Smart Contracts: Transactions Processing

Table of Content

Introduction

Smart contracts are one of the most fundamental building blocks of blockchain technology or distributed ledger technology. The term 'smart contract' was first introduced by computer scientist and cryptographer Nick Szabo in the 1990s. In his publication, Szabo defined smart contract as "a set of promises, specified in digital form, including protocols within which the parties perform on these promises."

As blockchains gained in popularity, smart contracts began to flourish — especially with the emergence of Ethereum, one of the most popular blockchains that supports smart contracts. These self-executing contracts automatically enforce the rules and agreements encoded within them, making them ideal for a wide range of use cases, from supply chain management to financial instruments.

Smart contracts can enhance blockchain security in several ways. First, they can eliminate the need for third-party intermediaries, such as banks, lawyers, or escrow agents, who may be vulnerable to fraud, corruption, or human error. Second, they can ensure that the transactions and interactions on the blockchain are immutable, verifiable, and traceable, as they are recorded and validated by the network. Third, they can enable more advanced and customized security features, such as multi-signature verification, time locks, oracles, or encryption.

What Are Smart Contracts?

A smart contract is an agreement between two people or entities in the form of computer code programmed to execute automatically. A simple example could be in the case of life insurance. The policy terms would be encoded into the smart contract. In the event of a passing, the notarized death certificate would be provided as the input trigger for the smart contract to release the payment to the named beneficiaries.

Smart Contracts simply defined: Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met. Smart contracts are executed on blockchain, which means that the terms are stored in a distributed database and cannot be changed.

History of Smart Contracts

Smart contracts were first proposed in 1994 by Nick Szabo, an American computer scientist who invented a virtual currency called "Bit Gold" in 1998, 10 years before Bitcoin was introduced. In fact, Szabo is often rumoured to be the real Satoshi Nakamoto, the anonymous Bitcoin inventor, which he has denied. Szabo defined smart contracts as computerized transaction protocols that execute the terms of a contract. He wanted to extend the functionality of electronic transaction methods, such as POS (point of sale), to the digital realm.

Szabo famously compared a smart contract to a vending machine. Imagine a machine that sells cans of soda for a quarter. If you put a dollar into the machine and select a soda, the machine is hardwired to either produce your drink and 75 cents in change, or (if your choice is sold out) to prompt you to make another selection or get your dollar back. This is an example of a simple smart contract. Just like a soda machine can automate a sale without a human intermediary, smart contracts can automate virtually any kind of exchange.

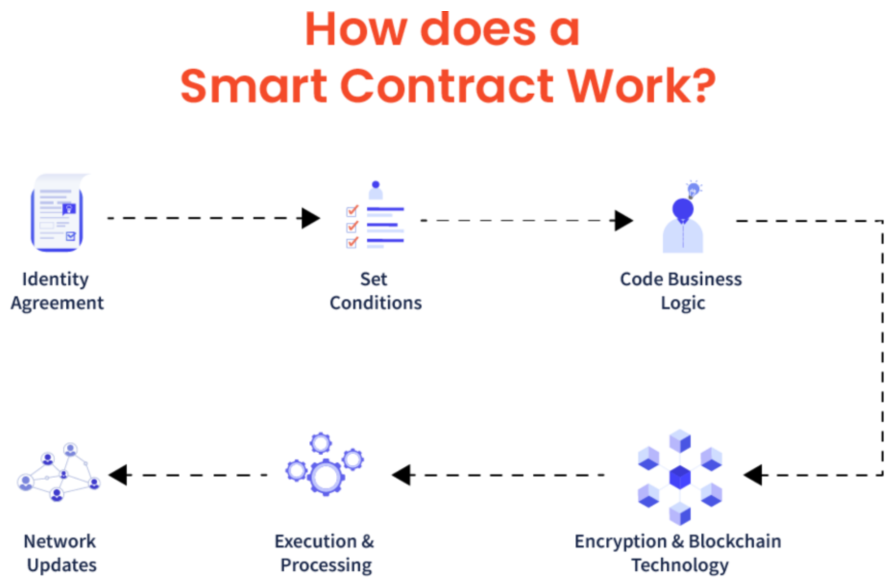
In his paper, Szabo also proposed the execution of a contract for synthetic assets, such as derivatives and bonds. Szabo wrote, "These new securities are formed by combining securities (such as bonds) and derivatives (options and futures) in a wide variety of ways. Very complex term structures for payments...can now be built into standardized contracts and traded with low transaction costs, due to computerized analysis of these complex term structures."

Many of Szabo's predictions in the paper came true in ways preceding blockchain technology. For example, derivatives trading is now mostly conducted through computer networks using complex term structures.

Smart contracts today also find their origin in Ricardian Contracts, a concept published in 1996 by Ian Grigg and Gary Howland as part of their work on the Ricardo payment system to transfer assets. Grigg saw Ricardian Contracts as a bridge between text contracts and code that had the following parameters: a single document that "is

i. a contract offered by an issuer to holders,
ii. for a valuable right held by holders, and managed by the issuer,
iii. easily readable by people (like a contract on paper),
iv. readable by programs (parsable like a database),
v. digitally signed,
vi. carries the keys and server information, and
vii. allied with a unique and secure identifier."

How does Smart Contracts work?



Identify Agreement: Multiple parties identify the cooperative opportunity and desired outcomes and agreements could include business processes, asset swaps, etc.
Set Conditions: Smart contracts could be initiated by parties themselves or when certain conditions are met like financial market indices, events like GPS locations, etc.
Code Business Logic: A computer program is written that will be executed automatically when the conditional parameters are met.
Encryption and Blockchain Technology: Encryption provides secure authentication and transfer of messages between parties relating to smart contracts.
Execution and Processing: In blockchain iteration, whenever consensus is reached between the parties regarding authentication and verification then the code is executed and the outcomes are memorialized for compliance and verification.
Network Updates: After smart contracts are executed, all the nodes on the network update their ledger to reflect the new state. Once the record is posted and verified on the blockchain network, it cannot be modified, it is in append mode only.

Types of Smart Contracts

When it comes to the types of smart contracts, they are classified into three categories — legal contracts, decentralized autonomous organizations or DAOs, and logic contracts.

1. 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. VitaDAO is an example of this type of smart contract, where the technology powers a community for scientific research.

3. Application Logic Contracts

ALCs, or application logic contracts, 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, these are not signed between humans or organizations but between machines and other contracts.

How Smart Contracts created on Ethereum?

Ethereum provides a decentralized virtual computer - the Ethereum Virtual Machine (EVM) - on which developers can build applications consisting of multiple smart contracts. Think of the EVM as a distributed global computer where all smart contracts are executed. (Ethereum is sometimes referred to as a "world computer.") Ethereum lets developers program their own smart contracts to define EVM instructions. The EVM executes a contract according to the rules the developer programmed. Smart contracts live in the form of bytecode within the decentralized database. This is the root of the innovation and disruptive potential of Ethereum. Contracts are written in Solidity, a Javascript-like language developed specifically for writing smart contracts.

The Ethereum white paper explains it:

The intent of Ethereum is to create an alternative protocol for building decentralized applications, providing a different set of tradeoffs that we believe will be very useful for a large class of decentralized applications […] Ethereum does this by building what is essentially the ultimate abstract foundational layer: a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions."

Paying with Gas

Whenever smart contract transactions occur within the Ethereum blockchain, it takes computational power to validate them across the network. To compensate for the time and energy it takes to perform those computations, transactions require a fee in the form of "gas" which is paid with ETH. The amount of gas depends on the amount of computation required to complete the transaction. This gives developers an incentive to create simple and efficient smart contracts. Moreover, the gas limit — the maximum amount of gas allotted to a particular transaction — means buggy code won't run indefinitely, costing an infinite amount of gas. (Unspent gas is automatically refunded, but if the transaction runs out of gas, it's aborted and no gas is refunded.)

The Technology behind Smart Contracts

The Ethereum blockchain powers the technology underneath. This decentralised and distributed ledger securely records transactions and data. Smart contracts are self-executing computer programs that run on the Ethereum blockchain and enforce the terms of an agreement automatically. Developers write these contracts in a high-level programming language and compile them into low-level bytecode, which the Ethereum blockchain stores. The Ethereum Virtual Machine, a computer network that runs the Ethereum blockchain, executes the bytecode. When someone makes a transaction on the Ethereum blockchain, it triggers the smart contract to run and enforce the agreement's terms. The decentralised and distributed nature of the ledger ensures the security and transparency of the agreement's terms, as multiple computers store the transaction details, and anyone can audit them. By using smart contracts, individuals can automate various agreements and transactions, reducing the risk of fraud and the need for intermediaries.

Companies Using Ethereum Smart Contracts

Many companies have adopted Ethereum smart contracts to provide secure and efficient solutions for their business processes. Some of the companies using Ethereum smart contracts include:

- Microsoft: Microsoft has adopted Ethereum smart contracts to provide a secure and transparent platform for managing the supply chain of its products.
- JPMorgan Chase: JPMorgan Chase is using them to increase the efficiency and security of its cross-border payments.
- Accenture: Accenture uses them to provide secure and transparent solutions for its clients' supply chains.

Features of Smart Contracts

The following are some essential characteristics of a smart contract:

Distributed: Everyone on the network is guaranteed to have a copy of all the conditions of the smart contract and they cannot be changed by one of the parties. A smart contract is replicated and distributed by all the nodes connected to the network.

Deterministic: Smart contracts can only perform functions for which they are designed only when the required conditions are met. The final outcome will not vary, no matter who executes the smart contract.

Immutable: Once deployed smart contract cannot be changed, it can only be removed as long as the functionality is implemented previously.

Autonomy: There is no third party involved. The contract is made by you and shared between the parties. No intermediaries are involved which minimizes bullying and grants full authority to the dealing parties. Also, the smart contract is maintained and executed by all the nodes on the network, thus removing all the controlling power from any one party's hand.

Customizable: Smart contracts have the ability for modification or we can say customization before being launched to do what the user wants it to do.

Transparent: Smart contracts are always stored on a public distributed ledger called blockchain due to which the code is visible to everyone, whether or not they are participants in the smart contract.

Trustless: These are not required by third parties to verify the integrity of the process or to check whether the required conditions are met.

Self-verifying: These are self-verifying due to automated possibilities.

Self-enforcing: These are self-enforcing when the conditions and rules are met at all stages.

Capabilities of Smart Contracts

- **Accuracy:** Smart contracts are accurate to the limit a programmer has accurately coded them for execution.
- **Automation:** Smart contracts can automate the tasks/ processes that are done manually.
- **Speed:** Smart contracts use software code to automate tasks, thereby reducing the time it takes to manoeuvre through all the human interaction-related processes. Because everything is coded, the time taken to do all the work is the time taken for the code in the smart contract to execute.
- **Backup:** Every node in the blockchain maintains the shared ledger, providing probably the best backup facility.
- **Security:** Cryptography can make sure that the assets are safe and sound. Even if someone breaks the encryption, the hacker will have to modify all the blocks that come after the block which has been modified. Please note that this is a highly difficult and computation-intensive task and is practically impossible for a small or medium-sized organization to do.
- **Savings:** Smart contracts save money as they eliminate the presence of intermediaries in the process. Also, the money spent on the paperwork is minimal to zero.
- **Manages information:** Smart contract manages users' agreement, and stores information about an application like domain registration, membership records, etc.
- **Multi-signature accounts:** Smart contracts support multi-signature accounts to distribute funds as soon as all the parties involved confirm the agreement.

Advantages of Smart Contracts

Recordkeeping: All contract transactions are stored in chronological order in the blockchain and can be accessed along with the complete audit trail. However, the parties involved can be secured cryptographically for full privacy.

Autonomy: There are direct dealings between parties. Smart contracts remove the need for intermediaries and allow for transparent, direct relationships with customers.

Reduce fraud: Fraudulent activity detection and reduction. Smart contracts are stored in the blockchain. Forcefully modifying the blockchain is very difficult as it's computation-intensive. Also, a violation of the smart contract can be detected by the nodes in the network and such a violation attempt is marked invalid and not stored in the blockchain.

Fault-tolerance: Since no single person or entity is in control of the digital assets, one-party domination and situation of one part backing out do not happen as the platform is decentralized and so even if one node detaches itself from the network, the contract remains intact.

Enhanced trust: Business agreements are automatically executed and enforced. Plus, these agreements are immutable and therefore unbreakable and undeniable.

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

Cost-efficiency: The application of smart contracts eliminates the need for intermediaries (brokers, lawyers, notaries, witnesses, etc.) leading to reduced costs. Also eliminates paperwork leading to paper saving and money-saving.

Prevention of errors: A fundamental prerequisite for any contract is that every term and condition is recorded in explicit detail. An omission may result in serious issues in the future, including disproportionate penalties and legal complexities. Automated smart contracts avoid form-filling errors. This is one of its greatest advantages.

Built-in 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.

Use cases of Smart Contracts

Businesses want to simplify their processes and speed up existing workflows. They can leverage digital contracts to do this. Here are fields/sectors that can benefit from smart contracts.

Real Estate: Reduce money paid to the middleman and distribute between the parties actually involved. For example, a smart contract to transfer ownership of an apartment once a certain number of resources have been transferred to the seller's account (or wallet).

Vehicle ownership: A smart contract can be deployed in a blockchain that keeps track of vehicle maintenance and ownership. The smart contract can, for example, enforce vehicle maintenance service every six months; failure of which will lead to suspension of driving license.

Music Industry: The music industry could record the ownership of music in a blockchain. A smart contract can be embedded in the blockchain and royalties can be credited to the owner's account when the song is used for commercial purposes. It can also work in resolving ownership disputes.

Government Elections: Once the votes are logged in the blockchain, it would be very hard to decrypt the voter address and modify the vote leading to more confidence against the ill practices.

Management: The blockchain application in management can streamline and automate many decisions that are taken late or deferred. Every decision is transparent and available to any party who has the authority (an application on the private blockchain). For example, a smart contract can be deployed to trigger the supply of raw materials when 10 tonnes of plastic bags are produced.

Healthcare: Automating healthcare payment processes using smart contracts can prevent fraud. Every treatment is registered on the ledger and in the end, the smart contract can calculate the sum of all the transactions. The patient cannot be discharged from the hospital until the bill has been paid and can be coded in the smart contract.

Supply Chain Management: With smart contracts, every transaction and movement of goods can be recorded and tracked on a shared ledger that is visible and verifiable by all participants, which can increase trust, accountability, and traceability in the supply chain. Additionally, smart contracts can execute predefined actions based on predefined conditions, such as releasing payments, transferring ownership, or triggering alerts, which can reduce manual errors, delays, and fraud, as well as save time and resources.

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.

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.

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. 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.

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 non-fungible tokens (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.

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.

Faster & Safer International Trade: Top blockchain development companies assist organizations in streamlining international trade with smart contract blockchain platforms. These platforms expedite transactions, provide transaction history, and monitor the entire trading process from start to finish. By using standardized rules, blockchain-based smart contract platforms manage trade options, reduce risk and friction, simplify the trading process, and create more trade opportunities.

Challenges of Smart Contracts

No Regulations: A lack of international regulations focusing on blockchain technology (and related technology like smart contracts, mining, and use cases like cryptocurrency) makes these technologies difficult to oversee.

Difficult to Implement: Smart contracts are also complicated to implement because it's still a relatively new concept and research is still going on to understand the smart contract and its implications fully.

Immutable: They are practically immutable. Whenever there is a change that has to be incorporated into the contract, a new contract has to be made and implemented in the blockchain.

Alignment: Smart contracts can speed the execution of the process that span multiple parties irrespective of the fact whether the smart contracts are in alignment with all the parties' intention and understanding.

Scalability Issues: There is the question of magnitude and scale. Visa can currently process approximately 24,000 transactions per second. According to Worldcoin's 2023 update, Ethereum, the world’s biggest blockchain for smart contracts, can only manage 30 transactions per second.

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.

Dependency on External Data: Smart contracts typically rely on external data sources, known as oracles, to retrieve information from the outside world. While smart contracts themselves are tamper-proof, these oracles can introduce potential vulnerabilities or inaccuracies because they are susceptible to manipulation and tampering.

Difficulty in Capturing Unquantifiable Data: For businesses with quantifiable data, such as finance and agriculture, it is relatively simple to put together smart contracts. However, not all industries use quantifiable metrics, like scenarios where creative work has to be evaluated.

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.

Code Vulnerabilities: Smart contract code, like any software, may contain vulnerabilities or bugs that can be exploited by malicious actors. Errors in code implementation or design can lead to security vulnerabilities that could result in financial loss or other negative consequences. Smart contracts need to undergo rigorous testing before being deployed to avoid exposing users to these dangers.

Third Party Involvement: Although smart contracts seek to eliminate third-party involvement, it is not possible to eliminate them. Third parties assume different roles from the ones they take in traditional contracts. For example, lawyers will not be needed to prepare individual contracts; however, they will be needed by developers to understand the terms to create codes for smart contracts.

Vague Terms: Since contracts include terms that are not always understood, smart contracts are not always able to handle terms and conditions that are vague.

However, the blockchain community actively addresses these challenges through bug bounty programs, smart contract audits, and collaborative development efforts. Security experts participate in bug bounty programs, audit firms conduct thorough security assessments, and developers work on creating tools and standards. Standardization efforts aim to improve interoperability and compatibility between blockchain platforms, collectively contributing to the improvement of smart contract technology.

Future of Smart Contracts

Today, smart contracts are a prototypical example of “Amara’s Law,” the concept articulated by Stanford University computer scientist Roy Amara that we tend to overestimate new technology in the short run and underestimate it in the long run. Although smart contracts will need to evolve before they are widely adopted for production use in complex commercial relationships, they have the impact to revolutionize the reward and incentive structure that shapes how parties’ contract in the future. To that end, and when thinking about smart contracts, it is important not to simply think how existing concepts and structures can be ported over to this new technology. Rather, the true revolution of smart contracts will come from entirely new paradigms that we have not yet envisioned.

Best Smart Contract Development Platforms

The foremost smart contract platform in our list of top smart contract platforms in widespread use is Ethereum, renowned for its exceptional advantages for users. Nonetheless, multiple other smart contract platforms have surfaced, offering versatility for commercial application development. Presently, the demand for smart contracts is surging in tandem with the increased adoption of Web3 technologies. The current best smart contract platforms - Ethereum, Hyperledger Fabric, Nem, Waves and Neo, are discussed in this section.

Ethereum

Ethereum stands as the pioneering force in smart contract technology, heralding the era of decentralized applications (dApps) and revolutionizing the blockchain landscape. As the first smart contract platform, Ethereum remains unrivalled, commanding the largest market share in the realm of smart contracts by a substantial margin. This dominance is attributed to Ethereum's robust infrastructure, fostering an expansive ecosystem of dApps, smart contract coins, and decentralized finance (DeFi) protocols.

The platform first went live in July 2015. Since then, it has grown by leaps and bounds and now facilitates smart contracts for everything from ICOs to facilitating smart contract use with almost any kind of decentralized application. In fact, the vast majority of ICOs now use the ERC-20 token standard to facilitate their offering. The beauty of Ethereum‘s smart contract platform is the degree of standardization and support it offers. Ethereum is completely dedicated to improving the way that smart contracts are created and operated.

Ethereum has published a set of clearly defined rules for developers to follow, making smart contract development easier and less risky. To this end, they have even developed their own smart contract programming language, Solidity, which not only helps with standardization but also makes setting up contracts much easier too.

An area of concern is Ethereum ‘s success itself. Its network has been recorded as frequently running at 100% capacity, something that may worry application developers who need the guarantee that their contracts will always be processed quickly. Also, developers have found a number of security issues with the Ethereum code. Added to this, poor-quality smart contract code has left many contracts exposed to hackers.

Hyperledger Fabric

The second in the list of smart contract platforms and top of the list of Ethereum's competitors is Hyperledger Fabric. The Hyperledger project began in December 2015 and was set up by the Linux Foundation. It is an open-source project that has the stated goal of supporting the development of blockchain-based distributed ledgers. There are now several different frameworks being developed under the Hyperledger banner. These include Hyperledger Burrow, Hyperledger Fabric, Hyperledger Sawtooth, and Hyperledger Indy.

Unlike Ethereum, which uses virtual machines (VMs) to run smart contracts (i.e., EVM), Hyperledger Fabric smart contract uses a Docker container to run the code. Unlike VMs, containers can allow smart contract applications at a reduced cost without compromising isolation (i.e., applications in one container are running on top of one operating system).

Like Ethereum, Hyperledger developers have created a set of extremely helpful tools that include Hyperledger Composer, a JavaScript-based set of tools that allow developers to create smart contracts more easily and efficiently. Developers can create smart contracts in JavaScript with Hyperledger Composer, in Go, and with a range of other common programming languages simply by installing the relevant modules. This feature makes Hyperledger far more flexible as developers do not need to rely on a single language such as Solidity to code their smart contracts in.

Hyperledger Fabric is a permissioned blockchain infrastructure that facilitates the execution of smart contracts or “chain codes”. Since it is a permissioned network meaning that all the participants in the network have known identities. This makes it the go-to choice for companies who wish to create smart contracts but need to comply with data protection laws that require them to be known. Hyperledger Fabric was designed with trust, secrecy, and security at the forefront of its mind. Users of the platform can construct secret channels for certain network members, allowing only selected participants to see transaction data. Furthermore, by including an extra hardware-based security model for identity management, the platform provides the greatest degree of business cybersecurity.

Hyperledge Fabric technology is new and with complex architecture and the lack of proper documentation further makes it more complex to understand. Besides it suffers from lack of proven use cases, limited understanding of technology and its potential, limited talent and skill-sets shortage, etc.

Steller

Stellar was founded back in 2014 by Jed McCaleb, one of the Ripple Founders, making it one of the oldest smart contract platforms. It is maintained by the Stellar Development Foundation and has been repeatedly heralded as one of the most exciting blockchain startups out there.

The facilitation of international payments is one of the primary areas in which organizations seek to employ Stellar. Late in 2017, it teamed with IBM and KlickEx to create a new low-cost method for transacting international business in the South Pacific area. When it comes to the top smart contract platforms, Stellar is arguably simpler and easier to use than Ethereum but perhaps not as straightforward as Nem. However, it is really designed to facilitate simple smart contracts such as ICOs. This is because Stellar is primarily a money exchange and so operates along those lines.

Stellar smart contracts (SSC) are much different from Ethereum smart contracts. They are not Turing complete and are implemented as an agreement between multiple parties and enforced by transactions. An SSC is expressed as compositions of transactions that are connected and executed using various constraints. Stellar smart contracts are linked and performed utilizing a variety of restrictions including multi-signature, batching, time limits, and more. In a word, Stellar is an excellent alternative for organizations seeking a simple yet powerful smart contract solution.

However, the platform's application range is limited. For developing more complicated smart contracts this does not rank well in the smart contract platforms list. This best smart contract platform enables money exchange and functions as such. More advanced smart contracts, such as those used in Dapps, are not appropriate for this platform.

Nem

NEM was released on March 31, 2015, and currently, it is backed by a large community of developers as it is written in Java programming language. It is the reason why this platform is super accessible as it is not restricted to platform-specific programming languages like Solidity or anything. NEM has customized data management apps called smart assets through which one can create tokens, data records, voting systems and other coins with just a few clicks.

According to multiple security experts, Nem's new Catapult or Mijin v.2 versions has made it the most secure smart contract platform available. It is an industry milestone that enables new blockchain database functionality possibilities. Nem's greatest selling feature is that it is very scalable. Additionally, although Ethereum can perform around 15 transactions per second, Nem can manage hundreds. This is the reason why more and more developers are abandoning platforms such as Ethereum.

NEM uses code off the blockchain which makes it less decentralized, the method does have its benefits, such as better security, easier updates, faster execution times and lighter code.

Nem has a smaller development community than other blockchain platforms due to its less decentralized nature. A small community can create problems for new developers to get help from the Nem blockchain experts regarding any difficulties in development and programming. Also, It comes with fewer development tools which overall limits the development potential of the Nem blockchain.

Waves

Another one of the best alternatives in the list of smart contract platforms, Waves was launched in June 2016. It is an open-source platform that aimed to address many of the existing barriers that stand in the way of more mainstream blockchain implementation, namely speed and scalability. Mainly, it permits users to program and deal with the terms of a contract without involving a middleman. Waves smart contract platform is deemed to be user-friendly and facilitates developers with the required tools and APIs to craft and launch their apps.

Wave has positioned itself as a platform to facilitate token operations. As such it is another excellent platform for ICOs. It takes just a few minutes to create your own tokens on the platform, something which is so easy to do as it requires next to no technical knowledge.

It has a minimal user base, and it lacks a development community, which can create problems for developers who want help from experienced developers during the development of crypto contracts.

Neo

Neo is an open-source blockchain platform created in China. Neo enables smart contacts to automate digital asset transactions, similar to the Ethereum (ETH) Network.

Neo supports many programming languages, making it easier for novice and experienced developers to develop smart contracts on its platform. Neo has also designed most of its features to provide more benefits for its users, such as NeoVM, NeoQS, and NeoFS.

- NeoVM: NeoVM is a virtual machine that can execute NEO smart contracts. Using its transpiler, NeoCompiler, it can compile code in various programming languages into its own bytecode. This type of feature does not exist in Ethereum, where developers often have to learn a new programming language to create smart contracts. Further, NeoVM has extensive functionalities to interact with other blockchains and services.
- NeoQS: NeoQS is a system that enhances Neo's security against advanced quantum computing attacks that threaten the privacy of the blockchain transaction records.
- NeoFS: NeoFS is a decentralized storage system for users to store Internet of Things (IoT) data in unstructured formats. Additionally, users can earn GAS for lending out storage space.

Neo offers digital identity solutions, making it suitable for the development of applications that require identity verification and compliance. In addition, Neo has created a dBFT mechanism that is low-cost with fast execution speed. However, the Neo ecosystem is not decentralized enough to truly ensure the safety of the protocol, which may deter users from utilizing the platform.

Comparison of Smart Contract Platforms

Platforms	Execution Environment	Programming Languages	Turing Completeness	Permission Type	TPS	Consensus Mechanism	Application
Ethereum	EVM	Solidity	Yes	Public	100,000	PoS	Decentralised Exchanges, Gambling
Hyperledger	DockerGo	JavaScript	Yes	Private	2,000	BFT	Smart Energy Management, Supply Chain

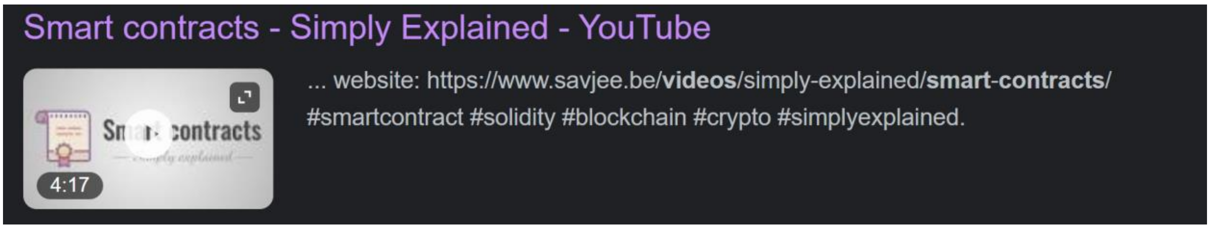
Stellar	DockerScala	Net, Go, C++	No	Consortium	4,000	FBA (5CP)	Universal Payment Solutions, Oil Trade
NEM	XEM	Solidity, Java	No	Private	4,000	Proof of Importance	Smart Asset Management, Cryptocurrency
WAVES	Decentralized	Ride	No	Public	100	PoS	ICOs and crowd-sales
Neo	NeoVM (Neo Virtual Machine)	Javascript, Java, Go, Python	Yes	Private	10,000	dBFT	Smart Ecobomy

Summing Up

Smart contracts self-execute agreements based on predefined protocols. Their capacity to perform this function in an automated and tamper-proof environment makes them one of the most innovative and promising technologies in development. By replacing trust in centralized institutions (such as banks, attorneys, and financial advisors) with trust in blockchain-compatible computer code, smart contracts could usher in an era of automated "trustless systems."

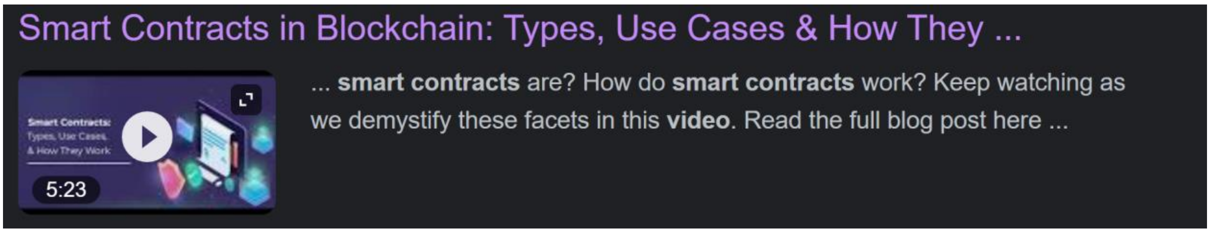
However, relying on computer code alone for important tasks can be efficient, but it is also risky. Most of us are not ready to sell a home or other large asset via an irreversible electronic smart contract. Plus, smart contracts are evolving, with basic legal and regulatory frameworks still taking shape. But as smart contracts use cases inch toward mainstream adoption, we can see hints of their potentially significant, if not disruptive, role in the future of our economy and society. It is expected that 2024 brings in an exciting chapter in the evolution of smart contract platforms.

References:
https://www.britannica.com/money/how-smart-contracts-work
https://www.spiceworks.com/tech/innovation/articles/what-are-smart-contracts/
https://www.geeksforgeeks.org/smart-contracts-in-blockchain/
https://emeritus.org/blog/technology-what-are-smart-contracts/


Smart contracts - Simply Explained - YouTube

... website: https://www.savjee.be/**videos**/simply-explained/**smart-contracts**/ #smartcontract #solidity #blockchain #crypto #simplyexplained.

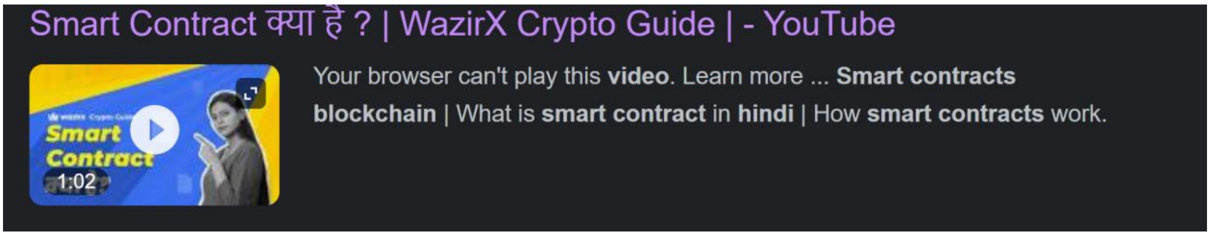
https://www.youtube.com/watch?v=ZE2HxTmxfrI



Smart Contracts in Blockchain: Types, Use Cases & How They ...

... **smart contracts** are? How do **smart contracts** work? Keep watching as we demystify these facets in this **video**. Read the full blog post here ...

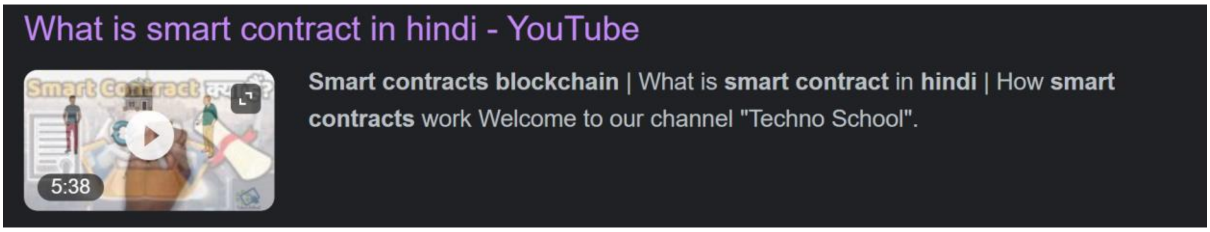
https://www.youtube.com/watch?v=wT6Qa_kxM6Y



Smart Contract क्या है ? | WazirX Crypto Guide | - YouTube

Your browser can't play this **video**. Learn more ... **Smart contracts blockchain** | What is **smart contract** in **hindi** | How **smart contracts** work.

https://www.youtube.com/watch?v=FYHvyRgYLhs



What is smart contract in hindi - YouTube

Smart contracts blockchain | What is **smart contract** in **hindi** | How **smart contracts** work Welcome to our channel "Techno School".

https://www.youtube.com/watch?v=KHRxt-XicN8