A
Project Report
On

“IMPLEMENTATION OF BLOCKCHAIN ARCHITECTURE”

Submitted in partial fulfillment of
the requirements for the 8th Semester Sessional Examination of

BACHELOR OF TECHNOLOGY
IN

COMPUTER SCIENCE & ENGINEERING

By

HRITIK KUMAR (1701210329)
SURENDRA BISOYI (1701210181)
A LAKSHMI NARASIMHA (1701210142)
AISHIK BHATTACHARJEE (1701210146)

Under the esteemed guidance of

DR. RAGHVENDRA KUMAR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GANDHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

GUNUPUR – 765022

2020 - 21

Gandhi Institute of Engineering & Technology

GUNUPUR – 765 022, Dist: Rayagada (Orissa), India
(Approved by AICTE, Govt. of Orissa and Affiliated to Biju Patnaik University of Technology)
☎: 06857 – 250172(Office), 251156(Principal), 250232(Fax),
e-mail: gandhi_giet@yahoo.com visit us at www.giet.org

ISO 9001:2000
Certified Institute

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project work entitled "Implementation of Blockchain Architecture" is done by Name- Hritik Kumar(1701210329), Surendra Bisoyi (1701210181), A Lakshmi Narasimha(1701210142) & Aishik Bhattacharjee(1701210146) in partial fulfillment of the requirements for the 8th Semester Sessional Examination of Bachelor of Technology in Computer Science and Engineering during the academic year 2020-21. This work is submitted to the department as a part of evaluation of 8th Semester Project.

Name
Dr. Raghvendra Kumar

Prof. (Dr) .Sanjay Kumar Kuanar
HoD, CSE

Acknowledgement

We would like to express our special thanks of gratitude to our guide ***Dr. Raghvendra Kumar*** as well as our ***HOD Dr. Sanjay Kumar Kuanar*** who gave us the golden opportunity to do this wonderful project on the topic ***IMPLEMENTATION OF BLOCKCHAIN ARCHITECTURE***, which also helped us in doing a lot of Research and we came to know about so many new things we are really thankful to them. Secondly, we would also like to thank our ***Teachers*** and ***Teammates*** who helped us a lot in finalizing this project within the limited time frame.

Abstract

Blockchain technology is revolutionary. It will make life simpler and safer, changing the way personal information is stored and how transactions for good and services are made. Blockchain technology creates a permanent and immutable record of every transaction. This impenetrable digital ledger makes fraud, hacking, data theft, and information loss impossible. The technology will affect every industry in the world, including manufacturing, retail, transportation, healthcare, and real estate Companies as Google, IBM, Microsoft, American Express, Walmart, Nestle, Chase, Intel, Hitachi, and Dole are all working to become early adopters of blockchain. Nearly $400 trillion across various industries is set to be transformed by blockchain.

Table of Content

Introduction

History of Blockchain

The blockchain technology was described in **1991** by the research scientist **Stuart Haber** and **W. Scott Stornetta**.

They wanted to introduce a computationally practical solution for **time-stamping digital documents** so that they could not be **backdated or tampered**.
They develop a system using the concept of cryptographically secured chain of blocks to store the time-stamped documents.

In **1992**, Merkle Trees were incorporated into the design, which makes blockchain more efficient by allowing several documents to be collected into one block. **Merkle Trees** are used to create a **'secured chain of blocks'**.
It stored a series of data records, and each data records connected to the one before it. The newest record in this chain contains the history of the entire chain. However, this technology went unused, and the patent lapsed in 2004.

In **2004**, computer scientist and cryptographic activist **Hal Finney** introduced a system called **Reusable Proof Of Work (RPoW)** as a prototype for digital cash. It was a significant early step in the history of cryptocurrencies. The RPoW system worked by receiving a non-exchangeable or a non-fungible Hashcash based proof of work token in return, created an **RSA-signed** token that further could be transferred from person to person.

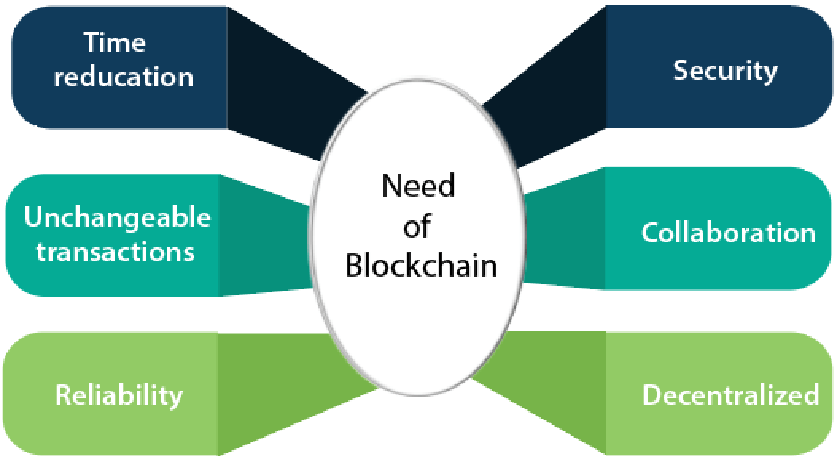
RPoW solved the double-spending problem by keeping the ownership of tokens registered on a trusted server. This server was designed to allow users throughout the world to verify its correctness and integrity in real-time.

Further, in **2008**, **Satoshi Nakamoto** conceptualized the theory of **distributed blockchains**. He improves the design in a unique way to add blocks to the initial chain without requiring them to be signed by trusted parties. The modified trees would contain a secure history of data exchanges.It utilizes a peer-to-peer network for timestamping and verifying each exchange. It could be managed autonomously without requiring a central authority. These improvements were so beneficial that makes blockchains as the backbone of cryptocurrencies. Today, the design serves as the public ledger for all transactions in the cryptocurrency space.

The evolution of blockchains has been steady and promising. The words block and chain were used separately in Satoshi Nakamoto's original paper but were eventually popularized as a single word, the Blockchain, by **2016**.

In recent time, the file size of cryptocurrency blockchain containing records of all transactions occurred on the network has grown from **20 GB** to **100 GB**.

Need of Blockchain



Blockchain technology has become popular because of the following:

- **Time reduction:** In the financial industry, blockchain can allow the quicker settlement of trades. It does not take a lengthy process for verification, settlement, and clearance. It is because of a single version of agreed-upon data available between all stakeholders.

- **Unchangeable transactions:** Blockchain register transactions in a chronological order which certifies the unalterability of all operations, means when a new block is added to the chain of ledgers, it cannot be removed or modified.

- **Reliability:** Blockchain certifies and verifies the identities of each interested parties. This removes double records, reducing rates and accelerates transactions.

- **Security:** Blockchain uses very advanced cryptography to make sure that the information is locked inside the blockchain. It uses Distributed Ledger Technology where each party holds a copy of the original chain, so the system remains operative, even the large number of other nodes fall.

- **Collaboration:** It allows each party to transact directly with each other without requiring a third-party intermediary.

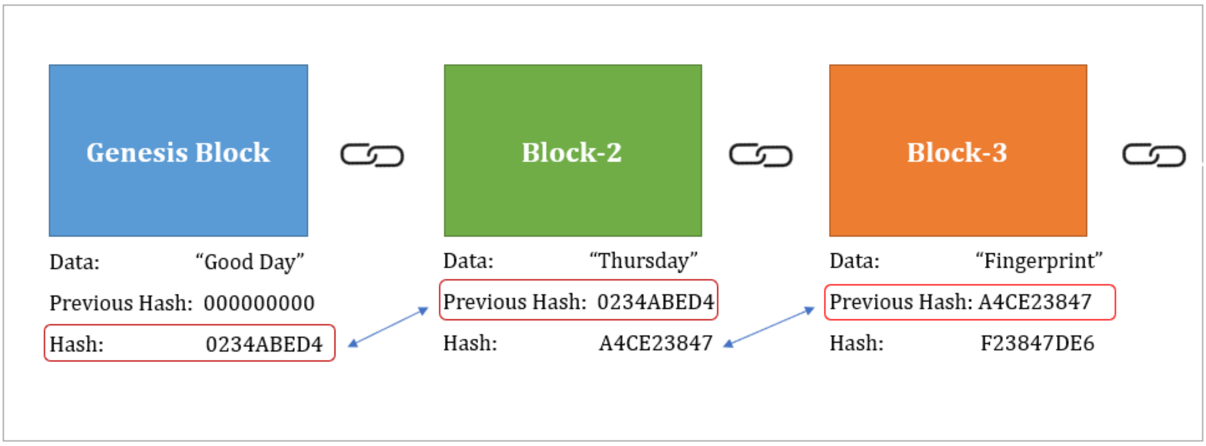
- **Decentralized:** It is decentralized because there is no central authority supervising anything. There are standards rules on how every node exchanges the blockchain information. This method ensures that all transactions are validated, and all valid transactions are added one by one.

What is Blockchain?

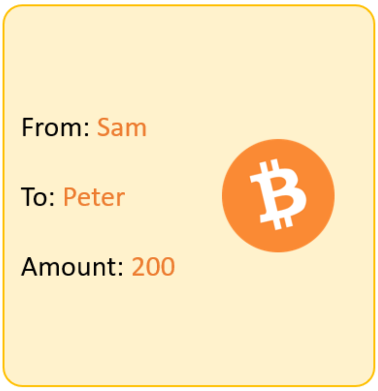
A Blockchain is a continuously growing list of records called blocks, which are linked and secure using Cryptography.

Each Block has

1. Data
2. Hash
3. Hash of the previous block



For Example: *A Bitcoin Block contains information about the Sender, Receiver, number of bitcoins to be transferred.*



Bitcoin Block Example

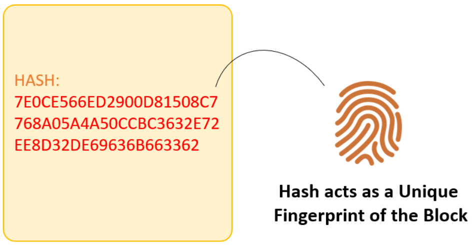
Genesis Block: The first block in the chain is called the **Genesis block**. Each new block in the chain is linked to the previous block.

SHA256 – Hash:

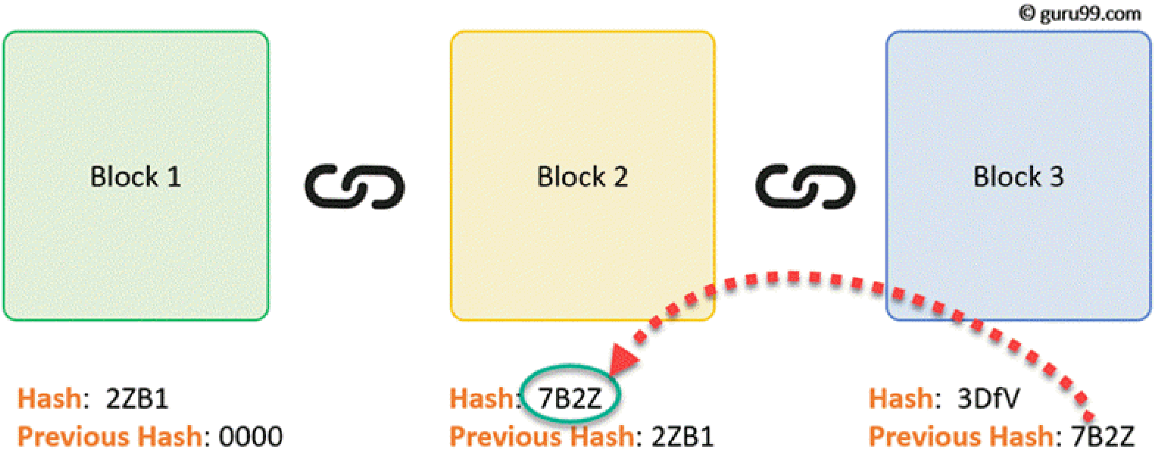
A **cryptographic hash** (sometimes called 'digest') is a kind of 'signature' for a text or a data file. SHA-256 generates an almost-unique 256-bit (32-byte) signature for a text.

A block also has a hash. can be understood as a fingerprint which is unique to each block. It identifies a block and all of its contents, and it's always unique, just like a fingerprint. So once a block is created, any change inside the block will cause the hash to change.

Therefore, the hash is very useful when you want to detect changes to intersections. If the fingerprint of a block changes, it does not remain the same block.

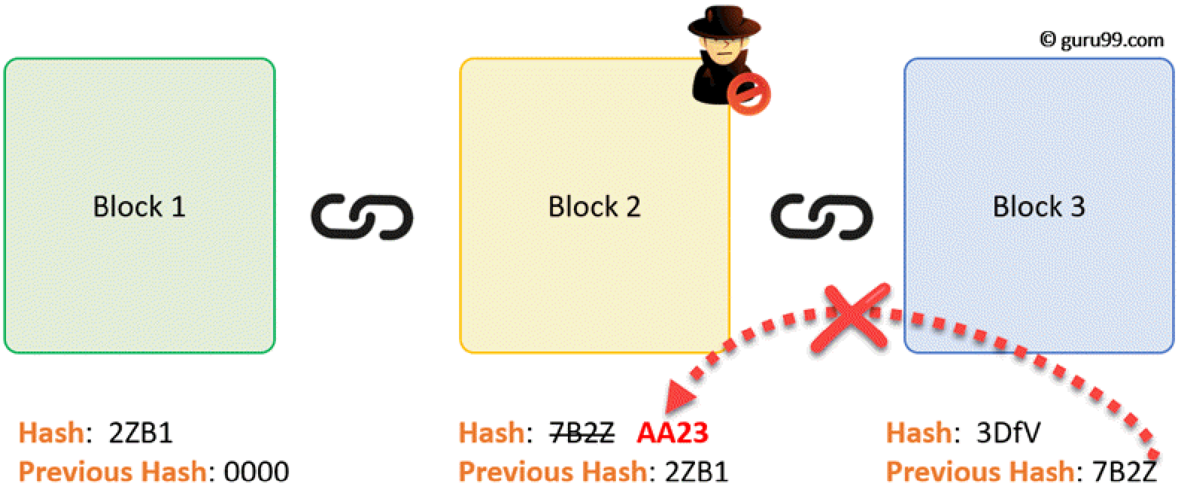


Consider following example, where we have a chain of 3 blocks. The 1st block has no predecessor. Hence, it does not contain has the previous block. Block 2 contains a hash of block 1. While block 3 contains Hash of block 2.



Hence, all blocks are containing hashes of previous blocks. This is the technique that makes a blockchain so secure. Let's see how it works -

Assume an attacker is able to change the data present in the Block 2. Correspondingly, the Hash of the Block also changes. But, Block 3 still contains the old Hash of the Block 2. This makes Block 3, and all succeeding blocks invalid as they do not have correct hash the previous block.



Therefore, changing a single block can quickly make all following blocks invalid.

Immutable Ledger:

The word Immutable means “cannot be changed.” And ledger is a fancy term for record, a record of something. Therefore an Immutable Ledger is a record that cannot be changed.

In the digital age we need data security and proof that the data has not been altered -- that’s the only way we can trust the digital data.

Such trust and proof of trust is very necessary when we are tracking transactions of money. Imagine if you sent me an electronic funds transfer of $1000 without any proof that you sent it and no way to verify that you sent it. I would not believe you until the money showed up, and if it didn’t show up--what then?

All banks and credit card processing systems use some sort of ledger to keep track of all the transactions that happen. But what if we don’t want to trust a big corporation, bank, or government with our money? Who do we trust?

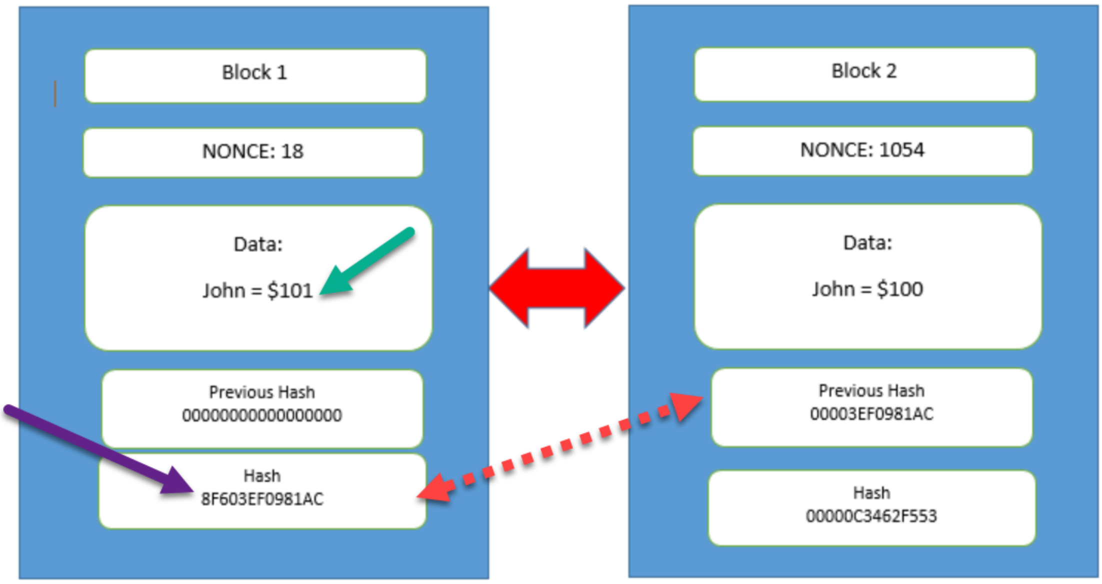
Blockchain technology introduced the Immutable Ledger. It’s based on math. You put your trust in that math, knowing no one can alter it or change it.

But how does blockchain ensure immutability of the ledger?

The foundation is what’s called the hash. The hash is like a digital signature and if a hacker tries to alter anything in the ledger, its hash will change.

Once the hash changes and no longer matches the previous hash in the ledger, the blockchain will reject that hash (making it null and void like a bad check). The hacker would have to change the next block, and the block after that, and basically the entire blockchain.

They can’t do that because a copy of the blockchain resides on multiple computers around the world. The hacker would need to make all of these changes simultaneously -- hacking into every computer all around the world. This is quite literally impossible. And therefore the ledger becomes immutable -- unable to be changed.

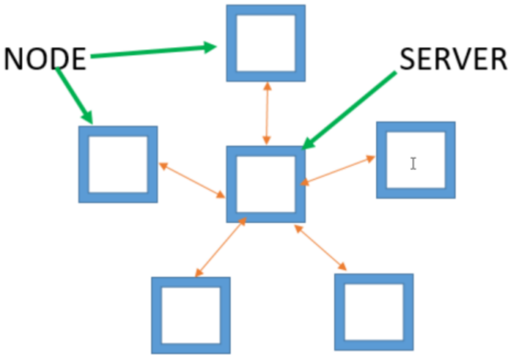


P2P Distributed Networks:

Blockchain is the concept of a P2p distributed network. This is the physical architecture that allows Blockchain to work and provides a blockchain with redundancy.

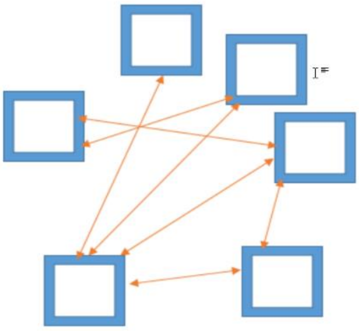
The P2P in P2P distributed network stands for peer to peer, indicating a network comprised of peers. The majority of computer networks in place right now are what is known as Server/Client networks.

In the picture below, the center square represents a server, with the boxes around it representing nodes (or in your case, the computer/tablet/phone you are reading this on). When you want to view a web page, you send a request from your node to a server. The server will then respond with the requested information.



While this works well, it does have some drawbacks. First off, since the server is central point of communication and the holder of all the information (webpages, databases, etc), if the server goes down, the network is essentially dead. This is the whole idea behind one of the more successful methods of cyber attack – the Denial of Service in which a server is targeted with more traffic than it can handle, shutting it down. You will often see it called a Distributed Denial of Server of DDoS as in order to hit the server with enough traffic to break it, hackers use multiple computers synced to deliver enough requests to the server all at the same time, overwhelming it. In other words, the attack is “distributed” across multiple computers.

Blockchain does not use a server client approach. Instead it uses a P2p or peer to peer network to function. In a peer to peer, the nodes (laptops, tablets, etc) all talk directly to each other. Instead of a server holding all the information, the data that makes up the blockchain is instead distributed across all the different nodes. So the more nodes that are part of the blockchain, the more copies of it that exist.



This works great for redundancy as even if you took out a couple of nodes in the network, it would still be able to function as normal. even if you were able to hack in and corrupt the blockchain in one of the nodes, the fact that copies of it exist on all the other nodes protect it from corruption.

How Mining Works?

Mining, at its simplest form, just means successfully adding a new block to the blockchain.

So why can't you just add a new block like you add a new element to a list or array in any other programming language? This has to do with the decentralized nature of blockchain. Since there is no centralized authority, blockchains rely on group consensus to verify that a new block added to the chain is valid. Keep in mind, this group is made up of anonymous nodes all over the world who do not know each other, and have no good reason to trust one another.

This block with contain the following items:

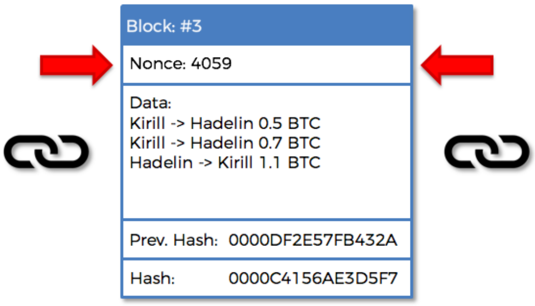
1. Block Number - Just the next number in the line

2. Previous Hash - This is the hash value of the current last block in the chain

3. Transactions - They will fill the block with verified transactions from the queue

4. A timestamp - The timestamp or timestamp is a small data stored in each block as a unique serial and whose main function is to determine the exact moment in which the block has been mined and validated by the blockchain network.

5. The Nonce - It stands for Number used only once. It is the field where the mining is all about. It gives us extra control and flexibility. Nonce is a number. It can go up to 4 Billion, as we change the value of nonce the value of hash changes completely which shows avalanche effect.



Block: #3

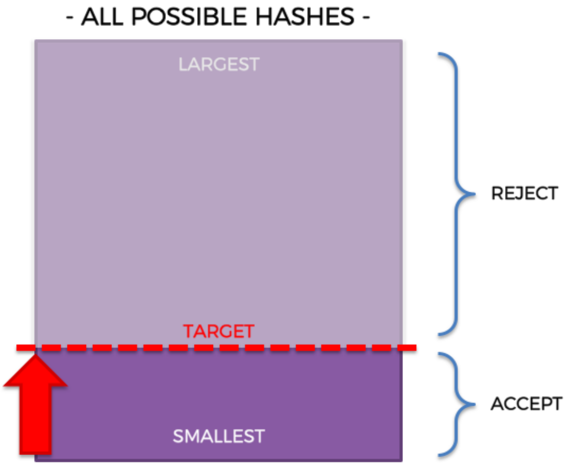
Nonce: 4059

Data:
Kirill -> Hadelin 0.5 BTC
Kirill -> Hadelin 0.7 BTC
Hadelin -> Kirill 1.1 BTC

Prev. Hash: 0000DF2E57FB432A

Hash: 0000C4156AE3D5F7

There is a total of 16^{64} possible SHA256 cryptographic hash values (each hexadecimal digit has 16 possible values and there are 64 of them in a hash). However, not all of them are valid hashes. Why is that? Well, every two weeks the Bitcoin network will define a minimal target for the hash. Anything above this target will be rejected, anything below — accepted.



The diagram above illustrates the pool (not to be confused with ‘mining pool’) of all possible SHA256 hashes — starting at the bottom with smallest and increasing towards the largest at the top. Somewhere along the vertical we have the target.

At the time of writing the target is:

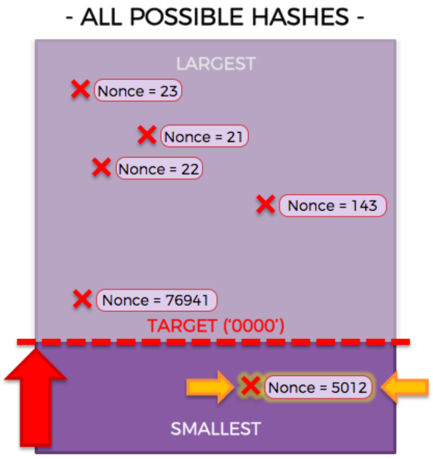
0000000000000000005d97dc0000000000000000000000000000000000000000

What is really important in the target is the number of leading zeroes. Just like in the decimal system, leading zeros in a fixed-size number will determine its magnitude. Every leading zero reduces the number’s magnitude by a factor of 16 (ten in the decimal system, but here we’re working with hexadecimals).There are 18 leading zeros in the current target, meaning that the number of total valid hashes is 16^{46} (only 64-18=46 non-zero digits remain). Therefore, the probability that a randomly picked hash is valid can be calculated as:

16^{46} / 16^{64} = 16^(-18) = 0.000000000000000000002%

In Bitcoin mining terms, this is the probability that any given Nonce value will generate a valid hash for the current block. We can now see why the diagram is out of proportion: the pool of valid hashes in reality is extremely small in comparison to the complete SHA256 pool.

And that’s what the cryptographic puzzle is all about: miners compete to find a Nonce (also called a Golden Nonce) which will generate a valid hash for the upcoming block. Whoever finds it first is allowed to add the block to the chain and get’s their reward of 12.5 Bitcoins. At the time of writing one Bitcoin is worth around $10,000 USD making mining a rather worthwhile activity.



The target is defined based on the network's hashrate (aggregate computational power of all Bitcoin miners). The more miners join the network — the lower the target will be, and therefore the harder it will be to find a suitable hash. The goal of this difficulty algorithm is to ensure that only one new block to is added every 10 minutes. This is part of the Bitcoin monetary policy to control the total number of coins in circulation.

In a nutshell, that's what the millions and millions of mining machines are doing day and night — they are simply iterating different values of the Nonce in hopes of being the first to find a valid hash for the next block. Once a valid hash if found, the block is added to the chain and the race starts over again, this time for the next block.

Byzantine Fault Tolerance:

Byzantine Fault Tolerance is the characteristic which defines a system that tolerates the class of failures that belong to the Byzantine Generals' Problem.

Byzantine Fault Tolerance(BFT) is the feature of a distributed network to reach consensus(agreement on the same value) even when some of the nodes in the network fail to respond or respond with incorrect information. The objective of a BFT mechanism is to safeguard against the system failures by employing collective decision making(both – correct and faulty nodes) which aims to reduce to influence of the faulty nodes. BFT is derived from Byzantine Generals' Problem.

Imagine that several divisions of the Byzantine army are camped outside an enemy city, each division commanded by its own general. The generals can communicate with one another only by messenger. After observing the enemy, they must decide upon a common plan of action. However, some of the generals may be traitors, trying to prevent the loyal generals from reaching an agreement. The generals must decide on when to attack the city, but they need a strong majority of their army to attack at the same time. The generals must have an algorithm to guarantee that (a) all loyal generals decide upon the same plan of action, and (b) a small number of traitors cannot cause the loyal generals to adopt a bad plan. The loyal generals will all do what the algorithm says they should, but the traitors may do anything they wish. The algorithm must guarantee condition (a) regardless of what the traitors do. The loyal generals should not only reach agreement, but should agree upon a reasonable plan.

Byzantine fault tolerance can be achieved if the correctly working nodes in the network reach an agreement on their values. There can be a default vote value given to missing messages i.e., we can assume that the message from a particular node is 'faulty' if the message is not received within a certain time limit. Furthermore, we can also assign a default response if the majority of nodes respond with a correct value.

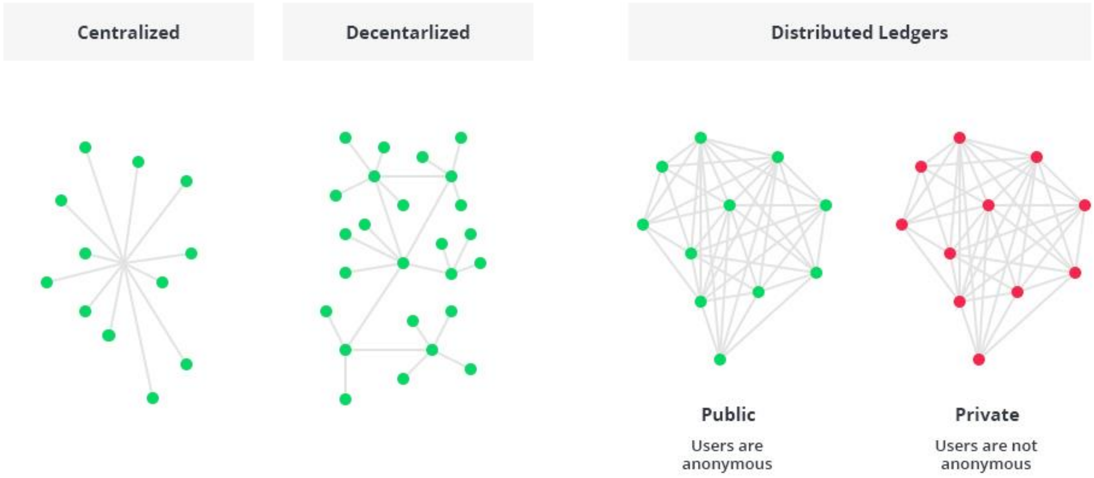
Leslie Lamport proved that if we have 3m+1 correctly working processors, a consensus(agreement on same state) can be reached if atmost m processors are faulty which means that strictly more than two-thirds of the total number of processors should be honest.

Types of Byzantine Failures:

There are two categories of failures that are considered. One is fail-stop(in which the node fails and stops operating) and other is arbitrary-node failure. Some of the arbitrary node failures are given below :

- Failure to return a result
- Respond with an incorrect result
- Respond with a deliberately misleading result
- Respond with a different result to different parts of the system

Blockchain Architecture



In other words, blockchain is a combination of computers linked to each other instead of a central server, meaning that the whole network is decentralized.

To make it even simpler, the blockchain concept can be compared to work done with Google Docs. You may recall the days of tossing over doc. documents and waiting for other participants to make necessary edits. These days, with the help of Google Docs, it is possible to work on the same document simultaneously.

The blockchain technique allows digital information to be distributed, rather than copied. This distributed ledger provides transparency, trust, and data security.

Blockchain architecture is being used very broadly in the financial industry. However, these days, this technology helps create software development solutions for cryptocurrencies and record keeping, digital notary, and smart contracts.

Components of the Blockchain Architecture

The blockchain architecture has many business benefits.

Here are some built-in characteristics:

- ***Cryptography*** -- Blockchain transactions are verified and trustworthy because of complex computations and cryptographic proof between the parties.

- ***Immutability*** -- Records in a blockchain can't be modified or deleted.

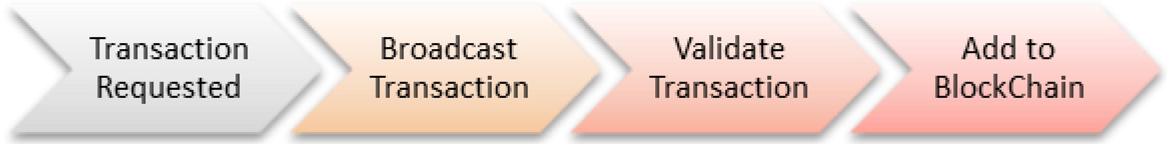
- ***Provenance*** -- It's possible to trace the origin of each transaction in the blockchain ledger.

- ***Decentralization*** -- Every member of the blockchain structure is able to access the entire distributed database. Unlike in a centralized system, a consensus algorithm is responsible for network management.

- ***Anonymity*** -- Every member of the blockchain network has a generated address, not a user ID. This preserves the anonymity of users, especially in a public blockchain.

- ***Transparency*** -- The blockchain system is unlikely to be damaged as it takes enormous computing power to completely rewrite the blockchain network.

How Blockchain Transaction Works?



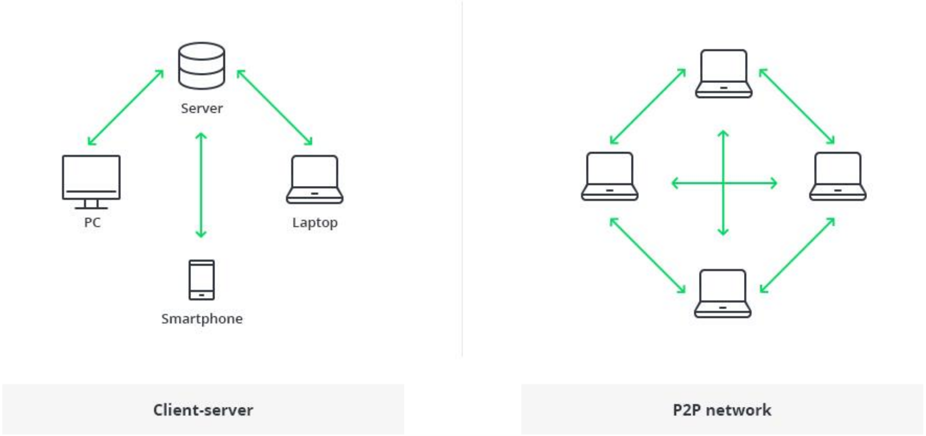
Step 1) Some person requests a transaction. The transaction could be involved cryptocurrency, contracts, records or other information.

Step 2) The requested transaction is broadcasted to a P2P network with the help of nodes.

Step 3) The network of nodes validates the transaction and the user's status with the help of known algorithms.

Step 4) Once the transaction is complete the new block is then added to the existing blockchain. In such a way that is permanent and unalterable.

Database vs. Blockchain Architecture



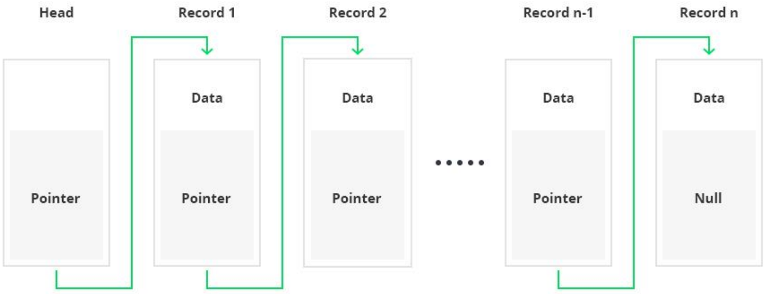
The traditional architecture of the World Wide Web uses a client-server network. In this case, the server keeps all the required information in one place so that it is easy to update, due to the server being a centralized database controlled by a number of administrators with permissions.

In the case of the distributed network of blockchain architecture, each participant within the network maintains, approves, and updates new entries. The system is controlled not only by separate individuals, but by everyone within the blockchain network. Each member ensures that all records and procedures are in order, which results in data validity and security. Thus, parties that do not necessarily trust each other are able to reach a common consensus.

To summarize things, the blockchain is a decentralized, distributed ledger (public or private) of different kinds of transactions arranged into a P2P network. This network consists of many computers, but in a way that the data cannot be altered without the consensus of the whole network (each separate computer).

The structure of blockchain technology is represented by a list of blocks with transactions in a particular order. These lists can be stored as a flat file (txt. format) or in the form of a simple database. Two vital data structures used in blockchain include:

- Pointers - variables that keep information about the location of another variable. Specifically, this is pointing to the position of another variable.
- Linked lists - a sequence of blocks where each block has specific data and links to the following block with the help of a pointer.



Blockchain architecture can serve the following purposes for organizations and enterprises:

- Cost reduction - lots of money is spent on sustaining centrally held databases (e.g. banks, governmental institutions) by keeping data current secure from cyber crimes and other corrupt intentions.

- History of data - within a blockchain structure, it is possible to check the history of any transaction at any moment in time. This is a ever-growing archive, while a centralized database is more of a snapshot of information at a specific point.

- Data validity & security - once entered, the data is hard to tamper with due to the blockchain's nature. It takes time to proceed with record validation, since the process occurs in each independent network rather than via compound processing power. This means that the system sacrifices performance speed, but instead guarantees high data security and validity.

Use Cases of Blockchain

1. Smart Contracts

Smart contracts Blockchain-based contracts enforced in real-time. They are created as an agreement between two or more parties without the involvement of any intermediary. The contract exists across a distributed and decentralized Blockchain network. Smart contracts are now a staple in healthcare, real estate, and even for government agencies.

2. Internet of Things (IoT)

The Internet of Things (IoT) industry is growing rapidly with billions of connected devices. The latest forecasts suggest that by 2030, there'll be 50 billion devices in use globally. As this number continues to grow, it will increase vulnerabilities as hackers can easily breach your data through a single connected device. By integrating Blockchain technology in IoT devices, the possibility of data breaches can be reduced to a great extent.

3. Money Transfer

Money transfer and payment processing are the most excellent Blockchain technology use cases. Blockchain tech enables lightning-fast transactions in real-time. This has already transformed the BFSI sector for good as it saves both time and money (mostly eliminates transaction fees charged by banks/financial institutions).

4. Personal Identity Security

Personal identity theft and hacking are hot crimes in the cybercrime domain. In 2019, nearly 14.4 million people fell victim to identity fraud, which roughly translates to about 1 in 15 people. From hacking and violating personal files to forging documents, identity theft comes in many different forms. Blockchain can help combat this menace by storing crucial personal information (for example, social security numbers, birth certificates, address, PAN, etc.) on a decentralized and immutable ledger.

5. Logistics

Data siloing and lack of communication and transparency are the most pertinent issues of the logistics industry. Such obstacles become even more pronounced since thousands of companies operate in this domain, costing business time and money. This is where Blockchain's data transparency comes in handy. Blockchain tech can acknowledge data sources and automate processes, thereby building greater trust and transparency within the logistics industry.

6. Digital Media

Digital media companies are burdened with many challenges like data privacy, piracy of intellectual property, royalty payments, and copyright infringement, among other issues. By incorporating Blockchain technology into the digital media infrastructure, companies can protect their intellectual property, maintain data integrity, target the right customers, and ensure that artists receive their royalty payments in due time.

Benefits of Blockchain

Better Transparency

Transparency is one of the big issues in the current industry. To improve transparency, organizations have tried to implement more rules and regulations. But there is one thing that doesn’t make any system 100% transparency, i.e., centralization.

With blockchain, an organization can go for a complete decentralized network where there is no need for a centralized authority, improving the system’s transparency.

Enhanced Security

Blockchain technology utilizes advanced security compared to other platforms or record-keeping systems.
Any transactions that are ever recorded needs to be agreed upon according to the consensus method. Also, each transaction is encrypted and has a proper link to the old transaction using a hashing method.

Reduced Costs

Right now, businesses spend a lot of money to improve to manage their current system. That’s why they want to reduce cost and divert the money into building something new or improving current processes.

By using blockchain, organizations can bring down a lot of costs associated with 3rd party vendors.

As blockchain has no inherited centralized player, there is no need to pay for any vendor costs. On top of that, there is less interaction needed when it comes to validating a transaction, further removing the need to spend money or time to do basic stuff.

True Traceability

With blockchain, companies can focus on creating a supply chain that works with both vendors and suppliers. In the traditional supply chain, it is hard to trace items that can lead to multiple problems, including theft, counterfeit, and loss of goods.With blockchain, the supply chain becomes more transparent than ever. It enables every party to trace the goods and ensure that it is not being replaced or misused during the supply chain process.

Organizations can also make the most out of blockchain traceability by implementing it in-house.

Improved Speed and Highly Efficient

The last industrial benefit that blockchain brings is improved efficiency and speed.
Blockchain solves the time-consuming process and automates them to maximize efficiency.
It also eradicates human-based errors with the help of automation.

The digital ledger makes everything this possible by providing a single place to store transactions.
The streamlining and automation of processes also mean that everything becomes highly efficient and fast.

Implementation of Blockchain

Technology Stack

- **Python 3:**

Python is a popular programming language.

It was created by Guido van Rossum, and released in 1991.

It is used for:

1. web development (server-side),
2. software development,
3. mathematics,
4. system scripting.

What can Python do?

1. Python can be used on a server to create web applications.
2. Python can be used alongside software to create workflows.
3. Python can connect to database systems. It can also read and modify files.
4. Python can be used to handle big data and perform complex mathematics.
5. Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

1. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
2. Python has a simple syntax similar to the English language.
3. Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
4. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
5. Python can be treated in a procedural way, an object-oriented way or a functional way.

Python Syntax compared to other programming languages

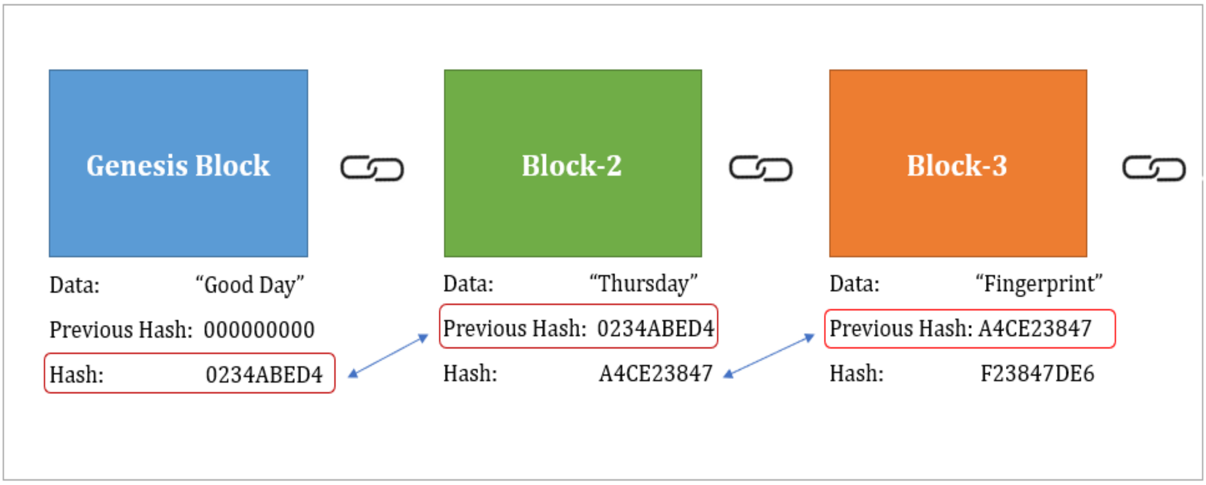
1. Python was designed for readability, and has some similarities to the English language with influence from mathematics.
2. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
3. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

- **Blockchain:**

A Blockchain is a continuously growing list of records called blocks, which are linked and secure using Cryptography.

Each Block has

1. Data
2. Hash
3. Hash of the previous block



Blockchain is a constantly growing **ledger** that keeps a **permanent** record of all the transactions that have taken place in a **secure**, **chronological**, and **immutable** way. It can be used for the secure transfer of money, property, contracts, etc. without requiring a third-party intermediary such as bank or government. Blockchain is a software protocol, but it could not be run without the Internet (like SMTP is for email).

1. **Ledger:** It is a file that is constantly growing.
2. **Permanent:** It means once the transaction goes inside a blockchain, you can put up it permanently in the ledger.
3. **Secure:** Blockchain placed information in a secure way. It uses very advanced cryptography to make sure that the information is locked inside the blockchain.
4. **Chronological:** Chronological means every transaction happens after the previous one.
5. **Immutable:** It means as you build all the transaction onto the blockchain, this ledger can never be changed.

Who uses the blockchain?

Blockchain technology can be integrated into multiple areas. The primary use of blockchains is as a distributed ledger for cryptocurrencies. It shows great promise across a wide range of business applications like Banking, Finance, Government, Healthcare, Insurance, Media and Entertainment, Retail, etc.

- **Flask**:

What is Web Framework?

Web Application Framework or simply Web Framework represents a collection of libraries and modules that enables a web application developer to write applications without having to bother about low-level details such as protocols, thread management etc.

What is Flask?

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO). It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework.

Features of Flask

Here, are important features of Flask

1. Integrated support for unit testing.
2. RESTful request dispatching.
3. Uses a Ninja2 template engine.
4. It is based on Werkzeug toolkit.
5. Support for secure cookies (client-side sessions).
6. Extensive documentation.
7. Google app engine compatibility.
8. APIs are nicely shaped and coherent
9. Easily deployable in production

Advantages of Flask

Here, are pros/benefits of using Flask

1. Higher compatibility with latest technologies
2. Technical experimentation
3. Easier to use for simple cases
4. Codebase size is relatively smaller
5. High scalability for simple applications,
6. Easy to build a quick prototype
7. Routing URL is easy
8. Easy to develop and maintain applications
9. Database integration is easy
10. Small core and easily extensible
11. Minimal yet powerful platform
12. Lots of resources available online especially on GitHub

Disadvantage of Flask

Here, are cons/drawback of Flask

1. Slower MVP development in most cases,
2. Higher maintenance costs for more complex systems
3. Complicated maintenance for larger implementations.
4. Async may be a little problem
5. Lack of database and ORM

CREATION OF BLOCKCHAIN USING PYTHON

Mining a new Block:

- The mining of the new block is done by finding the answer to the *proof of work*.
- To make mining hard the *proof of work* must be hard enough to get exploited.
- Mining a new Block means successfully adding a new block to the Blockchain.
- We keep iterating the *nonce* until you get the hash below the target.
- Once *nonce* gets below the target the block gets accepted.

Key Points:

Mining:

Mining is the process by which new bitcoin is added to the money supply.

Mining also serves to secure the bitcoin system against fraudulent transactions or transactions spending the same amount of bitcoin more than once, known as a double-spend. Miners provide processing power to the bitcoin network in exchange for the opportunity to be rewarded bitcoin.

Miners validate new transactions and record them on the global ledger. A new block, containing transactions that occurred since the last block, is "mined" every 10 minutes, thereby adding those transactions to the blockchain. Transactions that become part of a block and added to the blockchain are considered "confirmed," which allows the new owners of bitcoin to spend the bitcoin they received in those transactions.
To earn this reward, the miners compete to solve a difficult mathematical problem based on a cryptographic hash algorithm. The solution to the problem, called the proof of work, is included in the new block and acts as proof that the miner expended significant computing effort. The competition to solve the proof-of-work algorithm to earn reward and the right to record transactions on the blockchain is the basis for bitcoin's security model.

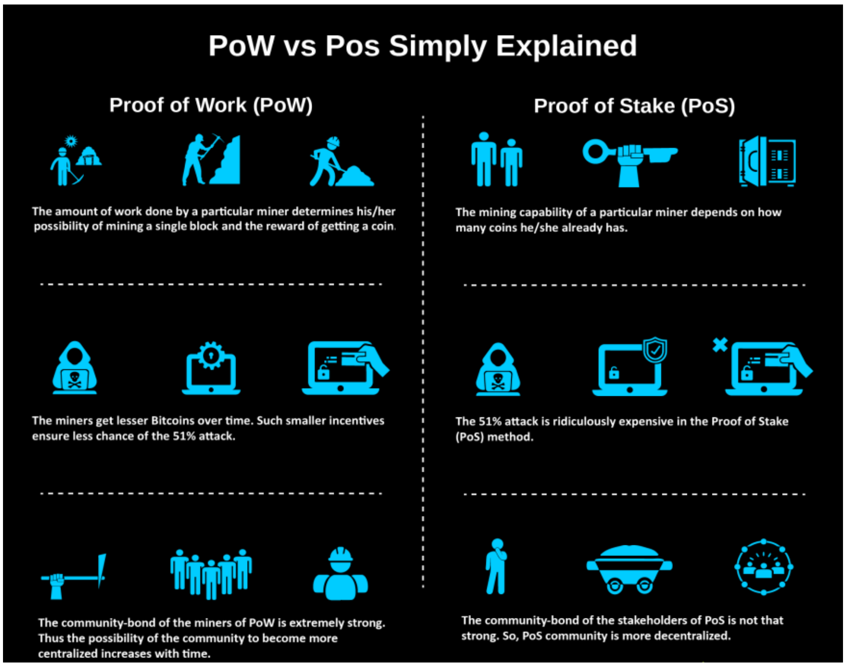
The process of new coin generation is called mining because the reward is designed to simulate diminishing returns, just like mining for precious metals. Bitcoin's money supply is created through mining, similar to how a central bank issues new money by printing bank notes. The amount of newly created bitcoin a miner can add to a block decreases approximately every four years (or precisely every 210,000 blocks). It started at 50 bitcoin per block in January of 2009 and halved to 25 bitcoin per block in November of 2012. It will halve again to 12.5 bitcoin per block sometime in 2016. Based on this formula, bitcoin mining rewards decrease exponentially until approximately the year 2140, when all bitcoin (20.99999998 million) will have been issued. After 2140, no new bitcoins will be issued.

Bitcoin miners also earn fees from transactions. Every transaction may include a transaction fee, in the form of a surplus of bitcoin between the transaction's inputs and outputs. The winning bitcoin miner gets to “keep the change” on the transactions included in the winning block. Today, the fees represent 0.5% or less of a bitcoin miner’s income, the vast majority coming from the newly minted bitcoins. However, as the reward decreases over time and the number of transactions per block increases, a greater proportion of bitcoin mining earnings will come from fees. After 2140, all bitcoin miner earnings will be in the form of transaction fees.

The word “mining” is somewhat misleading. By evoking the extraction of precious metals, it focuses our attention on the reward for mining, the new bitcoins in each block. Although mining is incentivized by this reward, the primary purpose of mining is not the reward or the generation of new coins. If you view mining only as the process by which coins are created, you are mistaking the means (incentives) as a goal of the process. Mining is the main process of the decentralized clearinghouse, by which transactions are validated and cleared. Mining secures the bitcoin system and enables the emergence of network-wide consensus without a central authority.

Mining is the invention that makes bitcoin special, a decentralized security mechanism that is the basis for peer-to-peer digital cash. The reward of newly minted coins and transaction fees is an incentive scheme that aligns the actions of miners with the security of the network, while simultaneously implementing the monetary supply.

Proof of Work:



PoW vs Pos Simply Explained

Proof of Work (PoW)

The amount of work done by a particular miner determines his/her possibility of mining a single block and the reward of getting a coin.

The miners get lesser Bitcoins over time. Such smaller incentives ensure less chance of the 51% attack.

The community-bond of the miners of PoW is extremely strong. Thus the possibility of the community to become more centralized increases with time.

Proof of Stake (PoS)

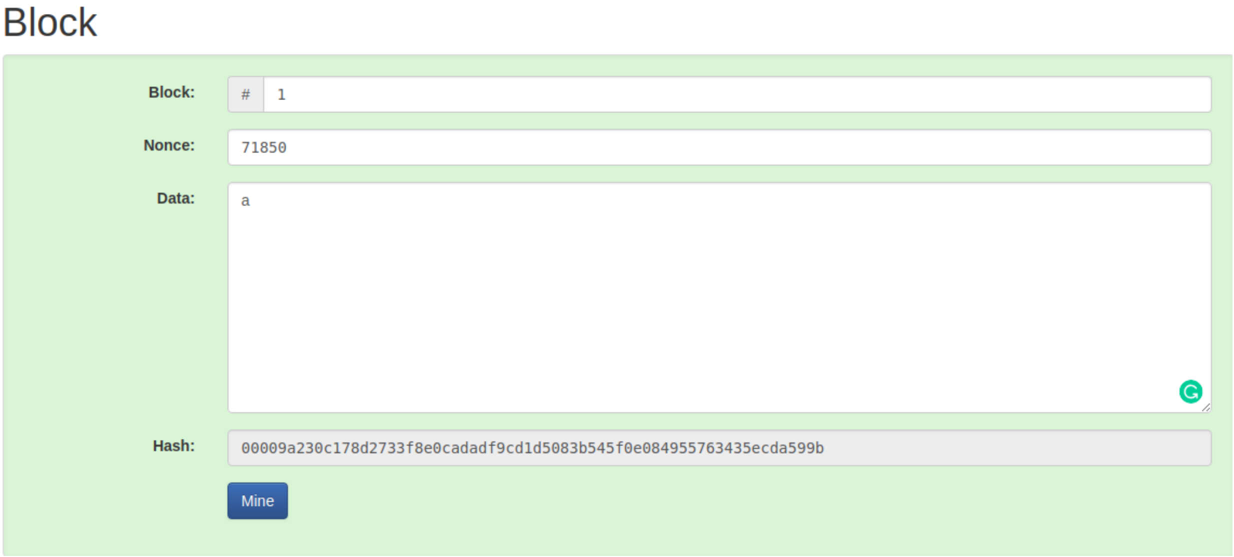
The mining capability of a particular miner depends on how many coins he/she already has.

The 51% attack is ridiculously expensive in the Proof of Stake (PoS) method.

The community-bond of the stakeholders of PoS is not that strong. So, PoS community is more decentralized.

Proof of Work (PoW) is the **original consensus algorithm** in a blockchain network. The algorithm is used to confirm the transaction and creates a new block to the chain. In this algorithm, **minors** (a group of people) compete against each other to complete the transaction on the network. The process of competing against each other is called **mining**. As soon as miners successfully created a valid block, he gets **rewarded**. The most famous application of Proof of Work (PoW) is Bitcoin.

Producing proof of work can be a random process with low probability. In this, a lot of **trial and error** is required before a valid proof of work is generated. The main working principle of proof of work is a mathematical puzzle which can easily prove the solution. Proof of work can be implemented in a blockchain by the Hashcash proof of work system.



Block

Block: # 1

Nonce: 71850

Data: a

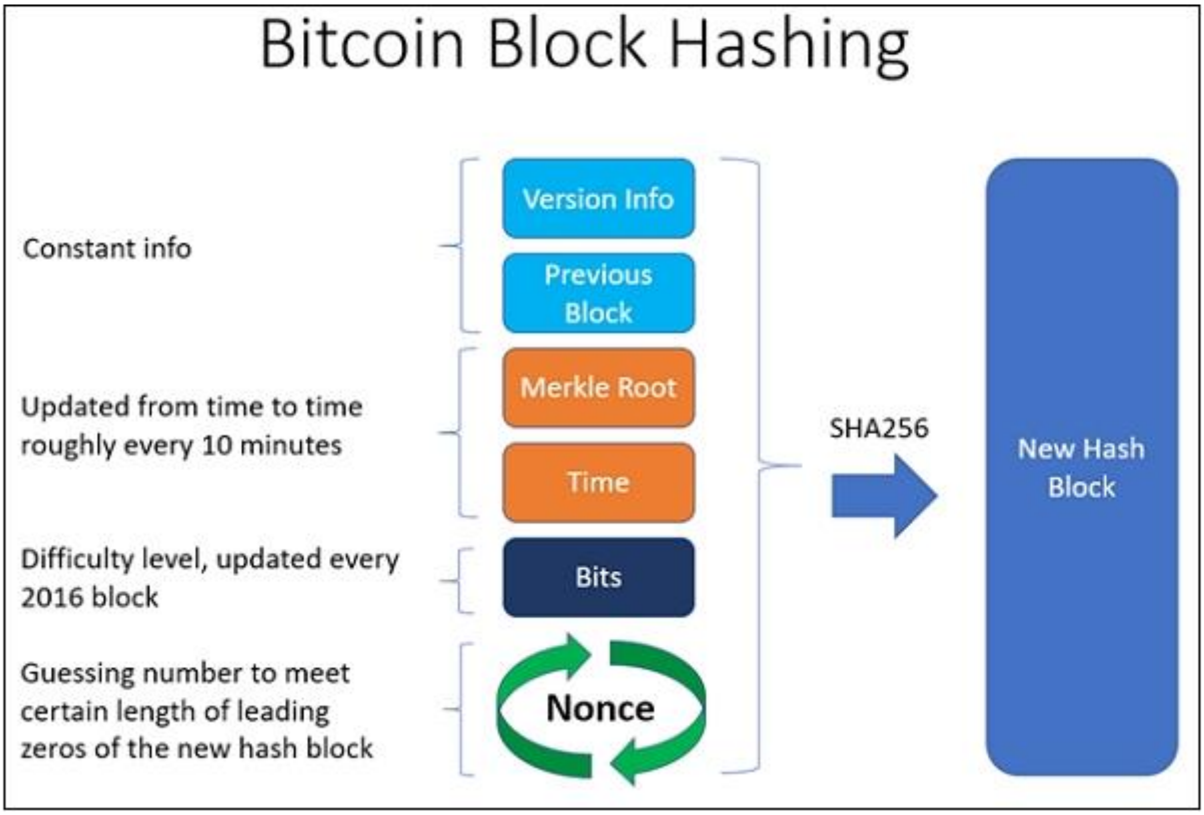
Hash: 00009a230c178d2733f8e0cadadf9cd1d5083b545f0e084955763435ecda599b

In the puzzle game, bitcoin software creates a challenge, and there is a game begins. This game involves all miners competing against each other to solve the challenges, and this challenge will take approximately 10 minutes to be completed. Every single miner starts trying to find the solution to that one Nonce that will satisfy the hash for the block. At some specific point, one of those miners in the global community with higher speed and great hardware specs will solve the cryptography challenge and be the winner of the game. Now, the rest of the community will start verifying that block which is mined by the winner. If the nonce is correct, it will end up with the new block that will be added to the blockchain. The concept of generating a block provides a clear explanation of proof of work (PoW).

Nonce:

Nonce is the central part of this Proof of Work. The Nonce is a random whole number, which is a 32-bit (4 byte) field, which is adjusted by the miners, so that it becomes a valid number to be used for hashing the value of block. **Nonce is the number which can be used only once**. Once the perfect Nonce is found, it is added to the hashed block. Along with this number, the hash value of that block will get rehashed and creates a difficult algorithm.

It is compared to the existing target, whether it is lower or equal to the current target. Miners test and discard millions of Nonce per second until they find that Golden Nonce which is valid. In order to complete the verification faster than other miners, miners compete with each other using their computer hashing power. Once the Golden Nonce is found, they can complete the Block and add it to the Block Chain and there by receive the Block reward.


Bitcoin Block Hashing
Constant info
Updated from time to time roughly every 10 minutes
Difficulty level, updated every 2016 block
Guessing number to meet certain length of leading zeros of the new hash block
Version Info
Previous Block
Merkle Root
Time
Bits
Nonce
SHA256
New Hash Block

This rules out the possibility of any duplication, or using the same bitcoin twice. Whether rest of the fields are changed or not, Nonce will change because it is unique and thus became the most important component of the Proof of Work. In cryptography, a **nonce** is an arbitrary number that can be used just once in a cryptographic communication. It is often a random number issued in an authentication protocol to ensure that, same communication is not reused.

There will be some constant information, timestamp, hash value with difficulty, and the nonce which when passed through Hash algorithm – SHA256 will become a new block. Here nonce plays a very important role. As we have already discussed, millions of nonce values are tried until the Golden Nonce is found.

The target hash value is defined as the difficulty and the iterative calculation of the hash value requires the miner’s computer resources. Only with the correct Nonce value, proof of work can be created and thus giving birth to a new Block in the Block chain.

Displaying a Block:

- The data will be stored and displayed in *JSON format* which is very easy to implement and easy to read.
- Each Block contains multiple transaction/data.
- Each and every minute multiple block are added and to differentiate one from other we will use *fingerprinting*.
- The *fingerprinting* is done by using hash and to be particular we will use the *SHA256 hashing algorithm*.
- Every block will contain its own hash and also the hash of the previous function so that it cannot get tampered.
- This *fingerprinting* will be used to chain the blocks together.

Key Points:

JSON:

JSON stands for **J**ava**S**cript **O**bject **N**otation

JSON is a lightweight format for storing and transporting data

JSON is often used when data is sent from a server to a web page

JSON is "self-describing" and easy to understand

JSON Syntax Rules

1. Data is in name/value pairs
2. Data is separated by commas
3. Curly braces hold objects
4. Square brackets hold arrays

JSON Objects

JSON objects are written inside curly braces.

Just like in JavaScript, objects can contain multiple name/value pairs:

```
{"firstName":"John", "lastName":"Doe"}
```

JSON Arrays

JSON arrays are written inside square brackets.

Just like in JavaScript, an array can contain objects:

```
"employees":[
   {"firstName":"John", "lastName":"Doe"},
   {"firstName":"Anna", "lastName":"Smith"},
   {"firstName":"Peter", "lastName":"Jones"}
]
```

In the example above, the object "employees" is an array. It contains three objects.

Each object is a record of a person (with a first name and a last name).

SHA-256:

The SHA-256 algorithm is one flavor of SHA-2 (Secure Hash Algorithm 2), which was created by the National Security Agency in 2001 as a successor to SHA-1. SHA-256 is a patented cryptographic hash function that outputs a value that is 256 bits long.

What is hashing? In encryption, data is transformed into a secure format that is unreadable unless the recipient has a key. In its encrypted form, the data may be of unlimited size, often just as long as when unencrypted. In hashing, by contrast, data of arbitrary size is mapped to data of fixed size. For example, a 512-bit string of data would be transformed into a 256-bit string through SHA-256 hashing.

In cryptographic hashing, the hashed data is modified in a way that makes it completely unreadable. It would be virtually impossible to convert the 256-bit hash mentioned above back to its original 512-bit form. So why would you want to create a scrambled message that can't be recovered? The most common reason is to verify the content of data that must be kept secret. For example, hashing is used to verify the integrity of secure messages and files. The hash code of a secure file can be posted publicly so users who download the file can confirm they have an authentic version without the contents of the file being revealed. Hashes are similarly used to verify digital signatures.

Password verification is a particularly important application for cryptographic hashing. Storing users' passwords in a plain-text document is a recipe for disaster; any hacker that manages to access the document would discover a treasure trove of unprotected passwords. That's why it's more secure to store the hash values of passwords instead. When a user enters a password, the hash value is calculated and then compared with the table. If it matches one of the saved hashes, it's a valid password and the user can be permitted access.

What role does SHA-256 hashing play in cybersecurity? SHA-256 is used in some of the most popular authentication and encryption protocols, including SSL, TLS, IPsec, SSH, and PGP. In Unix and Linux, SHA-256 is used for secure password hashing. Cryptocurrencies such as Bitcoin use SHA-256 for verifying transactions.

How secure is SHA-256?

SHA-256 is one of the most secure hashing functions on the market. The US government requires its agencies to protect certain sensitive information using SHA-256. While the exact details of how SHA-256 works are classified, we know that it is built with a Merkle-Damgård structure derived from a one-way compression function itself created with the Davies-Meyer structure from a specialized block cipher.

Three properties make SHA-256 this secure. First, it is almost impossible to reconstruct the initial data from the hash value. A brute-force attack would need to make 2^{256} attempts to generate the initial data. Second, having two messages with the same hash value (called a collision) is extremely unlikely.

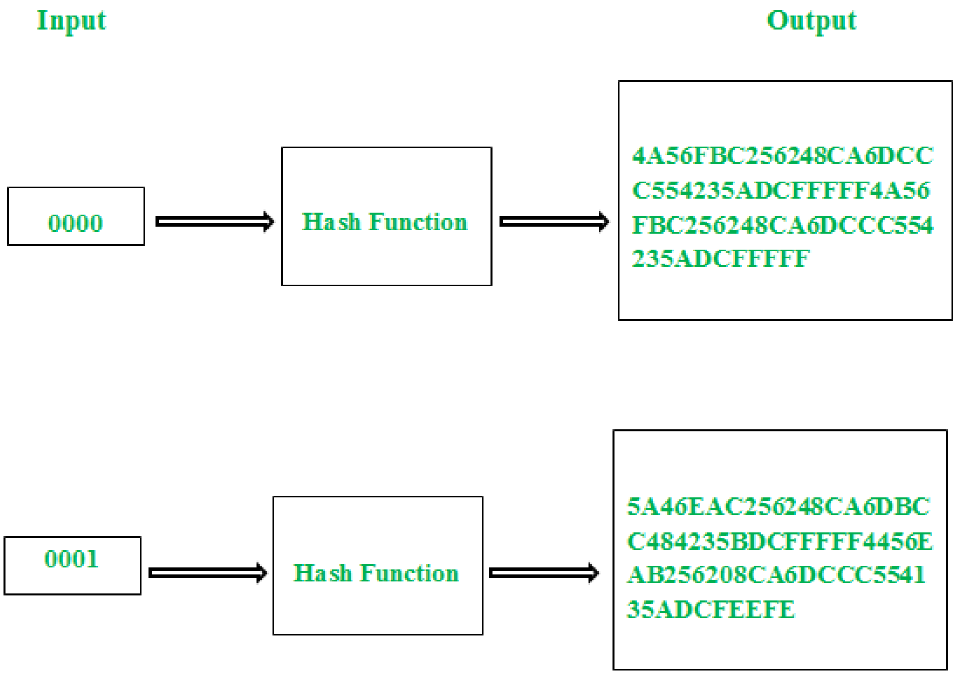
With 2^{256} possible hash values (more than the number of atoms in the known universe), the likelihood of two being the same is infinitesimally, unimaginably small. Finally, a minor change to the original data alters the hash value so much that it's not apparent the new hash value is derived from similar data; this is known as the *avalanche effect*.

Avalanche effect:

In cryptography, the avalanche effect is a term associated with a specific behavior of mathematical functions used for encryption. Avalanche effect is considered as one of the desirable property of any encryption algorithm. A slight change in either the key or the plain-text should result in a significant change in the cipher-text.

This property is termed as **avalanche effect**.

In simple words, it quantifies the effect on the cipher-text with respect to the small change made in plaint text or the key.



Even though the concept of avalanche effect was identified by “Shannon’s property of confusion”, the term was first mentioned by Horst Feistel. To implement a strong cipher or cryptographic hash function, this should be considered as one of the primary design objective.

In case of algorithm that uses hash value, even a small alteration in an input string should drastically change the hash value. In other words, flipping single bit in input string should at least flip half of the bits in the hash value.

A good encryption algorithm should always satisfy the following relation:

Avalanche effect > 50%

The effect ensures that an attacker cannot easily predict a plain-text through a statistical analysis. An encryption algorithm that doesn’t satisfies this property can favor an easy statistical analysis. That is, if the alteration in a single bit of the input results in change of only single bit of the desired output, then it’s easy to crack the encrypted text.

Checking the Validity of the Block:

- After mining several blocks the validity of the chain must be checked in order to prevent any kind of tampering with the blockchain.
- If the blockchain shows *invalid* as a output it means that tampering of blockchain has been done.
- If the blockchain shows *valid* as a output it means that no tampering of blockchain has been done.
- Then the web app will be made by using *Flask* and deployed locally.
- Now by *HTTP request* we call the different *GET* method and display the output.

Key Points:

HTTP Request:

The internet boasts a vast array of resources hosted on different servers. For you to access these resources, your browser needs to be able to send a request to the servers and display the resources for you. HTTP (Hypertext Transfer Protocol), is the underlying format that is used to structure request and responses for effective communication between a client and a server. The message that is sent by a client to a server is what is known as an HTTP request. When these requests are being sent, clients can use various methods.

Therefore, HTTP request methods are the assets that indicate the specific desired action to be performed on a given resource. Each method implements a distinct semantic, but there are some standard features shared by the various HTTP request methods.

What Are HTTP Request Methods?

An HTTP request is an action to be performed on a resource identified by a given Request-URL. Request methods are case-sensitive, and should always be noted in upper case. There are various HTTP request methods, but each one is assigned a specific purpose.

How Do HTTP Requests Work?

HTTP requests work as the intermediary transportation method between a client/application and a server. The client submits an HTTP request to the server, and after internalizing the message, the server sends back a response. The response contains status information about the request.

What Are the Various Types of HTTP Request Methods?

GET

GET is used to retrieve and request data from a specified resource in a server. GET is one of the most popular HTTP request techniques. In simple words, the GET method is used to retrieve whatever information is identified by the Request-URL.

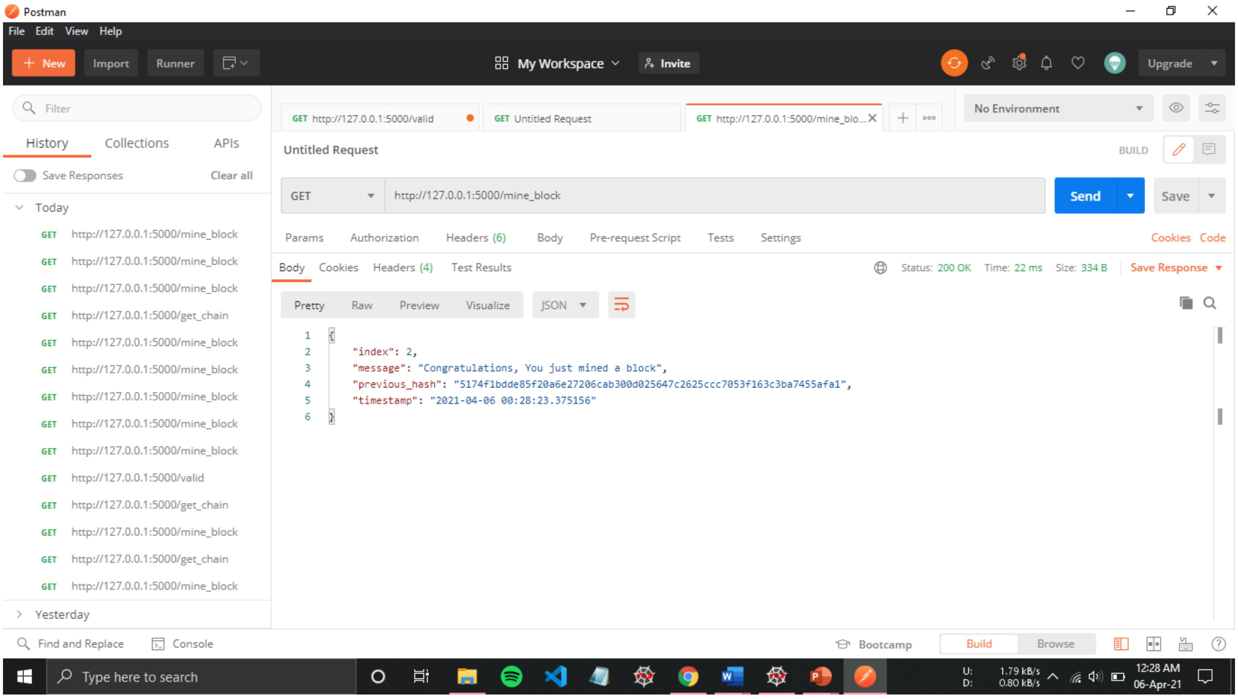
POST

Another popular HTTP request method is POST. In web communication, POST requests are utilized to send data to a server to create or update a resource. The information submitted to the server with POST request method is archived in the request body of the HTTP request. The HTTP POST method is often used to send user-generated data to a server. One example is when a user uploads a profile photo.

HEAD

The HEAD technique requests a reaction that is similar to that of GET request, but doesn’t have a message-body in the response. The HEAD request method is useful in recovering meta-data that is written according to the headers, without transferring the entire content. The technique is commonly used when testing hypertext links for accessibility, validity, and recent modification.

SNAPSHOT OF OUTPUT-1



Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
Filter
History
Collections
APIs
Save Responses
Clear all
Today
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/valid
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
Yesterday
GET http://127.0.0.1:5000/valid
GET Untitled Request
GET http://127.0.0.1:5000/mine_blo...
No Environment
Untitled Request
BUILD
Send
Save
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies Code
Body Cookies Headers (4) Test Results
Save Response
Pretty Raw Preview Visualize JSON
Find and Replace
Console
Bootcamp
Build
Browse
Type here to search
U: 1.79 kB/s
D: 0.80 kB/s
12:28 AM
06-Apr-21

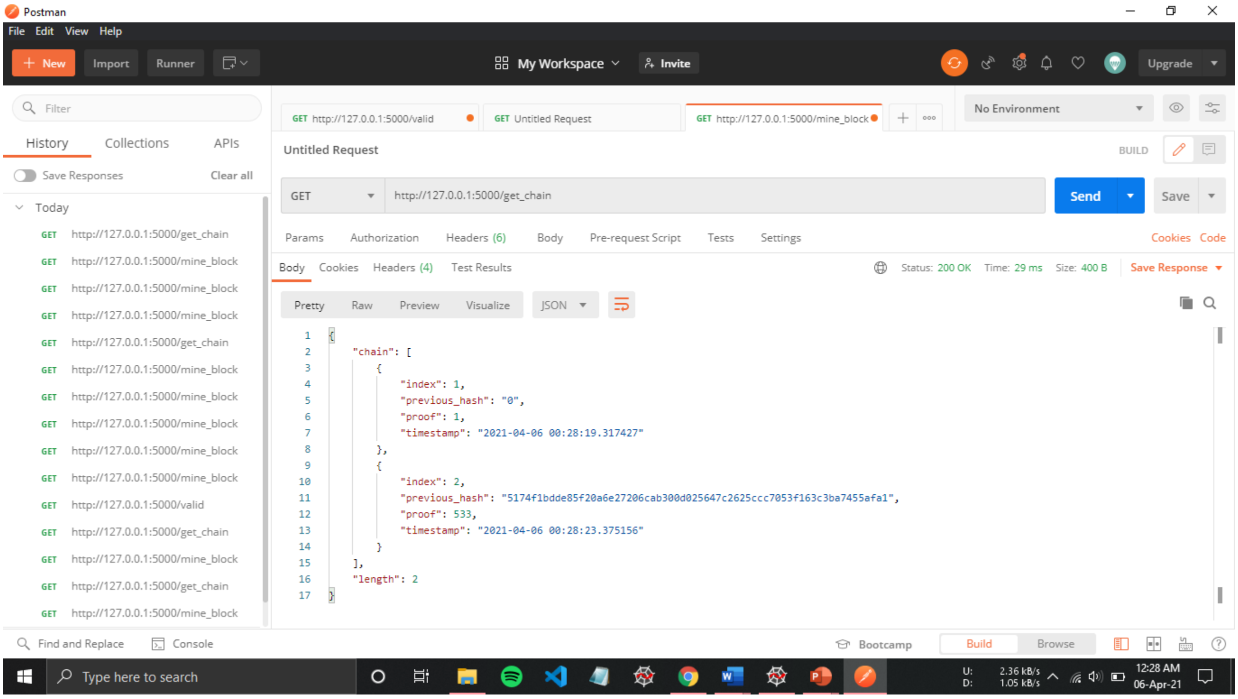
GET http://127.0.0.1:5000/mine_block

Status: 200 OK Time: 22 ms Size: 334 B

```
{
    "index": 2,
    "message": "Congratulations, You just mined a block",
    "previous_hash": "5174f1bdde85f20a6e27206cab300d025647c2625ccc7053f163c3ba7455afa1",
    "timestamp": "2021-04-06 00:28:23.375156"
}
```

In this image, we have mined the first block of the blockchain using **"mine_block"** HTTP request.

SNAPSHOT OF OUTPUT-2



Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
Filter
History
Collections
APIs
Save Responses
Clear all
Today
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/valid
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/valid
GET Untitled Request
GET http://127.0.0.1:5000/mine_block
No Environment
Untitled Request
BUILD
Send
Save
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies Code
Body Cookies Headers (4) Test Results
Save Response
Pretty Raw Preview Visualize JSON
Find and Replace
Console
Bootcamp
Build
Browse
Type here to search
U: 2.36 kB/s
D: 1.05 kB/s
12:28 AM
06-Apr-21

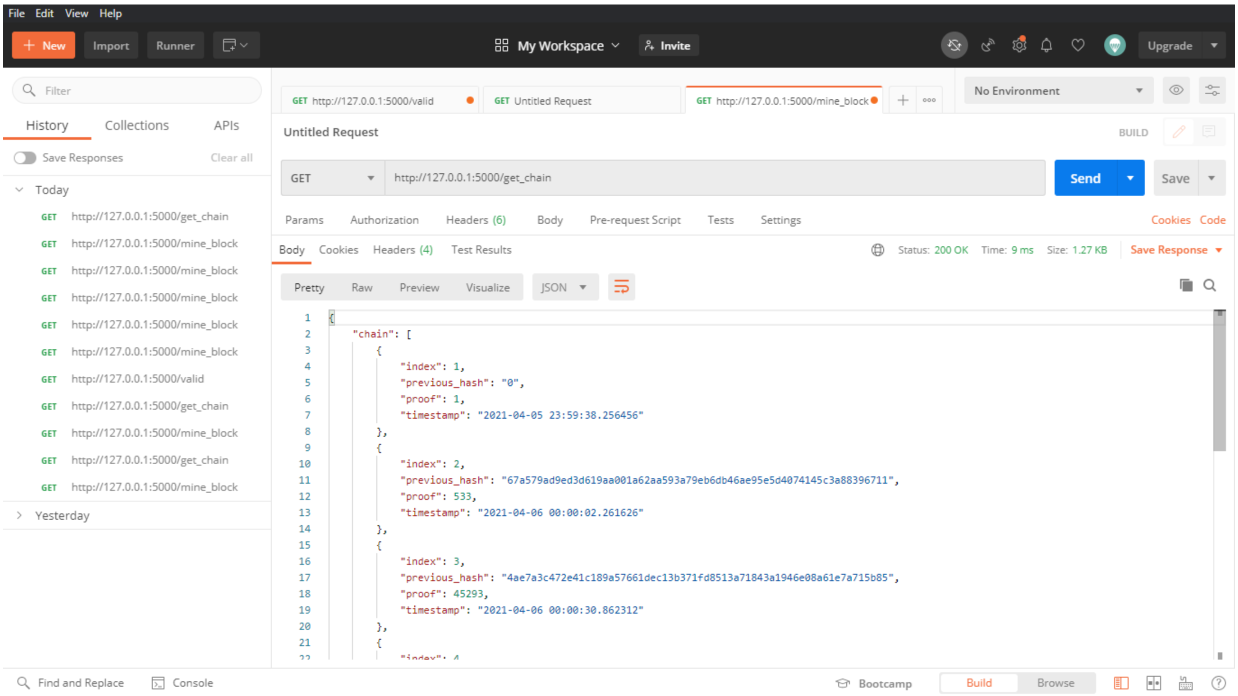
GET http://127.0.0.1:5000/get_chain

Status: 200 OK Time: 29 ms Size: 400 B

```
{
    "chain": [
        {
            "index": 1,
            "previous_hash": "0",
            "proof": 1,
            "timestamp": "2021-04-06 00:28:19.317427"
        },
        {
            "index": 2,
            "previous_hash": "5174f1bdde85f20a6e27206cab300d025647c2625ccc7053f163c3ba7455afa1",
            "proof": 533,
            "timestamp": "2021-04-06 00:28:23.375156"
        }
    ],
    "length": 2
}
```

In this image, we are displaying the Genesis Block and first block which he have mined previously using ***"get_chain"*** HTTP request.

SNAPSHOT OF OUTPUT-3



File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
No Environment
History
Collections
APIs
Save Responses
Clear all
Today
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/valid
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
Yesterday
GET http://127.0.0.1:5000/valid
GET Untitled Request
GET http://127.0.0.1:5000/mine_block
Untitled Request
BUILD
Send
Save
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Code
Body
Cookies
Headers (4)
Test Results
Save Response
Pretty
Raw
Preview
Visualize
JSON
Find and Replace
Console
Bootcamp
Build
Browse

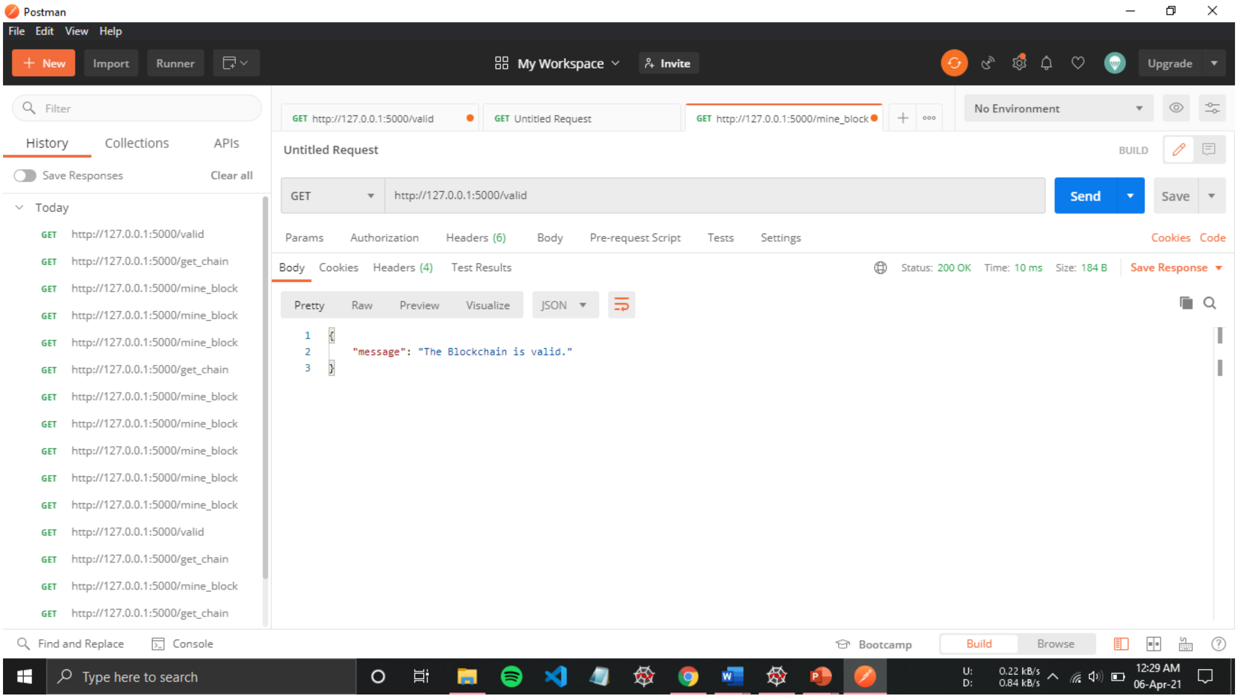
GET http://127.0.0.1:5000/get_chain

Status: 200 OK Time: 9 ms Size: 1.27 KB

```
{
    "chain": [
        {
            "index": 1,
            "previous_hash": "0",
            "proof": 1,
            "timestamp": "2021-04-05 23:59:38.256456"
        },
        {
            "index": 2,
            "previous_hash": "67a579ad9ed3d619aa001a62aa593a79eb6db46ae95e5d4074145c3a88396711",
            "proof": 533,
            "timestamp": "2021-04-06 00:00:02.261626"
        },
        {
            "index": 3,
            "previous_hash": "4ae7a3c472e41c189a57661dec13b371fd8513a71843a1946e08a61e7a715b85",
            "proof": 45293,
            "timestamp": "2021-04-06 00:00:30.862312"
        },
        {
            "index": 4
```

In this image, we are displaying the N number of Blocks in a blockchain using ***"get_chain"*** HTTP request.

SNAPSHOT OF OUTPUT-4



Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
No Environment
History
Collections
APIs
Save Responses
Clear all
Today
GET http://127.0.0.1:5000/valid
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/valid
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/mine_block
GET http://127.0.0.1:5000/get_chain
GET http://127.0.0.1:5000/valid
GET Untitled Request
GET http://127.0.0.1:5000/mine_block
Untitled Request
BUILD
Send
Save
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Code
Body
Cookies
Headers (4)
Test Results
Save Response
Pretty
Raw
Preview
Visualize
JSON
Find and Replace
Console
Bootcamp
Build
Browse
Type here to search
U: 0.22 kB/s
D: 0.84 kB/s
12:29 AM
06-Apr-21

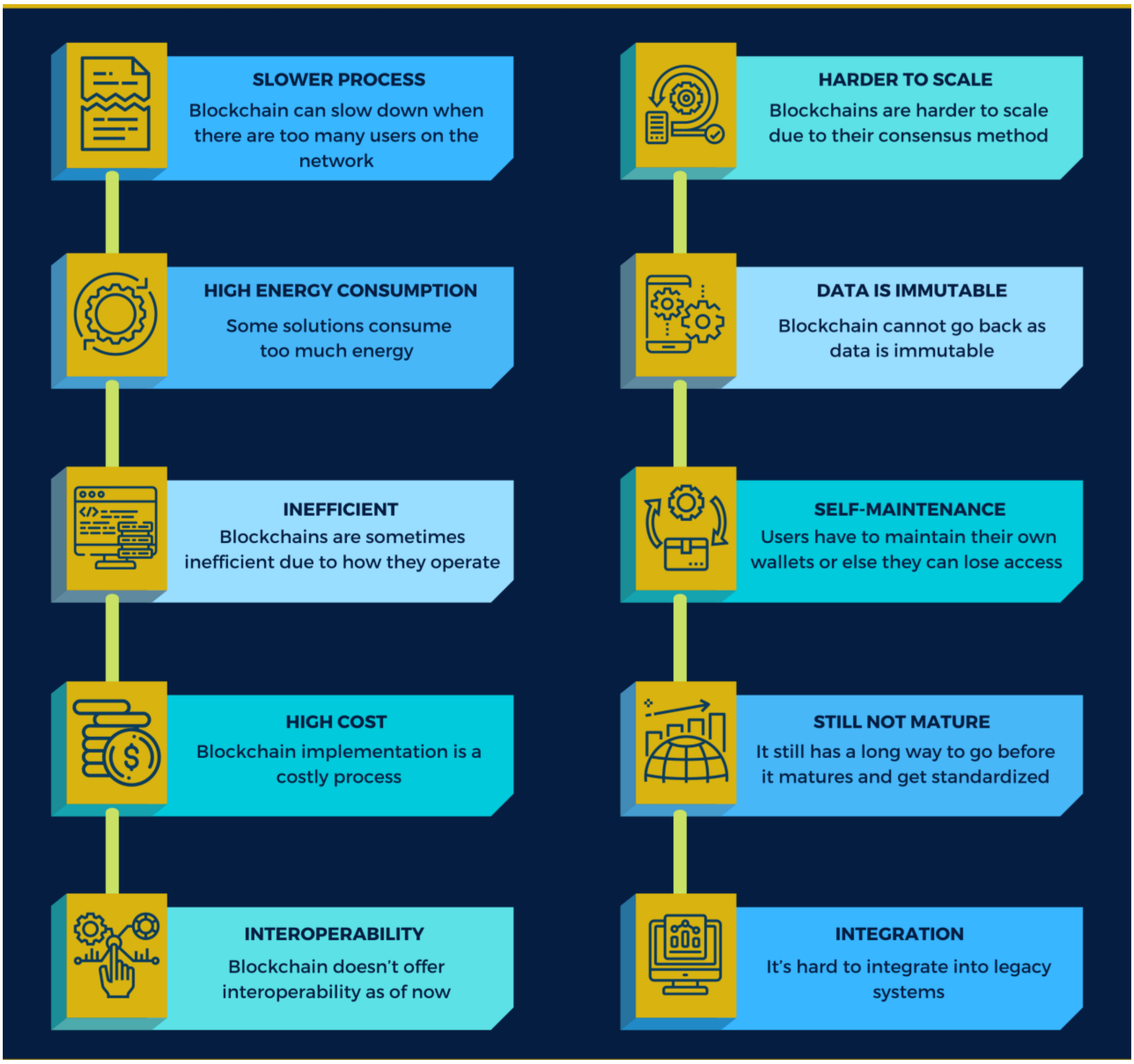
GET http://127.0.0.1:5000/valid

Status: 200 OK Time: 10 ms Size: 184 B

```
{
    "message": "The Blockchain is valid."
}
```

In this image, we are verifying the blockchain is valid or not as it get linked up using Hashing algorithm using ***"valid"*** HTTP request.

LIMITATIONS


SLOWER PROCESS
Blockchain can slow down when there are too many users on the network
HARDER TO SCALE
Blockchains are harder to scale due to their consensus method
HIGH ENERGY CONSUMPTION
Some solutions consume too much energy
DATA IS IMMUTABLE
Blockchain cannot go back as data is immutable
INEFFICIENT
Blockchains are sometimes inefficient due to how they operate
SELF-MAINTENANCE
Users have to maintain their own wallets or else they can lose access
HIGH COST
Blockchain implementation is a costly process
STILL NOT MATURE
It still has a long way to go before it matures and get standardized
INTEROPERABILITY
Blockchain doesn't offer interoperability as of now
INTEGRATION
It's hard to integrate into legacy systems

1. Blockchain is not a Distributed Computing System

Blockchain is a network that relies on nodes to function properly. The quality of the nodes determines the quality of the blockchain. For example, Bitcoin's blockchain is strong and incentivizes the nodes to participate in the network. However, the same cannot be true for a blockchain network that does not incentivize the nodes.

This means that it is not a distributed computing system where the network doesn't depend on the involvement and participation of the nodes. In comparison, a distributed computing system works to ensure that they verify the transactions according to the rules, ensure that they record the transactions, and also make sure that they have the transactional history for each transaction. Each of these actions is similar to that of blockchain, but there is a lack of synergy, mutual assistance, and paralleling for each one of them.

Clearly, blockchain might be a distributed network, but it lacks the features that make a distributed computing system so beneficial for the corporations.

2. Scalability Is An Issue

Blockchains are not scalable as their counterpart centralized system. If you have used the Bitcoin network, then you would know that the transactions are completed depending on the network congestion. This problem is related to scalability issues with blockchain networks. In simple words, the more people or nodes join the network, the chances of slowing down is more!

However, there has been an increasing change in how blockchain technology works. With the right evolution of the technology, scalability options are being integrated with the Bitcoin network as well. The solution is to do transactions off-blockchain and only use blockchain to store and access information.

Other than that, there are also new ways of solving scalability, including permissioned networks or using a different architectural blockchain solution such as Corda.

However, all these solutions are still not at par with the centralized systems. If you compare Bitcoin and VISA transaction speed, you will find a huge difference between them. Right now, Bitcoin can only do 4.6 transactions per second. In comparison, VISA can do a whooping 1700 transactions per second. This means that in a day, it can do 150 million transactions per second.

Lastly, we can say that blockchain might not be still well-equipped for real-world applications. It still needs significant improvement before it can be adopted in day-to-day life.

3. Some Blockchain Solutions Consume Too Much Energy

Blockchain technology got introduced with Bitcoin. It uses the Proof-of-Work consensus algorithm that relied on the miners to do the hard work. The miners are incentivized to solve complex mathematical problems. The high energy consumption is what makes these complex mathematical problems not so ideal for the real-world.

Every time the ledger is updated with a new transaction, the miners need to solve the problems which means spending a lot of energy. However, not all blockchain solutions work in the same manner. There are other consensus algorithms that have solved the problem. For example, permissioned or private networks do not have these problems as the number of nodes within the network is limited. Also, as there is no need for global consensus, they use efficient consensus methods to reach consensus.

But, if you take the most popular blockchain network, Bitcoin, the problem still persists that needs to be solved.

In short, permissioned networks are efficient when it comes to energy consumption whereas public networks can consume a lot of energy to remain operational.

4. Blockchain Cannot Go Back — Data is Immutable

Data immutability has always been one of the biggest disadvantages of the blockchain. It is clear that multiple systems benefit from it including supply chain, financial systems, and so on. However, if you take how networks work, you should understand that this immutability can only be present if the network nodes are distributed fairly.

What I mean to say is that a blockchain network can be controlled by an entity if he owns 50% or more of the nodes — making it vulnerable.

Another problem that it suffers from is the data once written cannot be removed. Every person on the earth has the right to privacy. However, if the same person utilizes a digital platform that runs on blockchain technology, then he will be unable to remove its trace from the system when he doesn't want it there. In simple words, there is no way, he can remove his trace, leaving privacy rights into pieces.

5. Blockchains are Sometimes Inefficient

Right now, there are multiple blockchain technologies out there. If you pick up the most popular ones including the blockchain technology used by Bitcoin, you will find a lot of inefficiencies within the system. This is one of the big disadvantages of blockchain.

First of all, when I tried to set up the bitcoin miner on my system, I quickly found out that the ledger can easily cross 100's of GBs. It was not efficient in data storage which can lead to storage problems for multiple nodes who want to become part of the network.

Clearly, there needs to be a better way to handle this as whenever the data is updated, nodes need to replicate it. Moreover, the size of the blockchain grows with more transactions and nodes. If it continues to grow, then the whole network is slowed down. This is not ideal for commercial blockchains where it is essential for the network to be fast and secure at the same time.

Slowly inefficiencies are being improved with the help of other blockchain solutions. Bitcoin is also trying to solve inefficiencies with the help of lightning networks.

6. Not Completely Secure

Blockchain technology is more secure than other platforms. However, this doesn't mean that it is not completely secure. There are different ways the blockchain network can be compromised. Let's go through them below one by one to make more sense out of it.

- 51% attack: In the 51% attack, if an entity can control 51% or more of the network nodes, then it can result in control of the network. By doing so, they can modify the data in the ledger and also do double-spending. This is possible on networks where the control of miners or nodes are possible. This means that private networks are more likely to be safe from 51% attacks, whereas public ones are more vulnerable to this.
- Double-spending: Double-spending is yet another problem with the current blockchain technology. To prevent double-spending the blockchain network deploys different consensus algorithms including Proof-of-Stake, Proof-of-Work, and so on. Double spending is only possible on networks with a vulnerability to the 51% attack.
- DDoS's attack: In a DDoS attack, the nodes are bombarded with similar requests, congesting the network and bringing it down.
- Cryptographic cracking: Another way the blockchain technology is not secure is that the cryptographic solution that it utilizes. Quantum algorithms or computing are more than capable of breaking cryptographic cracking. However, blockchain solutions are now implementing quantum-proof cryptographic algorithms.

7. Users Are Their Own Bank: Private Keys

To make blockchain decentralized, it is important to give individuals the ability to act as their own bank. However, this also leads to another problem.

To access the assets or the information stored by the user in the blockchain, they need private keys. It is generated during the wallet creation process, and it is the responsibility of the user to take proper note of it. They also need to make sure that they do not share it with anyone else. If they fail to do so, their wallet is in danger. Also, if they lose the private key, they will lose access to the wallet forever. The reliance on users makes it as one of the disadvantages of blockchain.

So, if you as a user who forgets its private key, are eventually logged out of their wallet and no one can get it back. This is a serious drawback as not all users are tech-savvy and have more chances to make mistakes. If there is a centralized authority that takes care of it, then it defeats the purpose of decentralization.

8. Cost And Implementation Struggle

The underlying cost of implementing blockchain technology is huge. Even though most of the blockchain solutions including Hyperledger are open source, they require a lot of investment from the organization that is willing to pursue it.
There are costs associated with hiring developers, managing a team that excels at different aspects of blockchain technology, licensing costs if you opt for a paid blockchain solution, and so on.

You also need to take care of the maintenance cost associated with the solution. For enterprise blockchain projects, the cost can go over a million dollars as well.

So for businesses who like the idea of blockchain, but do not have the funds or budget to carry out, might need to wait more before they can jump into the blockchain bandwagon.

9. Expertise Knowledge

Implementing and managing a blockchain project is hard. It requires thorough knowledge from the business to go through the whole process.

They need to hire multiple experts in the blockchain field that leads to the problem and hence it is counted as one of the disadvantages of blockchain.

Not only that they also need to train their existing professionals on how to utilize blockchain and then ensure that the management team can understand the complexities and outcomes of a blockchain-powered business.

This way, they can understand their requirements and help transform their business processes to utilize blockchain.

Not to mention, if you find blockchain developers and specialists, they are harder to find and will cost more compared to traditional developers due to their demand and supply ratio.

If you are eager to learn about Blockchain use-cases then you can check out the articles listed below.

- 12+ Practical Blockchain Use-Cases 2020
- 10+ Must Know Enterprise Blockchain Use Cases

10. Maturity

Blockchain technology is only a decade old. This means that it is a new technology that requires time to mature. If you take the different consortium into account, you will notice multiple players trying to solve the decentralized problem with their unique solution.
For example, we have Corda, Hyperledger, Enterprise Ethereum, Ripple, and so on! All-in-all, there is still a lot of time left before the blockchain technology matures and businesses will have less hesitation to adopt blockchain technology.

Like any other new technology, maturity is another problem that blockchain has to solve, and hence it is one of the disadvantages of blockchain.

Blockchains are also not getting matured in a long time for now. There is still a lot to go before we can see changes in standardizing blockchain technology. Right now, there are too diverse solutions that aim to solve the core problems, but are not working together to standardize it.

11. Interoperability

Another disadvantage that blockchain technology suffers from is interoperability. As mentioned in the last point, there are multiple types of blockchain networks which work differently, trying to solve the DLT problem in their own unique way. This leads to interoperability issues where these chains are not able to communicate effectively.

The interoperability issue also persists when it comes to traditional systems and systems using blockchain technology.

12. Legacy Systems

Not all businesses have changed from legacy systems. There are still many organizations that rely on legacy systems to run their business. However, if they want to adopt blockchain technology, they need to completely get rid of their systems and change to blockchain technology — which is not feasible for every business out there.

CONCLUSION

Many enthusiasts believe that the future lies with the blockchain. Objectively, we will see the real impact across the industries with time. On the other hand, the chances that the domains of finance, insurance, and healthcare will utilize blockchain technology are pretty high.

The Bitcoin is the first successful implementation of blockchain. Today, the world has found applications of blockchain technology in several industries, where the trust without the involvement of a centralized authority is desired. So welcome to the world of Blockchain.

References/Bibliography

a. https://www.superdatascience.com/

b. https://www.javatpoint.com/

c. https://www.w3schools.com/

d. https://www.geeksforgeeks.org/

e. https://www.tutorialspoint.com/index.htm

f. https://101blockchains.com/disadvantages-of-blockchain/

g. https://rapidapi.com/

h. https://www.movable-type.co.uk/scripts/sha256.html

i. https://qvault.io/cryptography/how-sha-2-works-step-by-step-sha-256/

j. https://mlsdev.com/blog/156-how-to-build-your-own-blockchain-architecture

k. https://medium.com/mobindustry/designing-a-blockchain-architecture-types-use-cases-and-challenges-9894fb7b58e

l. https://mlsdev.medium.com/blockchain-architecture-basics-components-structure-benefits-creation-beace17c8e77

m. https://www.guru99.com/blockchain-tutorial.html

n. https://www.edureka.co/blog/blockchain-architecture/

o. https://www.upgrad.com/blog/blockchain-architecture/

p. https://www.upgrad.com/blog/what-is-blockchain-how-to-create-networkcode-its-architecture/