feature of blockchain smart contracts is its wide application towards economic products. People can even create contracts for minute things to establish easy trust. Whereas considering the time, cost and security factors of conventional contracts it is not possible to create a contract with so much of ease. In spite of all this, a blockchain smart contract lags due to the factor of user-interface. Not every person who wants to contract knows coding. As a result, templates are made available for such users. This technology mainly targets the upcoming generation. In this way, the conventional systems with the add-ons of blockchain technology lead to the advent

of smart contracts. However, research is still in progress to implement these parameters effectively. The smart contract shall replace the way people contract.

IV. SMART CONTRACT ON BLOCKCHAIN

Smart contracts run Ethereum, which in turn runs on each node in the network. In simpler terms, like most virtual machines, it runs for users to easily code their contracts. Several coding languages have been developed for this purpose. One of the most popular ones is solidity.

Solidity is specially designed for Ethereum smart contract. The compiler converts this code into Ethereum Virtual Machine bytecode, which can execute as a transaction over the Ethereum network. Terms to be understood regarding a smart contract:

i. Accounts

Ethereum consists of two types of accounts which share the same address space. The ones controlled by human i.e. by means of private-public key pairs are called as External accounts and the ones that are controlled by the legal codes stored with accounts are called as Contract accounts.

ii. Transactions

Transaction is the exchange of a message among two accounts. The code contained in the target account is executed and input data is in the form is the payload.

iii. Gas

Gas works as a limiting agent for transactions in EVM in terms of the amount of work that needs to be executed. So the transactions are charged with a certain amount of gas, which in turn depletes as the transactions are executed. The value of the gas price is set by the creator of the transaction, who pays from sending account

iv. Storage, Memory and the Stack

The memory area assigned for each account is called as the storage. A contract cannot read or write to any storage of another contract; it is restricted only to its own storage. Memory is the second memory area. For every message call, the contracts are assigned upon with a new instance. Memory gets costlier with its increasing size, as the payment in gas is expected at the time of expansion. EVM is a stack machine, so the stack is the area which holds all the computations. Its maximum size is of 1024 elements and each word consists of 256 bits.

v. Instruction Set

To avoid consensus problems caused due to the incorrect implementations, the instruction set of the EVM is usually kept minimal. Its basic data type is 256-bit words. The instruction set contains basic operations like logical, arithmetic, bit and comparison. Properties of the block such as number and timestamp can be accessed by the contracts.

vi. Message Calls

Message calls are somewhat like transactions which help contract to call other contracts and they consist of a source, target, Ether, gas, data payload (input) and return data. The message calls at the top level of a transaction can further create message calls. Noncontracts receive Ether from contracts by means of message calls.

vii. Delegatecall / Callcode

The variant type of message calls is called as the Delegatecall. Here target address' code is in context of the calling contract. At runtime, the contracts dynamically load the code from various other locations. Calling address only determines the storage, current address and balance whereas only the code is referred from the called address.

viii. Logs

The feature that stores data in a structured index format that helps in mapping is called as logs. Solidity makes use of this feature for implementing events.

ix. Create

A special opcode helps a contract create another contract. Hereby create calls, the data of payload gets executed and the result is stored in form of code. The creator, on the other hand, is updated with the new contracts' address on the stack.

x. Self-destruct

If a contract at a certain address executes self-destruct operation, then the code is removed from the blockchain [3].

Let's take an example:

```
pragma solidity ^0.4.2;
contract Claims {

    mapping(address => mapping(string => string))
private owners;

    function sclaim(string key, string value){
        owners[msg.sender][key] = value;
    }

    function gclaim(address owner, string key)
constant returns (string) {
        return owners[owner][key];
    }

    function sdefaultclaim(string value)
        { sclaim(default, value);
    }

    function gdefaultclaim(address owner) constant
returns (string) {
        return gclaim(owner, default);
    }
}
```

This is a simple example to trade for a claim over private data. This certifies that the owner of certain address is the only one to claim the connection to the other address. For an instance, one can set a claim called "Email", so that anyone who wants to transact with the owner can get the owner's email address. The contract is simple to understand. "Contract" keyword is used to mark the beginning of the contract. Thereafter "Claims" indicates the contract. There are two types of elements: variables and functions inside the contract. Variables can again be classified into two types: constants and writable variables. Writable variables are used

for saving the state in the blockchain. It's these variables that encode the state saved in the blockchain.

Functions are pieces of code to read or modify the state. Read-only functions which are marked as constant do not require gas. On the contrary, the functions that mutate state require operating and encoding the new blocks in the network. The "owners" variable acts as a map (associative array). It matches the key to a value. In our case, the key is an address. Addresses in Ethereum are identifiers for accounts. We are not only mapping from an address to a claim but also the group of key values that constitutes the group of claims. This makes it easier for the owner to reveal multiple details to the transacting user. One might create two claims: one as "Email" and the other under "Name" key. The contracts leave it up to the owner to decide the entries to be created. [5]

V. CONCLUSION

With increasing advancements in the technology of today's online environment, this diverse information once clubbed can provide as a valuable reference for digital identity and credits. In this paper, we analyzed the concept of the blockchain, the domains where smart contracts can be useful and its systematic flow on the blockchain network. We also described the features of Ethereum Virtual Machine used for smart contracts along with an example using solidity for blockchain which is solidity. However, our next step is to focus on the implementation of smart contracts on the Ethereum Virtual Machine. Like any other technology, smart contract too has its own pros and cons. They execute quickly and at a fraction of cost. On the other hand, it's difficult for every person to code by themselves and if any amendments are required, it proves to be costly since it requires re-establishing the block chained ahead of it.

Smart Contracts is not able to uplift under the legal system. It doesn't have the security against the incorrect coding of a user. It has no governing rules as to which users can be assigned to program contracts equivalent to legal support. So, it has a huge scope for improvement in the future. However, this system is still in progress. But it is Smart Contract that will do for business what internet did for communication.

REFERENCES

[1] Jenny Cieplak and Simon Leefatt "SMART CONTRACTS: A SMART WAY TO AUTOMATE PERFORMANCE" 1. GEO. L. TECH.REV. 417(2017)

[2] P. Satyavolu and A Sangamnerkar. "Blockchain's Smart Contracts: Driving the Nect Wave of Innovation Across Manufacturing Value Chains", Connizant, Chennai, Tamil Nadu,2017.

[3] Introduction to Smart Contracts,[Online]. Available: http://solidity.readthedocs.io/en/latest/introduction-to-smart-contracts.html#simple-smart-contract

[4] E. Karafiloski and A Mishey, "Blockchain Solutions for Big Data Challenges", IEEE EUROCON 2017- 17th Int. Conf. On Smart Technologies.

[5] An introdution to Ethereum and Smart Contracts: A programmable Blockchain, March 28, 2017,[Online]. Available: https://auth0.com/blog/an-introduction-to-ethereum-and-smart-contracts-part-2/

[6] F.Zaninotto, The Blockchain Explained to Web Developers, Part 1: The Theory, April 28, 2016,[Online]. Available: https://marmelab.com/blog/2016/04/28/blockchain-for-web-developers-the-theory.html

[7] Manav Gupta, "Grasping Blockchain Fundamentals," in Blockchain for Dummies, IBM Limited Edition, Hoboken, NJ, 2017.

[8] Massimo Bartolleti and Livio Pompianu, "An emirical analysis of smart contracts: platforms applications and design patterns", Universit` degli Studi di Cagliari, Cagliari, Italy, March, 2017.

[9] SWIFT, Distributed Ledgers, Smart Contracts, Business Standards and ISO 20022,2016.

[10] Kieron O'Hara, " Smart Contracts- Dumb Idea, University of Southampton",March 2017.

[11] Hiroki Watanbe, Shigeru Fujimura, Atsushi Nakadaira, Yasuhiko Miyazaki and Akihito Akutsu, "Blockchain Contract: A Complete Consensus using Blockchain", IEEE 4th Global Conference on Consumer Electronics(GCCE), 2015.

[12] Konstantinos Christidis and Michael Devetsikiotis, "Blockchains and Smart Contracts for The Internet of Things", May 2016.

[13] Patrick Dai, Neil Mahi, Jordan Earls and Alex Norta, "Smart Contract Vaue Transfer Protocols on a Distributed Mobile Application Platform", Qtum, Singapore. March 2017.

[14] Hiroki Watanbe, Shigeru Fujimura, Atsushi Nakadaira, Yasuhiko Miyazaki, Akihito Akutsu and Jay Kishigami, "Blockchain Contract: Securing a Blockchain Applied to Smart Contracts."

[15] O. Gadyatskaya and F. Massacci, "Controlling Application Interactions on the Novel Smart Cards with Security-by-Contract", LNCS, vol 7866