Introduction to Ethereum and Smart Contract

Table of Content

S.No	Topic
1	Introduction
2	Key Characteristics of Ethereum
3	What are Blocks?
4	Understanding Ethereum and Smart Contracts
5	Introduction to Ethereum and Smart Contracts
6	Ethereum: A Decentralized Platforms
7	Smart Contracts: Concept and Functionality
8	Solidity: Ethereum's Smart Contract Language
9	Creating and Deploying Smart Contracts
10	Ethereum Gas and Transactions
11	Ethereum Improvement Proposals
12	Use Cases and Applications of Smart Contracts

Although the Bitcoin network provided a base for a medium of exchange, a 19-year-old programmer saw it as a method capable of challenging centralized entities across the economy. Further, unable to convince the Bitcoin community of the need for a scripting language, Vitalik and a like-minded group of people created Ethereum. The main motivation behind Ethereum was to support building decentralized applications on the powerful medium of blockchain

Vitalik Buterin published the Ethereum whitepaper in 2013, in which he introduced a novel, general-purpose blockchain network with a built-in Turing-complete language, which is a programming language that can be used to embed logic and complete more advanced transactions than simple payments. The introduction of this language has allowed developers to create and integrate applications into Ethereum, serving as the base layer of an open ecosystem capable of hosting smart contracts and decentralized applications (DApps). In reality, Buterin created a system of programmable money that revolutionized how people think about, create, and deploy blockchain technology.

Ethereum was initially released in 2015. Within two years of its release, it was ranked the second best blockchain network, after Bitcoin. The Ethereum network attained more global interest when China stated that it is the best blockchain network ever created.

Ethereum is a decentralized blockchain platform that allows users to create and execute smart contract and decentralized applications (dApps). It helps ensure transparency and security through cryptographic techniques. Thus, Ethereum is a one-of-its-kind programmable blockchain rather than a cryptocurrency focused on payments. The platform utilizes a cryptocurrency called Ether (ETH) as its native currency. So, like Bitcoin, the Ethereum architecture also has two aspects:

- The cryptocurrency
- The Ethereum blockchain.

The official website's definition of Ethereum is as follows: "Ethereum is a global, open-source platform for decentralized applications. On Ethereum, one can write code that controls digital value, runs exactly as programmed, and is accessible anywhere in the world".

One significant contribution of Ethereum is that it liberated the blockchain technology from the financial limits of Bitcoin and expanded its scope. It showed the world how other industries could benefit from its application.

Ethereum's vision is to build a 'new internet.' One that would be decentralized as it was always meant to be. An internet in which,

- Peer-to-peer networks would replace the client-server model.
- Any data would be owned only by its creator.
- There would be no monopoly of data.
- Applications won't steal data in the name of 'tailor-made' services.

Key Characteristics of Ethereum

Ethereum offers several key characteristics that set it apart from other blockchain platforms:

- **Turing-Completeness**
  Ethereum's programming language, Solidity, allows developers to create complex and versatile smart contracts. This feature gives Ethereum the ability to handle a wide range of computational tasks, making it highly flexible.

- **Decentralization**
  Decentralization lies at the heart of Ethereum. It operates on a global network of computers called nodes, which collectively maintain the blockchain and validate transactions. This decentralized architecture ensures the security and immutability of the platform.

- **Interoperability**
  The Ethereum blockchain's design allows it to interact with other blockchains and platforms through various protocols and standards. This interoperability facilitates cross-chain operations and the integration of its assets into other blockchain ecosystems. Thus, Ethereum provides a standardized framework for DApps to interact with each other. This interoperability fosters collaboration and enhances the overall functionality of the ecosystem.
  In blockchain technology, a block is a record of new transactions that have been added to the blockchain. Each block contains a unique code called a "hash" that allows it to be distinguished from every other block, as well as a "hash" of the previous block in the chain, linking the two. This creates a chain of blocks, or a "blockchain," that cannot be altered or tampered with because any change to a block would also change its hash and would therefore no longer match the hash of the previous block. This is what makes blockchain technology secure and tamper-proof.

What is a Block in Blockchain?

In the blockchain, a block is a core concept that can be considered a page in a ledger. Blocks contain transactions and critical data, such as previous hash, that ensure immutability and security in the blockchain network. Each Block stores a previous hash sequentially, so it is almost infeasible to reverse and tamper data. The Block will be created when a miner finds a specific value(nonce) by calculating.

What is Block in Ethereum?

In Ethereum, a block is a collection of transactions and other data that are added to the Ethereum blockchain. Each block contains a unique code called a "hash" that allows it to be distinguished from every other block, as well as a "hash" of the previous block in the chain, linking the two.

- In addition to transactions, blocks in Ethereum also contain other types of data such as smart contract code and the results of that code being executed. Each block also includes a timestamp and information about the miner who mined the block.
- The blocks in Ethereum blockchain are added through a consensus mechanism called Proof of Stake, which is different from Bitcoin's Proof of Work mechanism.
- Ethereum blocks are mined at a fixed rate of around 15 seconds, which makes the Ethereum blockchain faster than Bitcoin's, which has a block time of 10 minutes.
- An Ethereum block is a collection of transactions that are processed and verified by the network's nodes. Each block contains a block header and body.
- The block also contains the transactions themselves, which are grouped into a single Merkle tree. This allows for efficient verification of transactions without having to include the entire block data in the header.

Understanding Ethereum and Smart Contracts

1. Introduction

Blockchain technology revolutionized digital transactions by introducing decentralized, immutable ledgers. Ethereum, launched in 2015 by Vitalik Buterin, expanded this concept by offering a decentralized platform beyond simple transactions. It introduced the concept of smart contracts, enabling developers to build decentralized applications (dApps) that execute code automatically without intermediaries.

2. Ethereum: A Decentralized Platform

Ethereum Overview: Ethereum is a decentralized platform that uses blockchain technology to enable developers to create dApps and smart contracts. The Ethereum Virtual Machine (EVM) processes smart contracts, executing code across the network. Ether (ETH), Ethereum's native cryptocurrency, powers transactions and incentivizes miners to secure the network through proof-of-work (PoW) consensus (soon to transition to proof-of-stake, Ethereum 2.0).

Key Features of Ethereum:

- **Decentralization:** Eliminates the need for centralized authorities in transactions and applications.
- **Smart Contracts:** Automate agreements without intermediaries, enhancing efficiency and reducing costs.
- **Ether (ETH):** Acts as a medium of exchange within the Ethereum ecosystem.

3. Smart Contracts: Concept and Functionality

Smart Contract Fundamentals: Smart contracts are self-executing contracts with predefined conditions. They facilitate, verify, or enforce the negotiation or performance of a contract without intermediaries. These contracts are immutable, transparent, and execute automatically once conditions are met. They have the potential to revolutionize various industries by automating complex processes and reducing reliance on intermediaries.

Advantages of Smart Contracts:

- **Automation:** Execute actions automatically when predefined conditions are met.
- **Transparency:** Immutable on the blockchain, providing transparency and traceability.
- **Security:** Immutable nature reduces fraud and ensures data integrity.

4. Solidity: Ethereum's Smart Contract Language

Introduction to Solidity: Solidity is Ethereum's primary programming language for writing smart contracts. It's statically typed and supports inheritance, libraries, and complex user-defined types. Developers use Solidity to define the logic of smart contracts, specifying conditions, functions, and interactions within the Ethereum ecosystem.

Solidity's Key Components:

- **Syntax:** Similar to JavaScript and C++, making it accessible to developers.
- **Data Types:** Supports various data types, including integers, strings, arrays, and structs.
- **Functions and Modifiers:** Defines contract functions and applies modifiers to restrict access.

5. Creating and Deploying Smart Contracts

Developing Smart Contracts: Developers create smart contracts using Solidity or other Ethereum-compatible languages. Tools like Remix, Truffle, and Hardhat aid in development, testing, and deployment. Smart contracts define the rules and logic governing transactions and interactions within decentralized applications.

Steps for Creating Smart Contracts:

a) **Design Phase:** Define the contract's purpose, structure, and functionalities.

b) **Coding:** Write Solidity code implementing the contract's logic and functions.

c) **Testing:** Use test networks or frameworks to test contract functionality and identify bugs.

d) **Deployment:** Deploy the smart contract to the Ethereum network using addresses generated for interactions.

6. Ethereum Gas and Transactions

Understanding Gas: Gas is the unit used to measure the computational effort required to execute operations or contracts on Ethereum. Transactions and smart contract executions consume gas, and users pay gas fees to miners for processing. Gas fees fluctuate based on network demand and the complexity of operations.

Gas Optimization: Developers optimize smart contracts to reduce gas consumption, enhancing efficiency and cost-effectiveness. Techniques include using data types efficiently, minimizing loops, and avoiding unnecessary computations to reduce gas fees.

7. Ethereum Improvement Proposals (EIPs)

Purpose of EIPs: Ethereum Improvement Proposals (EIPs) are design documents proposing changes, improvements, or new features to the Ethereum ecosystem. They facilitate community discussions and help in the evolution and development of the network.

Notable EIPs:

- **ERC-20:** Standard for fungible tokens, enabling the creation of various tokens on Ethereum.
- **EIP-1559:** Introduces changes to Ethereum's fee market, aiming for better fee estimation and management.

8. Use Cases and Applications of Smart Contracts

Real-world Applications: Smart contracts find diverse applications across multiple industries:

- **Finance:** Automated lending, decentralized exchanges (DEXs), and stablecoins.
- **Supply Chain:** Traceability, authenticity verification, and logistics optimization.
- **Healthcare:** Patient data management, insurance claims processing, and drug traceability.

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.