# An Overview about Smart Contracts

**[1]Mr. Alen Geo Alex, [2]Ms. Sigma Sathyan**

[1]Student, [2]Assistant Professor
Department of Computer Science,
Santhigiri College of Computer Sciences, Vazhithala Thodupuzha, Kerala

**Abstract: The blockchain technology's smart contracts are a compelling aspect. Traditional contracts can take weeks or even months to begin, and both in the public and private sectors, there have been many instances of breaches and a lack of trust in contracts. In order to negotiate, run, and implement a settlement between two parties, a digital contract executes at the top of the blockchain [1]. Smart contracts can promote transaction credibility between contracting parties without the need for third parties as demonstrated by traditional contracts thanks to the security and decentralized system exhibited by blockchain technology. This paper highlights the uses, advantages, and limitations of smart contracts and demonstrates how they can be deployed in an organization to improve performance using blockchain technology.**

**Index Terms: Blockchain, Bitcoin IoT, Cryptography, solidity, Ethereum**

## I. INTRODUCTION

The rise of cutting-edge technology has increased business competition as each seeks to use them to boost workforce productivity and the overall effectiveness of the company. Technology has consequently evolved into a crucial part of corporate operations and a key driver of innovations and competitiveness for companies. Smart contracts are one of the modern, more dependable, and cost-effective business mechanisms that have replaced old ways of doing business by companies.

Currently Blockchain, the underlying technology for Bitcoin provides various services, including data storage, digital asset transfer, and transaction management, via a decentralized computer architecture. Intelligent contracts, also known as smart contracts, are another innovation in the blockchain space that automate contract clauses using triggers programmed in software. These previously generated and defined triggers could, for instance, be dates when particular terms of a contract will be executed.

## II. WHAT ARE CONTRACTS

### Definition of Traditional Contracts

The phrase "contract" is widely used and has been for a long time. A collection of commitments made by the parties done for a deal are recorded in the contract. The term contract is a common way to communicate with the buyer, seller, and any other parties engaged in a transaction. The term "conventional contracts" refers to written contracts where the terms are specified along with any relevant conditions. They are also known as building lock of a free market economy. These contracts are a crucial component of daily life that benefit us in a variety of ways, particularly when interacting with other people or businesses.

### Problems with Traditional Contracts

Despite the fact that we have been using these contracts for a very long time, there are many issues with them when it comes to safety, security, autonomy, integrity or any other crucial property in a trade. Since the globe is witnessing a digital revolution, everything is changing quickly and the world is heading in that direction. Still with the traditional contracts if a contract breach occurs, the participants must go through a tough procedure. To accomplish their goals, they must interact with law enforcement officials, validate the genuineness and integrity of the contract to the authorities and may even go to court by coercing other parties or participants through the use of force, whether that be financial or otherwise.

The completion of traditional contracts takes time, The amount of time required to organize, design, and create a standard contract can range from one to several days, depending on the quality of the legal services and the preparedness of the contract's parties. A traditional contract is thought to be expensive since there are allegedly "hidden expenses" when taking into account prospective contract complications, such as arbitration, in addition to the third party's need to make a profit. The contractual process demands physical labor, supplies like paper and ink, and other resources, which raises the transaction cost of a contract.

Furthermore, a contract should undoubtedly possess the quality of authenticity. Traditional contracts can indeed be altered or forged if they are not properly safeguarded or examined by a qualified professional. A few minor phrase adjustments can have a significant impact on the final agreement. Traditional contracts also have the drawback of being signed by one party on both ends without the other party's knowledge.

We required a solution that would substitute digital contracts for the conventional paper-based ones. In order to address this issue, Nick Szabo was the first person to use the term "Smart Contracts" in 1996.

## III. BLOCKCHAIN AND BITCOIN

### Blockchain technology

Blockchain is a growing collection of information known as blocks that are connected and secure via cryptography. The P2P protocol used by blockchain can tolerate single points of failure. The consensus process guarantees the consistency and integrity of the blockchain across geographically dispersed nodes and a consistent, clear ordering of transactions and blocks. Blockchain was created with attributes like decentralization, integrity, and auditability. Blockchain has the potential to act as a revolutionary type of software connector that can potentially replace the current centralized shared data storage with a decentralized one.

**Bitcoin and its origin**

In 2008, Satoshi Nakamoto—possibly a pseudonym used by one or more people to remain anonymous—expressed the need for a more secure and reliable payment system that would eliminate the use of third parties, such as banking institutions, in order to operate, allowing value transfers to take place directly between interested parties without the use of a middleman.

The open-source Bitcoin technology was introduced in 2009 and enables value transfers using the bitcoin cryptocurrency, which is created by the system itself. A public blockchain manages and stores transactions with the goal of lowering fees, such as those imposed by banks, and facilitating international negotiations. Bitcoin was the first decentralized cryptocurrency to be created.

## IV. SMART CONTRACTS

Smart contracts were characterized by researchers as automated transaction protocols that carry out contract terms. Other authors define smart contracts as software that uses the dependable computational features of a blockchain network to automatically implement requirements that two parties can agree on when signing a contract in an untrusted environment [2]. In layman's words, smart contracts are a digital version of a traditional contract that contained a series of promises. It is essentially a collection of software programs that operate on a platform and are executed when particular circumstances are satisfied. They can be represented as traditional contract terms that are defined in a computer program and then performed. They are used as real-world traditional contracts for negotiating contract conditions, but they are digitally programmed and saved on a blockchain network, which means they are not centralized but decentralized, making them secure to use digitally.

To improve transactional safety, efficiency, and minimize potential contract beaches, smart contract technology is designed to replace conventional kinds of contracts. The efficiency, dependability, and security that have been noted in decentralized cryptocurrencies like Litecoin, Ethereum, and Bitcoins among others, which the authors point out may be the future of online financial transactions. These factors all led to the emergence of smart contracts systems, which is based on blockchain technology. Smart contracts are also referred to as digital contracts that enable the creation of tamper-proof that are typically self-applicable through automated execution. However, unlike traditional contracts, smart contracts lack the legal status of those contracts and do not make use of artificial intelligence resources. In an idea more closely related to computing, smart contracts are, in essence, small programs that are intended to automate actions based on conditional "if" and "then" instructions, stored and executed without intermediaries in a number of devices connected in a peer-to-peer network, and capable of fulfilling contractual clauses.

**The classic Vending Machine analogy**

Nick Szabo in 1997 described the greatest analogy for a smart contract as a vending machine. Consider your most experience buying a snack from a vending machine. You pushed the button after inserting your card or cash. The vending machine delivered the snack you ordered once you had paid and made your selection [3].

The vending machine is comparable to a smart contract in this way. According to Investopedia, it is a self-executing contract with the terms between buyer and seller directly in the lines of code. In essence, smart contract code states, "Execute output 'y' if input 'x' occurs." As a result, between you (the customer) and the snack merchant, the vending machine serves as a smart contract (the seller). Anyone with coins can trade them for goods or services with the seller. The lockbox and other security measures sufficiently shield the coins and contents from attackers to enable the cost-effective placement of vending machines in a variety of locations. The terms under which the machine releases snacks are specified by the seller. In other words, the buyer chooses and pays, and the vending machine only dispenses goodies if these requirements are completed. When that occurs, the machine immediately carries out its pre-programmed tasks. Below shows a simple and straight forward Solidity code the classic vending machine with the usage of Ethereum [4].

```
**********/
 * Title: A smart contract for Digital Vending Machine
 * Author: Ethereum
 * Date: 2020
 * Code version: 0.8.7
 * Availability: Ethereum.org
**********/ pragma solidity 0.8.7; contract VendingMachine {
 // Declare state variables of the contract  address public owner;
 mapping (address => uint)  public cupcakeBalances;
 // When 'VendingMachine' contract is deployed:
 // 1. set the deploying address as the owner of the contract  // 2. set the deployed smart contract's cupcake balance to 100 constructor() { owner = msg.sender; cupcakeBalances[address(this)] = 100;
 }
 // Allow the owner to increase the smart contract's cupcake balance  function refill(uint amount) public {  require(msg.sender == owner, "Only the owner can refill.");  cupcakeBalances[address(this)] += amount;
 }
 // Allow anyone to purchase cupcakes  function purchase(uint amount) public payable {  require(msg.value >= amount * 1 ether, "You must pay at least 1 ETH per cupcake");  require(cupcakeBalances[address(this)] >= amount, "Not enough cupcakes in stock to complete this purchase");  cupcakeBalances[address(this)] -= amount; cupcakeBalances[msg.sender] += amount;
 }
}
```

**Importance of Blockchain in Smart Contracts**

The computer code for smart contracts operates on the "when X happens, do Y" concept. Therefore, smart contracts are theoretically possible even without networks like Ethereum. One could object that until the invention of the blockchain, smart contracts were not really viable as trustworthy instruments (since it's not legally binding as a traditional contracts).

Why is that so? Let's revisit the example of the vending machine. This analogy was first conceived in the late 1990s by computer scientist Nick Szabo. Before the person or group who goes by the name Satoshi Nakamoto created blockchain, it took more than ten years. Szabo said that smart contracts could only truly function if "breaching the contract was expensive (if desired, sometimes prohibitively so) for the breacher" in the white paper titled "The Idea of Smart Contracts" that he wrote in 1997. Instead of making contract breaching "expensive," as Nick Szabo described, blockchain technology makes breaching smart contracts virtually impossible. Due to the decentralized nature of blockchain networks like Ethereum, a hacker would need to gain access to more than 50% of the computers supporting the network in order to alter a smart contract. In this manner, smart contracts created on open blockchains are unchangeable or immutable.

Additionally, imagine that several parties with known connections formed a smart contract (for example, a record company, a radio station, and a band). In that case, the contract might be written so that any modifications would need the consent of all parties. These agreements are immutable, so there is no chance that one party will try to take advantage of the other by bending the terms of the agreement [5]. This is the reason why we can use smart contracts to create decentralized cryptocurrency exchanges, for example (such as Uniswap). Transactions cannot be carried out by a smart contract until the necessary conditions have been satisfied. Therefore, the contract won't necessarily enforce the rule if someone is attempting to underpay.

In this way, smart contracts are also trustless - a quality of a decentralized blockchain, whereby in using the network there is no need to rely on trust in a third party. In the analogy of the vending machine, "the vending machine has no choice but to deliver the goods upon receiving the money,". As Nick Szabo put it, "smart contract interactions require little to no trust amongst the contracting parties." The machine's technological basis provides assurance that the contract will be carried out as intended.

**V. IMPLEMENTATION OF SMART CONTRACTS**

For instance, an organization that deals with customer credit cards, like banks, would have an aggregate database designated for the task. This database would hold data about purchases made and credit card account balances. A typical table in such a database would have columns for "amount," "owner," and "asset type," among other things. In the database, the table may have entries such as "Sam," "20". A different entry might be "Tim," "0," or something similar ". According to the first record's interpretation, Sam has a credit balance of $20 while Tim has "$0." Sam may start a transaction to transfer the set amount to Tim.

Consider Sam sending Tim $10. Sam's account would be debited $10 and Tim's account would be credited $10. Sam would have $10 after the transaction is completed, and Tim would get $10. Since money is regarded as an asset that can be expressed in digital form, this is an example of a transfer of a digital asset. Though the end users instantly receive the updated account balance, database manipulation is actually taking place, which is logical.

A business organization can use blockchain technology to implement smart contracts and do so in a trustless environment [6]. This would require the implementation of a shared database, in which each row of each table in the database would represent the specifics of a particular entity, which could be a supplier or a customer. However, the attribute would contain the public key of the node or user authorized to change the record rather than the "owner" as in the banking example provided.

A shared database would exist between the company and all other parties in the blockchain network, typically all the customers.

Suppose Customer Y has five units of X [X may represent credit card balance], the respective record in the database would contain Customer Y's Public key in the "owner" column of the respective table, as well as values 5 and X in the quantity and asset type respectively. Customer Y should be able to transfer a certain amount of the digital asset to Company B if they are aware of Company B's public key. In this instance, Customer Y would need to start a signed transaction with her private key that would decrease the specified asset type by the specified units in the amount.

However, the blockchain technology requires the use of transaction blocks, whereby the transactions are linked to one another, as opposed to relational and object-oriented database systems where the manipulation is done directly to the records. As a result, a new record or row is created with the updated information and a timestamp after customer Y purchases a new package and transfers the digital asset [money] to Company B.

```
**********/
*       Title: Simple Amount Transfer Smart Contract
*       Author: OpenAI
*       Date: 2020
*       Code version: 0.6.0 * Availability:
*
**********/ pragma solidity ^0.6.0; contract SimpleContract {  address payable
    public sender;  address payable public receiver;  uint public amount;  bool public transferred;
 constructor(address payable _sender, address payable _receiver, uint _amount) public
{  sender = _sender;  receiver = _receiver;
 amount = _amount;  transferred = false;
 }
 function transfer() public
{   require(!transferred, "Money has already been transferred");  require(sender.balance >= amount, "Insufficient funds");
sender.transfer(amount); receiver.transfer(amount);  transferred = true;
 }
}
```

This is a simple implementation without defining any pre-conditions or event listening to demonstrate both the simplicity and the potential of Smart Contracts.

This contract defines four public variables: sender, receiver, amount, and transferred. The sender and receiver variables are addresses for the two parties involved in the transaction, amount is the amount of money to be transferred, and transferred is a Boolean variable that indicates whether the money has been transferred.

The contract has a constructor function that is called when the contract is deployed. The constructor takes three arguments: the sender address, the receiver address, and the amount to be transferred. These values are stored in the contract's variables, and the transferred variable is initialized to false. The contract also has a transfer function that allows the money to be transferred automatically when certain conditions are met. The function checks that the transferred variable is false, indicating that the money has not yet been transferred, and that the sender has sufficient funds. If these conditions are met, the function transfers the amount of money from the sender to the receiver and sets the transferred variable to true.

A smart contract object is saved to the blockchain as part of a transaction. When a smart contract is deployed to the blockchain, a transaction is created that includes the contract code and any necessary arguments. This transaction is broadcast to the network, where it is validated and added to a block. Once a block containing the transaction is added to the blockchain, the smart contract object is saved to the blockchain and is available for execution by other parties. The smart contract object is stored as part of the state of the blockchain, which is a record of all the current values of the contract's variables.

It's important to note that the smart contract object itself is not stored on the blockchain as a whole. Instead, the blockchain stores the contract's code and the current state of its variables. When the contract is executed, the blockchain executes the code and updates the state of the contract's variables based on the actions taken by the contract. The process of adding a smart contract object to the blockchain is decentralized and secure, as it involves multiple nodes on the network validating and agreeing on the transaction before it is added to the blockchain. This ensures that the smart contract object is saved to the blockchain in a tamperproof and immutable manner.

## VI. BENEFITS OF USING SMART CONTRACTS

### Accuracy

Accuracy is one of the benefits that business organizations would experience from the adoption of smart contracts. All contract information is expressed using if-then statements in a conditional format, as described in the steps for setting up a smart contract. For instance, if customer x orders a certain service package and pays x units of y, the amount will be immediately credited to the recipient and the service package will be opened for customer x. Considering that the majority of contracts involve the exchange of money. The robustness, accuracy, and performance of the entire system would then be further improved by syncing the smart contracts with cryptocurrencies like Ethereum, Lite Coin, or bitcoin, among others. A smart contract must accurately and explicitly state all terms and conditions. In essence, this is a crucial requirement because any omission could result in transaction errors. Therefore, the smart contracts' automation mitigates many of the problems that are present in conventional contracts.

### Clear Communication and Transparency

The various network participants of the specific blockchain virtually have explicit access to the contract terms and conditions. Therefore, changes cannot be easily implemented once the contract is established. Other network nodes in the blockchain keep track of and regulate each transaction made by either party to the contract. As a result, fraud problems are resolved and transparency is promoted. Numerous instances of organizations being accused of defrauding their customers and failing to provide them with value for their money have been reported in the modern era.

The use of smart contracts makes every aspect of the contract transparent. In the virtual world, other nodes in the network are all that is required to ensure that every transaction related to the contract is accurate and valid, as opposed to the traditional contract where the organization would have to use the legal framework as the intermediary.

### Speed and Efficiency

Fundamentally, smart contracts don't require human involvement; instead, other nodes in the blockchain network direct and monitor their execution. Consequently, the scripted contract self-executes after being triggered. This is frequently accomplished by using trigger events when scripting the contact. An example of a trigger event is the transfer of specific cryptocurrency units from a customer's wallet to the business's wallet. A trigger event could also be a specific date, time, or even an activity that is started by one of the parties to the contract. The contract now begins executing itself when a trigger event occurs. For online subscriptionbased businesses, the customer's subscription is automatically renewed after a specific amount of cryptocurrency is received.

The blockchain network's nodes here verify whether the correct amount has been paid, as well as whether the right subsection, service, and related aspects, have been given to the number, as opposed to less efficient traditional contracts that call for some sort of human verification. As a result, the organization no longer relies on its developed system to decide on contracts with customers. Additionally, the organization lacks sovereign authority over both the contractual agreement with the partners and the transactions. Each contract is targeted as a distinct entity, and every transaction is first validated regardless of where it came from. Overall, this produces a quick, durable, and reliable method of contract execution.

### Security

The smart contracts are found to have one of the highest security measures in a study by Marino and Juels. Utilizing a decentralized network of parties is necessary for the implementation of smart contracts using block chin technology. Due to their lack of trust for one another, the network's parties constantly monitor one another to ensure that each transaction is completed successfully and that everyone has a shared understanding of how each transaction is progressing.

Once more, cryptography is used to implement blockchain technology. High data encryption is required by this technology, and both private and public keys must be used to read each blockchain's transactions as well as to carry out any transaction[7]. The

security of the smart technology is increased by the requirement that all nodes within the blockchain network validate a transaction prior to any node committing it.

**Cost reduction**

Creating plans and methods to cut costs within an organization is essentially the responsibility of top business managers. The primary goal of establishing a business enterprise is to generate profits, so all organizational activities must be designed to maximize shareholder wealth while also promoting the achievement of corporate goals.

But it's important to remember that even though smart contracts have advantages in terms of efficiency, cost savings, and security, they are not magical and may have flaws. For instance, the input, which is essentially the coded version of the contract, has a significant impact on the quality and execution of the contract. As a result, if the smart contracts are set up incorrectly, this could have negative effects and result in output that is of low quality.

## VII. LIMITATION OF SMART CONTRACTS

**Immutability**

In principle, once created, smart contracts are difficult to change because they are written as a piece of code. Traditional contracts frequently use amendments to terms and conditions, particularly long-term contracts whose execution is dependent on actual-world dynamics and the conditions are ever-changing. After being created, smart contracts' rigidity causes a variety of practical issues, especially when it comes to how simple it is to change the terms of the contract to suit different circumstances. B.

**Contractual Secrecy**

Since all transactions are recorded on a general ledger using encoded permissions in each node, blockchain technology typically involves sharing smart contracts across all of the nodes in the blockchain network. In simple terms, blockchain technology involves the use of anonymity, whereby each member of a blockchain network is protected and anonymous. However, there is no assurance that the contract will be carried out. This is due to the public ledger being maintained, which makes the transactions visible and unsecure even though the nodes operate in an anonymous manner. Buterin explains that this is an area that needs to be focused because despite the nodes being anonymous, the maintenance of a public ledger in the distributed environment results in a privacy lapse.

## VIII. APPLICATIONS OF SMART CONTRACTS

Smart contracts have the potential to revolutionize the way we conduct business and exchange value, as they can be used to automate many processes and reduce the need for intermediaries. Here are a few examples of real-life applications of smart contracts:

**Decentralized finance platforms**

Without the need for a middleman, decentralized finance platforms can now offer financial services. This is made possible by cryptocurrencies and smart contracts. By the end of 2021, DeFi had a total locked value of $94 billion. More than just peer-to-peer transactions are now a part of DeFi. On DeFi platforms, smart contracts have made it possible to conduct complex transactions like lending, borrowing, and derivatives.

**Non-Fungible Tokens**

NFT trading reached $17 billion in 2021, making it one of the most significant use cases for smart contracts. Even though the market has slowed down in the second quarter of 2022, NFTs have practical applications that could result in their long-term use.

By distributing ownership and controlling the transferability of non-fungible tokens (NFTs), smart contracts have made it possible to create NFTs. These agreements can also be changed to include extra provisions like royalties and software or platform access rights.

**Gaming**

The global gaming market is a $100 billion ecosystem that is expanding quickly, but the way value is created and distributed within the market can sometimes be unfair. The blockchain networks that support NFTs enable player ownership, provable scarcity, interoperability, and immutability, while these tokens are one-of-a-kind, uncommon, and indivisible. These features of blockchain in gaming could promote widespread adoption and a more equitable value model when taken as a whole. You can save in-game purchases, sell them to other players, or transfer them to other supported games thanks to the adoption of blockchain technology in the gaming sector.

**Real Estate**

Particularly, there have been a number of successful attempts to tokenize real estate assets, including through real estate-focused blockchain platforms like RealT and SolidBlock. By incorporating blockchain into real estate transactions, smart contract technology can also rework the documentation and transaction processes. For instance, since 2016, the Republic of Georgia (in the Caucasus region) has been developing a blockchain-based land title registry. Similar initiatives are also ongoing in other countries, like the United Arab Emirates (UAE). Anyone who has bought a house or another piece of property is probably aware of the potential for hidden costs related to closing costs, title transfers, and broker fees.

**Insurance**

With a compound annual growth rate of 85%, the global market for blockchain in insurance is projected to reach $1.39 billion in 2023. The management of claims and data collection can be automated by smart contracts to enhance the insurance process. For instance: Blockchain technology has been tested by sizable insurance firms. AXA Insurance introduced the insurance product Fizzy in 2017, which uses smart contract technology to handle claims for flight delay insurance. When a delay of more than two hours occurs, the smart contract is connected to global air traffic databases, triggering payment automatically.

**Supply Chain Management**

Supply chain management, which involves actively streamlining a company's supply-side operations, is the management of the flow of goods. Once an item travels to its destination in a supply chain network, its ownership status is altered [8]. With the aid of IoT sensors and smart contracts, everyone in the supply chain can track the location of the item. Smart contracts have the ability to

locate lost items if they are found later in the process. To eliminate the need for document-based communication between organizations, smart contracts can also automate payments and routine tasks.

For instance: By utilizing blockchain technology and smart contracts, which shorten the time it takes to settle disputes with vendors, Home Depot has enhanced its vendor management process. The ability for vendors and retailers to view the same data concurrently cuts down on the time needed for disputes and enhances invoice management.

## IX. CONCLUSION

Smart contracts are not a new concept but along with the use of blockchain network it got popularity. It took a lot of time for everyone to recognize the potential of smart contracts which has still not been recognized completely. It is obvious that every technology comes with some pros and cons, hence smart contracts are no exception but the problem can be resolved and the smart contracts can be used to its full potential with various different technologies. Smart contracts play a great role to digitalize the world as it is included in many aspects of our daily life. In other words, smart contracts require initial fine-tuning but then operate independently once set up. If you can accomplish this fine tuning, the advantages are comparable to those of conventional contracts. When we arrive in India, we anticipate it will take some time for the locals to comprehend and accept this idea. We are still tinkering with the traditional contracts and trying to make them better. It will take some time for smart contracts to adapt to the Indian environment, but once they do, they will be around for a very long time.

## REFERENCES

1. A Literature Review about Smart Contracts Technology
2. IRJET- Smart Contracts using Blockchain
3. An Overview of Smart Contract and Uses
4. Etherum Docs on smart contracts [5] Smart Contracts by Sajid Baloch
5. The Budding World of Smart Contracts
6. Real World Examples of Smart Contracts and dApps
7. Smart Contract and its Uses