Blockchain Fundamentals And Smart Contracts

Given the enormous interest in Web3 technologies, I wanted to take a deep dive into understanding the applications of blockchain, how they are built, and what system design principles underpin them. Therefore, for this blog post, I decided to focus on two topics: **Blockchain** and **Smart Contracts**. This will be fun. So, let us get started.

Terminologies

Before we understand the fundamentals of Blockchain and Smart Contracts, let us understand some of the common terminologies used. Understandably, some terms such as Merkle Trees, Proof-Of-Stake, and Proof-Of-Work systems will require separate blog posts. I plan to cover them in the future to understand them myself.

- Cryptographic Hash: A cryptographic hash function (CHF) is a mathematical algorithm that maps data of arbitrary size (often called the "message") to a bit array of a fixed size (the "hash value," "hash," or "message digest"). It is a one-way function, that is, a function for which it is practically infeasible to invert or reverse the computation. Ideally, the only way to find a message that produces a given hash is to attempt a brute-force search of possible inputs to see if they produce a match. We often use the SHA-256 hash function to generate private and public keys for SSH. It is a patented cryptographic hash function that outputs a value that is 256 bits long.
- Merkle Tree: A hash tree or Merkle tree is a tree in which every "leaf" (node) is labeled with the cryptographic hash of a data block, and every node that is not a leaf (called a branch, inner node, or inode) is labeled with the cryptographic hash of the labels of its child nodes. A hash tree allows efficient and secure verification of the contents of a large data structure. A hash tree is a hash list and a hash chain generalization.

- Peer To Peer Network: A peer-to-peer network is a distributed system architecture that partitions work between different nodes (workstations) within the network. The nodes represent servers or workstations, and the edges represent the communication channels.
- Distributed Ledger: A distributed ledger is a decentralized database wherein the data is digitally replicated amongst geographically distributed nodes. The nodes participate in ensuring the validity of the data using a consensus algorithm. The data is usually replicated, synchronized, and shared using computational algorithms.
- Proof-Of-Work System: Proof of work (PoW) is a form of cryptographic proof in which one party (the prover) proves to others (the verifiers) that a certain amount of a specific computational effort has been expended. Verifiers can subsequently confirm this expenditure with minimal effort on their part. A simple way to understand a proof-of-work system would be as follows:
  a) A prover is asked to calculate the sum of all natural numbers from 1 to 1 trillion without the knowledge of a formula for it (i.e., the sum is **[n(n+1)]/2**, where n is the number of natural numbers)
  b) A verifier can easily verify the output using the formula.
  The effort required on the end of the prover and the verifier is asymmetric and still verifiable.

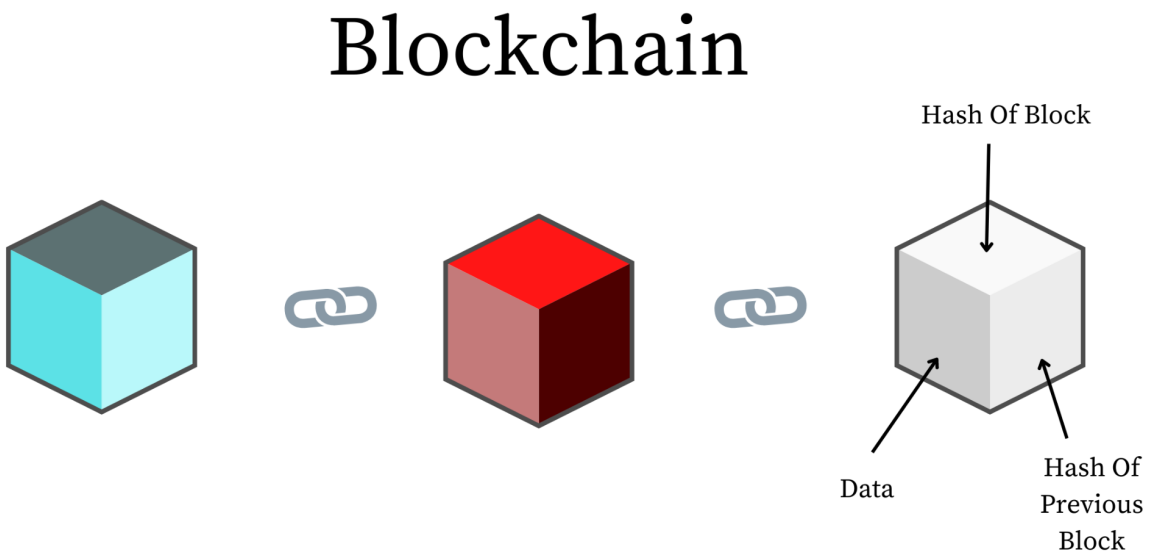
- Block Time: The block time is the time it takes for a network to create an extra block in the blockchain. The process requires the network to ensure that the data in the new block is verifiable and consistent with the constraints of the network (e.g., the digital signature on the new block should not be malformed). For the Ethereum blockchain, the block time is between 14 and 15 seconds, while the average block time for bitcoin is 10 minutes.

What is a blockchain?

As the name indicates, a blockchain is a blockchain containing information. It is a growing list of records, called blocks, linked together by cryptography. Each block contains a

cryptographic hash of the previous block, a timestamp, and transaction data (generally represented by a Merkle Tree). The timestamp proves that the transaction data existed when the block was published to get into its hash. As blocks each contain information about the block previous to it, they form a chain, with each additional block reinforcing the ones before it. Peer-to-peer networks typically manage blockchains for use as a publicly distributed ledger.

To understand how distributed systems work, I would encourage you all to go through the series of essays I have written in the past on distributed systems, starting with the fundamentals of distributed systems.



History of blockchain

In his 1982 dissertation "Computer Systems Established, Maintained, and Trusted by Mutually Suspicious Groups, Cryptographer David Chaum first proposed a blockchain-like protocol. [1] However, it was not popularized until Satoshi Nakamoto wrote a white paper on the design and implementation of blockchain technology — Bitcoin. I would highly

encourage anyone interested in taking a deep dive into blockchain technology to read the white paper. It will take a few attempts but it is certainly worth the read.

Structure of a blockchain

A blockchain can be visualized as a collection of distributed blocks running on a peer-to-peer network. The ledger is decentralized, distributed, and public and consists of blocks that are connected through a series of hashes (or signatures) which are calculated through a cryptographic function. Logically, a blockchain can be seen as consisting of several layers:

- infrastructure (hardware)
- networking (node discovery, information propagation, and verification)
- consensus (proof of work, proof of stake)
- data (blocks, transactions)
- application (smart contracts/decentralized applications, if applicable)

For the scope of this newsletter series, I will be primarily focused on discussing the application layer and what elements are necessary for you to develop exciting use-cases on the blockchain.

Structure of a block

Blocks hold batches of valid transactions hashed and encoded into a Merkle Tree. Each block holds the cryptographic hash of the previous block linking the two. The linked blocks form a chain. The iterative process ensures the integrity of the previous block, all the way to the initial block, which is called the **genesis block.** To ensure the integrity of each block, it is usually digitally signed.

Typically, a block consists of the following information (in the bitcoin blockchain):

- Information of the sender;
- Information of the receiver;
- Information of the amount transacted;

- A cryptographic hash of this specific block;
- A cryptographic hash of the previous block.

How does a blockchain ensure security?

This section discusses some of the techniques with which a blockchain ensures that the validity of the blocks in the blockchain is maintained.

Proof-Of-Work

To ensure that the newly generated block is secure and built by a trustable entity, blockchain employs a mechanism called proof-of-work. Proof of work (PoW) is a form of cryptographic proof in which one party (the prover) proves to others (the verifiers) that a certain amount of a specific computational effort has been expended. Verifiers can subsequently confirm this expenditure with minimal effort on their part. One of the critical features of the Proof-Of-Work algorithm is that there should be an asymmetrical work requirement on the part of the prover and the verifier i.e., the work required by the prover should be computationally high. In contrast, the work required by the verifier should be computationally light. The purpose of proof-of-work algorithms is not to prove that certain work was carried out or that a computational puzzle was "solved" but deterring manipulation of data by establishing considerable energy and hardware-control requirements to be able to do so. Environmentalists have criticized Proof-of-work systems for their energy consumption. In the next section, let us discuss an application of the blockchain — Smart Contracts.

Smart Contracts

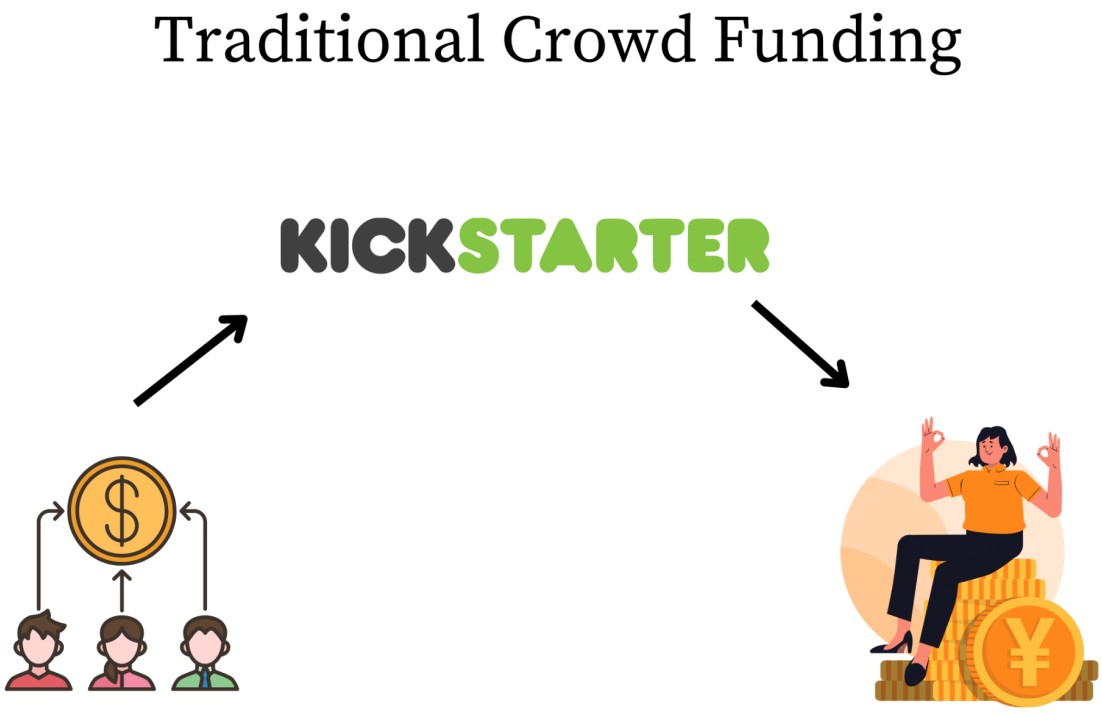
In 1997 Nick Szabo, a computer scientist, law scholar, and cryptographer, used the term Smart Contracts for the first time. One can understand smart contracts as simple contracts on top of transactions in the real world — the only difference being they are digital. A smart contract is abstracted as a computer program stored within a blockchain.

Kickstarter: An example of a Smart Contract

Kickstarter is an excellent example of an application that can be built using a smart contract. The typical application of a Kickstarter campaign is as follows: let us say that you have a project in mind — maybe you want to develop a productivity tool for software engineers and need to raise some capital to build it. You can go on Kickstarter and share the idea, put in a request for a specific amount of capital that you require, and people on the internet can vote for the project and pay for it. One of the constraints of the system is as follows:

"*Any project that is not fully funded goes to void, and money that the users contribute has to be returned.*"

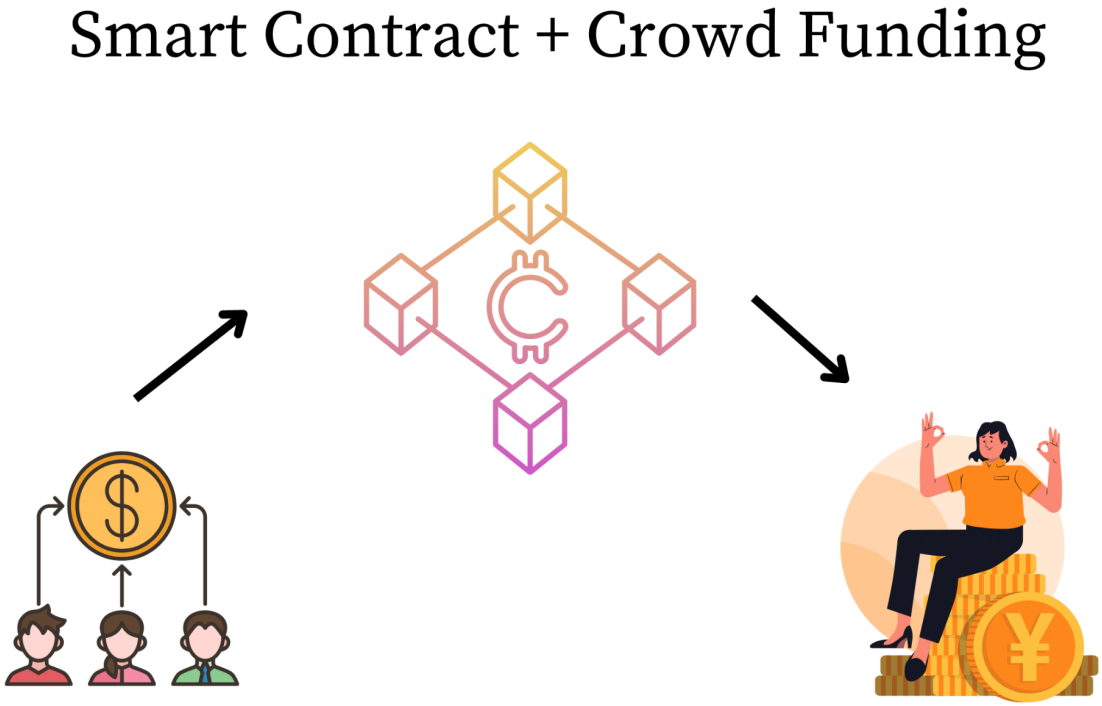
So, assume you want to raise $100,000 on your project for building a tool for software engineers. If you can raise only $90,000, then the project is voided, and the money has to be returned to the contributors. However, let us say that you can raise $100,000 or $200,000, or $1,000,000, then the money will go into your bank account, and you use it to build the system.



Today, Kickstarter acts as an intermediary (a third party) between the project developer and the contributor to ensure that the system's constraints are met. However, one can argue that it may happen that the third party goes rogue in the future. It can certainly happen. How do we ensure that the system remains foolproof in that scenario? That is where Smart Contracts come into the picture.

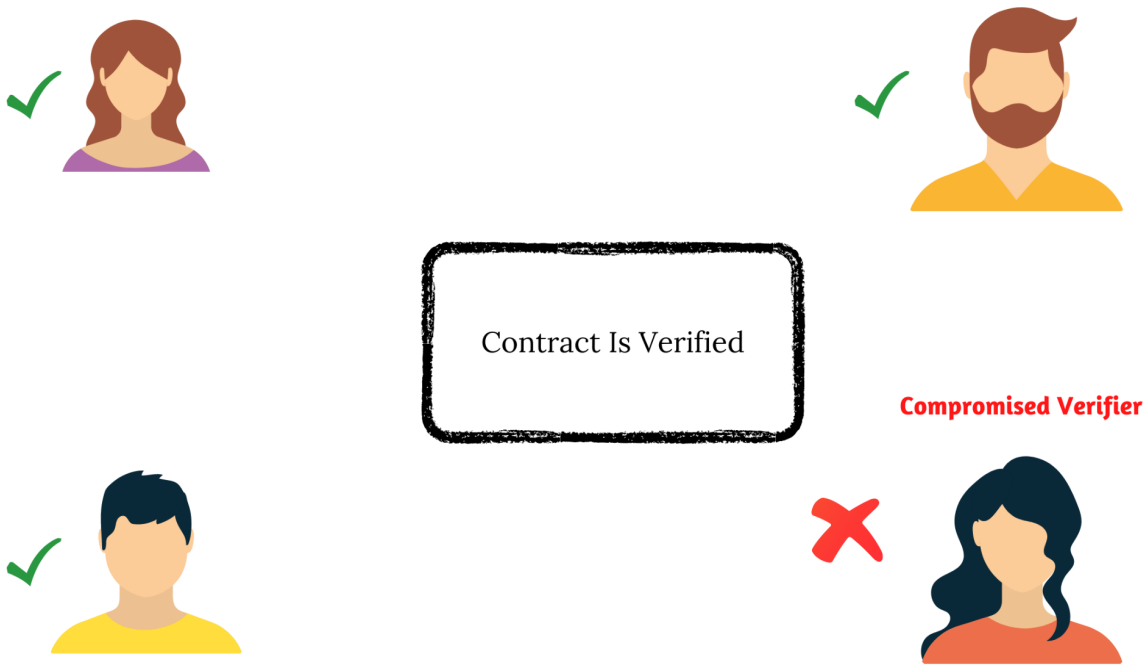
Kickstarter - With Smart Contracts

With the help of a blockchain, the developer of the project can write a smart contract where the following constraint can be met (in a fully distributed, digital manner):
"*Allow the money to be transferred to the project developer's wallet only if the money exceeds the target amount.*"



Here are the steps that occur when the project is funded using a Smart Contract:

1. The project owner creates the Smart Contract that sits in the middle of the contributors and the project owner. (See Fig. XXX).
2. The funders put money into the Smart Contract that sits on the distributed ledger in the middle of the funders and the project owner.
3. The Smart Contract will hold all the received funds in an escrow account until the goal is reached. The escrow account may be a crypto wallet.
4. If the amount of money contributed exceeds the goal, the contract succeeds, and the money is transferred to the project owner's account.
5. If the timebound set on the fundraises exceeds, the money is sent back to the individual contributor's account and voided the contract.



Why can we trust the Smart Contract?

A Smart Contract exhibits the following properties due to which it can be trusted:

- **Immutable**: Being immutable ensures that it can never be changed again once a Smart Contract is created. Remember that this flows from the design constraint that blocks in a blockchain are immutable. Once a block has been constructed on a blockchain, it can never be tampered with - without changing its hash function. In case a given block's hash function is tampered with. It results in the invalidity of the complete blockchain since the blockchain is secured.
- **Distributed**: The output of the contract is validated by everyone in the network. Let us say that even if a few entities in the network are compromised and say that the contract is met even if sufficient money has not been transferred (the case of fraud). In that scenario, the vote of the majority counts, and unless a majority of the users are compromised, the smart contract remains valid.

In all, we can say that a blockchain remains secure, and therefore, a Smart Contract is pretty secure, and tampering with smart contracts is almost impossible. If you want to go through the concepts on YouTube, I found this lecture very useful.

If you ❤️ reading this article, you can find me 👇👇

Newsletter On Substack
Blog Posts On Medium
Profile Page On LinkedIn

"Education is the most powerful weapon we can use to change the world." — Nelson Mandela.