NAME: MESHACK NYAGWENCHA BOSIRE

NEPTUN CODE: IVF370

BLOCKCHAIN TECHNOLOGY PROJECT

# SMART CONTRACTS

Guide

What are smart contracts?

History of smart contracts

What led to the introduction of smart contracts in the world of blockchain technology

How do smart contracts work?

Types of smart contracts

How are the smart contracts created?

Composition of a smart contract

What it takes to be able to write a smart contract

Sample smart contracts

Current applications of smart contacts

Benefits and challenges of smart contracts

Suggested improvements and potential transformation of smart contracts.

## What is a smart contract?

A smart contract is defined as a digital agreement that is signed and stored on a blockchain network, which executes automatically when the contract’s terms and conditions are met.

Code that **facilitates**, **verifies** or **enforces** the negotiation or execution of a digital contract a. Trusted entity must run this code - (lecture definition)

Citation, definition from https://www.spiceworks.com/tech/innovation/articles/what-are-smart-contracts/#:~:text=A%20smart%20contract%20is%20defined,programming%20languages%20such%20as%20Solidity.

5

## History of Smart Contracts

The history of smart contracts dates back to 1997 when Nick Szabo published a paper titled ‘The Idea of Smart Contracts’. Nick states in his third paragraph that ‘Smart contracts go beyond a vending machine in proposing to embed contacts in all sorts of property that is valuable and controlled by digital means. Smart contracts reference that property in a dynamic, often proactively enforced form, and provide much better observation and verification where proactive measures must fall short’

“As another example, consider a hypothetical digital security for automobiles. The smart contract design strategy suggests that we successively refine security protocols to more fully embed in a property the contractual terms which deal with it. These protocols would give control of the cryptographic keys for operating the property to the person who rightfully owns that property, based on the terms of the contract. In the most straightforward implementation, the car can be rendered inoperable unless the proper challenge-response protocol is completed with its rightful owner, preventing theft.“

**CT:** From “<https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html>" by Nick Szabo.

Many predictions made by Szabo in his paper are now part of our daily lives in ways that precede blockchain technology.

The invention of Ethereum blockchain platform by programmer **Vitalik Buterin** in 2013 made practical use of smart contracts achievable. Ethereum is still one of the most prevalent platforms enabling smart contract implementation.

## HOW DO SMART CONTRACTS WORK?

A smart contract is a binding agreement between two parties. It used code to leverage the blockchain technology, thereby unlocking greater efficacy, openness, and confidentiality.

The execution of smart contracts is controlled by relatively easy “if/when…then…” statements written in code on the blockchain.

### STEPS NEEDED FOR THE FUNCTIONING OF SMART CONTRACTS

**1, AGREEMENT:**

The parties wanting to conduct business or exchange products or services must concur on the arrangement’s terms and conditions. Furthermore, they must determine how a smart contract will operate, including the criteria that must be fulfilled for the agreement to be fulfilled.

**2. CONTRACT CREATION**

Participants in a transaction may create a smart contract in many ways, including building it themselves or collaborating with a smart contract provider. The provisions of the contract are coded in a programming language. During this stage, verifying the contract’s security thoroughly is critical.

**3. DEPLOYMENT**

When the contract has been finalized, it must be published on the blockchain. The smart contract is uploaded to the blockchain in the same way as regular crypto transactions, with the code inserted into the data field of the exchange. Once the transaction has been verified, it’s deemed active on the blockchain and cannot be reversed or amended.

**4. MONITORING CONDITIONS**

A smart contract runs by tracking the blockchain or a different reliable source for predetermined conditions or prompts. These triggers can be just about anything that can be digitally verified, like a date attained, a payment made, etc.

**5. EXECUTION**

When the trigger parameters are met, the smart contract is activated as per the “if/when…then…” statement. This may implement only one or multiple actions, like passing funds to a vendor or registering the buyer’s possession of an asset.

**6 RECORDING**

Contract execution results are promptly published on the blockchain. The blockchain system verifies the actions taken, logs their completion as an exchange, and stores the concluded agreement on the blockchain. This document is available at all times.

## TYPES OF SMART CONTRACTS.

Smart contracts are classified into three categories namely:

* Legal contracts
* Decentralized autonomous organizations (DAOS)
* Logic contracts

### **SMART LEGAL CONTRACT**

Smart contracts are guaranteed by law. They adhere to the structure of legal contracts: “If this happens, and then this will happen.” As smart contracts reside on blockchain and are unchangeable, judicial or legal smart contracts offer greater transparency than traditional documents among contracting entities.

The parties involved execute contracts with digital signatures. Smart legal contracts may be executed autonomously if certain prerequisites are fulfilled, for example, making a payment when a specific deadline is reached. In the event of failure to comply, stakeholders could face severe legal repercussions.

### 2. **DECENTRALIZED AUTONOMOUS ORGANIZATIONS**

DAOs are democratic groups governed by a smart contract that confers them with voting rights. A DAO serves as a blockchain governed organization with a shared objective that is collectively controlled. No executive or president exists. Instead, blockchain-based tenets embedded within the contract’s code regulate how the organization functions and funds are allocated. Examples of DAO include Decentraland, an online virtual world, where administrative and governance decisions are taken by stakeholders in a democratic fashion. Anyone who owns the platform’s token, known as a MANA, can take part in the process and the platform is quickly becoming popular with global brands including Morgan Stanley reaching out to digital-native audiences.

More examples of DAOs can be found on this website: <https://www.forbes.com/sites/bernardmarr/2022/05/25/the-best-examples-of-daos-everyone-should-know-about/>

### 3. **APPLICATION LOGIC CONTRACTS.**

AICs, as the are mostly called, consist of application-based code that typically remains synced with various other blockchain contracts. It enables interactions between various devices, like the Internet of Things(IoT) or blockchain integration. Unlike the other types of smart contracts, there are not signed between humans or organizations but between machines and other contracts.

## HOW ARE SMART CONTRACTS CREATED?

To be able to create a smart contract, think of a smart contract as a computer program that runs on a blockchain. These programs have a set of transaction terms built into the code of the “contract”. They perform all sorts of functions by establishing terms, managing the parties involved, and then carrying out transactions. Smart contracts can be found on networks like Ethereum, Hedera, Fantom, Avalanche, and Harmony. They rely on coding languages like Solidity.

Let’s use a typical idea to understand the ease the understanding of the smart contract. When buying a home, both parties involved will sign a contract. Yet, the contract only lays out the terms, it cannot enforce them. Instead, a third party, like the court, must do the enforcement.

However, a smart contract can enforce itself, using its code, which eliminates the need for a middleman.

Here is a detailed dive into how to get a smart contract written. This takes several stages, from creating your own smart contract idea to writing, compiling, and deploying the code using your own smart contract language. Finally, you can execute the contract and verify its effectiveness.

**The concept**

Let’s you’re looking to raise funds for a business venture, You can use a smart contract to automatically run your fundraising operations. Your smart contract can recognize when you reach your goal and can stop the fundraise return unneeded donation and do other tasks.

Whatever need you’re looking to address, knowing your desired outcome will help you begin the coding process with clarity

**The coding**

To begin you’ll need an environment(Integrated Development Environment) in which to write the code.

A great example of a good IDE is Remix. It works on the desktop and as a web application. Remix specializes in Ethereum development, so you can use Ethereum’s Solidity programming language to write the smart contract code. A benefit of Remix is that it offers a quick development cycle with a host of useful plugins.

**The testing**

Rigorous testing is absolutely vital. Smart contracts are immutable once placed on the mainnet, which means you can’t edit any errors found after deployment.

To test smart contracts, people use testnets( or test chains) or local blockchain networks. Testing doesn’t require real cryptocurrency. Instead, you can use dummy currency for testing purposes to immediately get feedback on how well your contract works. Smart contracts are immutable once placed on the mainnet, which means you can’t edit any errors found after deployment.

**Compiling**

Before deploying smart contracts, they must be compiled. This refers to converting your contract’s code into a JSON file so it can be read by a normal web app. For example, an Ethereum smart contract, after being written in solidity, the contracts are compiled to the byte code of the EVM, or Ethereum Virtual Machine, making it compatible with all EVM networks.

**Deploying**

Deploying is placing the contract into your chosen network. When you deploy smart contracts, you will execute them and make a transaction using real crypto. To do so, you’ll specify details like the appropriate wallets and rewards. After this stage, your deployed contract will be launched and all your coded functions will kick into gear after established conditions are met.

**Follow through**

After your contract is executed, deployed and live on the main net, you’ll want to do your due diligence to see that it’s running smoothly. This includes checking your wallet to ensure the right balance appear when they should. This also is the time to address any storage issues and maintenance tasks.

## How much does a smart contract cost?

Smart contracts are priced by the amount of “gas” it costs for nodes on the network to execute the contract multiplied by the **gas price.**

Gas is a measurement unit for the fees associated with each smart contract transaction. Gas is the price of doing business in a smart contract.

Nodes process smart contracts, and in exchange for their work, they receive a fixed payment of gas. On the Hedera network, for example, a node processes a contract, and then gets paid in HBAR.

In designing your smart contract, you’ll specify the amount of gas it will require. There are a few considerations to keep in mind in estimating and setting this gas limit. The amount of gas earned by processing a smart contract depends on how complicated it is. You will want to specify the right amount needed. I you over-specify the gas needed, the extra will be returned to you. But if you under price, your contract execution will fail and generate an “out of gas“ error.

**The development cost.** Development shops can charge anywhere from $7000 to $45,000 to create and deploy a smart contract. For a contract to be used by a large organization, the price might hit around $100,000. Of course, people with the necessary coding skills could create a DIY smart contract for a fraction of those costs.

**The audit cost.**  Because of the immutable nature of a smart contract, most companies pay for an audit of a smart contract before it goes on the main net. A third party checks all the coding and the performance of the smart contract. This costs a few thousands dollars to $15000. The price can be much higher for a large, complex contract.

Source: https://hedera.com/learning/smart-contracts/how-to-create-a-smart-contract

**Uses of smart contracts**

Uses of smart contracts are wide and varied, spread across industries:

1. Royalty payment in media and entertainment

New artists rely on revenue from streaming services. Smart contract apps can facilitate easier royalty payments. There contracts can outline, for instance, the share of royalties payable to the record company and the artist. Instantaneous handling of these payments is an enormous advantage for everyone involved.

Smart contracts could also potentially solve the challenge of royalty distribution in an over-the-top(OTT) content world where traditional network agreements do not apply. This technology allows emerging artists and lesser known actors to get small but regular payments.

2. Decentralized finance(DeFi) applications

Using cryptocurrencies and smart contracts, DeFi apps can offer financial services without an intermediary. DeFi is no longer limited to peer-to-peer transactions. On DeFi platforms, smart contracts facilitate complex processes like borrowing, lending or derivative transactions.

3. Conversion of assets into non-fungible tokens(NFTs)

By assigning ownership and administering the movable nature of digital assets, smart contracts have made it possible to create NFTs. Contracts like this can also be altered to include added stipulations, like royalties, along with access rights to platforms or software. Essentially, smart contracts make it possible to treat digital assets just like physical ones, with real tangible value.

4. B2B data marketplaces

A data marketplace is a portal where users can buy and sell diverse datasets or data streams from a wide range of sources. Intelligent contracts facilitate the creation of dynamic and fast-evolving markets that support automated and secure transactions without the hassle of human intervention. Datapace is a good example of this particular smart contract use case.

5. Supply Chain Management

Smart contracts may work autonomously without mediators or third parties because they are self-executing. An organization can create smart contracts for an entire supply chain. This would not require regular management or auditing. Any shipments received beyond the schedule might trigger stipulated escalation measures to guarantee seamless execution.

6. Digital identity cards

Users can store reputational data and digital assets on smart contracts to generate a digital identification card. When smart contracts are linked to multiple online services, other external stakeholders can learn about individuals without divulging their true identities.

For instance, these contracts may include credit scores lenders can use to verify loan applicants without the risk of demographic profiling or discrimination. Similarly, candidates can share resumes without the risk of gender bias in hiring.

7. Electoral polls

Voting could occur within a secure environment created by smart contracts, minimizing the likelihood of voter manipulation. Due to the encryption, every vote is ledger-protected and extremely difficult to decode. Additionally, smart contracts might boost voter turnout. With an online voting system driven by smart contracts, one can avoid making trips to a polling location.

8. Real estate

Smart contracts can accelerate the handover of property ownership. Contracts can be autonomously created and executed. After the buyer’s payment to the vendor, for instance, the smart contract may immediately assign control over the asset dependent on the blockchain’s payment record.

9. Healthcare data management

Smart contracts can revolutionize healthcare by making data recording more open and efficient. For instance, they might encourage clinical trials by guaranteeing data integrity. Hospitals can maintain accurate patient data records and effectively manage appointments.

10. Civil law

Smart contracts can also flourish in the legal industry. It can be used to create legally binding business and social contracts. In certain regions of North America, governments have authorized smart contracts for digitized agreements. For example, California can issue marital and birth certificates as smart contracts.

## Benefits and challenges of smart contracts

The key reasons to use smart contracts include:

1. **Single source of truth**

Individuals have the same data at all times, which reduces the likelihood of contract clause exploitation. This enhances trust and safety because contract-related information is accessible throughout the duration of the contract. As the transactions are replicated, all involved parties have a copy.

2. **Trust and Transparency.**

Since there’s no third party involvement and because transaction records are shared across the blockchain network, there’s no need to question whether the information has been altered for personal gain.

3. **Reduction in human effort**

Smart contracts don’t need third-party verification or human oversight. This provides participants autonomy and independence, particularly in the case of Decentralized Autonomous Organizations (DAO).

This intrinsic characteristic of smart contracts offers additional benefits, including cost savings and faster processes.

4. P**revention of errors**

A fundamental prerequisite for any contract is that every term and condition is recorded in explicit detail. Automated smart contracts avoid form-filling errors. This is one of its greatest advantages.

5. **Built-n backup**

These contracts capture essential transactional details. Therefore, whenever your data is used in a contract, it is stored indefinitely for future reference. In an instance of data loss, it is simple to retrieve these properties.

### Challenges of smart contracts

Here are the potential downsides of smart contracts and the challenges to be aware of:

1. **Rigidity and inconsistent support**

Modifying smart contract protocols is nearly impossible, and fixing code errors can be costly and time-consuming. Even if smart contracts conform to the laws of different countries, it might be tough to guarantee that they are adhered to globally.

2 **Difficulty in capturing unquantifiable data**

It is difficult to represent works from industries such as art where creative work is the core of such industries and the works do not have quantifiable metrics.

3. **Conflict with GDPR**

The General Data Protection Regulation (GDPR) guarantees ***the right to be forgotten by its citizens****.*They can request that digital data about them be deleted. Nevertheless, if a digital legal contract binds an individual, it cannot be erased or redacted.

4. **Skills Shortage**

The creation of smart contracts demands expertise in software engineering. Smart contract development is distinct from traditional software development in that it requires coders with organizational expertise and comprehension of non-traditional programming languages such as Solidity. These skills are hard to come by.

## Discussion.

Smart contracts have a very big potential of transforming the world into a more functional place to live. From our daily lives to the highest stages of power, agreements are continuously made and deals struck.

But in all agreements done, the risk of compromising the deal is inevitable. Having smart contracts which execute contracts as along a certain condition is met levels the ground sealing all loopholes and possibilities of disregarding the agreement terms.

I see smart contracts as a solution to some of the biggest problems in my country like corruption, disrespect of court orders, debt unworthiness, election rigging and even overstaying in power of greedy leaders.[](https://www.youtube.com/watch?v=pA6CGuXEKtQ)

How Smart Contracts Will Change the World | Olga Mack | TEDxSanFrancisco

Smart contracts can be written that follows the constitution of a country that dictates when the term of a president ends and this date can not be extended.

Transparency can be achieved in the electoral process and this will reduce the destruction of property and loss of life after every election year when citizens across the political divide disagree on the election results and start rioting.

## Conclusion.

I fully advocate for implementation of smart contracts in governments worldwide and daily live transactions as they can make the world a more functional and just place to inhabit.