BLOCKCHAIN EXPERT

BLOCKCHAIN

E-BOOK

CYBROSYS™ Technologies

Cybrosys Limited Edition

BLOCKCHAIN

E-BOOK

Table of Content

Section I

Section II

About The Publisher

Cybrosys is a well-established ISO Certified software development company which provides quality services all over the world. ERP solutions being our core area of work, we are also into Source code sale, Custom software development, and Employee outsourcing. We serve our widespread customers around the globe via our offices located in London, Bangalore, Kochi, and Calicut. Our partnership with technology leaders like Microsoft, Sun, IBM, Symantec, and Odoo assist us to thrive to the best of industry standards. We always kept abreast with changing technologies to deliver the best to our customers. And now our latest research and developments are being conducted in Blockchain. Our dedicated Blockchain R&D lab is making promising progress in blockchain backed solutions like DApp and Tokens. Our mission is ‘Develop most reliable cost effective software based on innovation and creativity. We believe in consistency, competence, flexibility, commitment and most of all we value our existing customers and continued customer satisfaction’.

CEO’s Message

“Be updated, before you get outdated”

It’s been a long journey since we established Cybrosys. We have seen tides of growth and decline during the voyage. But we endured everything, and here we are, as one of the fast-growing players, constantly striving to be better. It is always been the crew, their perseverance, and efforts, that lead us forward. And we never gave up on our mission, in fact, our mission steered our drive. We always ventured beyond the boundaries and explored the latest, our arsenal always filled with the advanced, and Blockchain is the latest in the list.

A book like this is indeed a need of the hour, it is an aspiration of our research team; comprehending all the basics of blockchain in a single space to help those who wish to start with blockchain. Despite their tight R&D schedules, they have made incredible efforts to accomplish this work. In this occasion, I would like to express my sincere gratitude to all the team members who worked behind this work. And I wish the work be an excellent guide to all.

Sainul Abideen

CEO, Cybrosys Technologies

Preface

Many people use the term ‘Blockchain Technology’ to mention different things. Sometimes they may be talking about Bitcoin, sometimes it’s about the cryptocurrencies or digital tokens, sometimes it is about Ethereum Blockchain, or it may be about the smart contracts. The fact is that often the term ‘Blockchain’ is used inappropriately by many people, consequently, not only the word ‘Blockchain Technology’ but also the whole terms and terminologies related to it became confusing for many. However, all these usages have a common thread; which is the distributed ledger technology underlying it. In distributed ledger technology, the transactions are copied and stored across individual computers on the network rather than storing on a central server.

Even though the Blockchain technology has grown faster than expected; the exact information regarding it hardly touched the common developers and techno enthusiasts. Many blogs and dedicated websites are also coming to explain the entire technology and surrounding developments. But it seems like that, lack of a comprehensive guide which acquaints, compare and contrast the overlapped terms and terminologies related wto blockchain is still missing. It is in this context the Cybrosys technologies has decided to come up with a comprehensive guide which covers all the basics of Blockchain Technology as well as popular technologies and terminologies related to it.

It is a basic guide for anyone who wishes to start with blockchain technology. We tried our best to organize the topics in a way that both developers and techno enthusiast can go through it and understand the topics without any effort. But please remember that this guide does not include everything you may possibly come across while dealing with blockchain; rather it is a starter pack. Use this as a starting point to further explore and expand your knowledge base in the Blockchain technology.

Introduction

Blockchain- Despite the inherent ‘Block’ in it, the name has traversed more miles than any other technical term in the recent past. It is echoing in almost all existing IT infrastructures; posing a potential threat to the very existence of the present establishments. The blockchain is said to be the technology of future. Here we are trying to simplify the things for all those who wish to understand the technology. As we indicated in the preface, the book is meant for anyone who wishes to start with blockchain technology.

We have organized the book into two major sections, while the first section provides the basics of Blockchain and related terminologies, the second section is purely dedicated to different tools and technologies that emerged along with blockchain.

First section is further divided into six major topics. Blockchain, Cryptocurrency, Bitcoin, Ethereum, Hyperledger and Tokens which covers all the basic ingredients for starting with the blockchain technology. In the first part, which is about blockchain, we have discussed what blockchain, its working principles, the historical developments, the technical implementations, its application areas and the possible future of Blockchain. The second topic is about Cryptocurrencies, which is an essential topic that must be learned before going deep into the famous blockchain protocol Bitcoin. A general overview of cryptocurrencies as well as their working principles is discussed here. The third topic is about the most popular blockchain platform- Bitcoin. Here we discussed the topics like bitcoin and its background, bitcoin working, bitcoin mining, the value of bitcoin etc. In the next topic which is about Ethereum, we have included the details of another popular blockchain platform – Ethereum. The reader will get an overview of the second most popular blockchain platform from this section.
Ethereum related terms and terminologies like the smart contract, Solidity, DApp, Etherscripter, Ether etc. are also simplified here. The fifth topic is about the

ambitious open source project Hyperledger. The project, its objectives, and the products that have developed under the project etc. are discussed in this section. We have alsoincluded a comparative study of all of these technologies to give a better understanding. The final topic is exclusively dedicated to -Tokens, which is a thriving application area of blockchain technology.

Second section of this book doesn't need much introductory comments, all the topics in the second section is more or less independent. The section provides information about different blockchain related tools like wallets, Programming languages and IDEs, Blockchain platforms and development frameworks.

To simplify the things further we have tried to include images, infographics, tips and quick info bars wherever possible. Moreover, most of the terms and terminologies we used are explained in the beginning of the book. Make use of all these extra information provided while going through the book and have a good read.

Terms and Terminologies

Some of the terms and terminologies you may encounter while going through this book is described here.

Use it as a quick reference.

Block

- Block is used to store the transaction along with their hash value and data

Transaction

- Any state change occurred in a blockchain

Smart contract

- self executing contract with terms and conditions written in lines of codes

Ledger

- Blockchain ledger is used to record the transactions in a blockchain

Token

- Digital asset

Cryptocurrency

- Digital asset

Bitcoin

- Most popular cryptocurrency

Hash

- The encrypted value of the data in the block.

SHA256

- Hashing Algorithm

Node

- Each computer connected to the blockchain network

Solidity

- Programming language for writing smart contracts in Ethereum

Hyperledger

- Blockchain platform

Ethereum

- blockchain platform

Baas

- Blockchain as a service

ERC20

- Ethereum token standard

ICO

- Initial coin offering

DApp

- Decentralized applications

IoT

- Internet of things

PoW

- Proof of work

PoS

- Proof of stake

Mining

- The validation process in a blockchain (in Bitcoin and Ethereum)

Miner

- The nodes which perform mining

Wallets

- Digital wallet to store, send and receive cryptocurrencies and other digital assets.

Testnet

- Test blockchain networks for development and testing purpose

BFT

- Byzantine fault tolerance principle.

BIP

- Bitcoin improvement proposal

Genesis block

- First block in the blockchain

Composer

- blockchain development framework in hyperledger fabric

Participants

- Those who have an account in the blockchain and performing any transactions.

Peer2Peer(P2P)

- Decentralized network architecture. There is no dedicated server in this case

Consensus

- General agreement between the participants in the blockchain

An Introduction to Blockchain

The Beginning

Before going into the details of working principles and other aspects of blockchain; let’s look into the genesis of the technology itself. The conceptual framework behind blockchain was first put forward by a group of researchers in 1991. The idea was initially intended for time-stamping digital documents such that backdating them will not be possible thereafter. However, the idea went mostly unused until it was again mentioned by Satoshi Nakamoto in his white paper “Bitcoin: A Peer-to-Peer Electronic Cash System”.

It may be the first time in history that the inventor of a game-changing technology has completely gone anonymous. Satoshi Nakamoto; an anonymous person/group is said to be behind the first blockchain, which is Bitcoin. Bitcoin is the first blockchain came into existence and it was in 2009. In the following years, the bitcoin became popular, and the underlying technology became even more popular. **So the confusion and lack of clarity among people start from the origin itself; a product and its related terminologies went viral before the technology behind it. And when the blockchain displayed its real potential, people were trying to relate it with the bitcoin terminologies; the result was total misconception and confusion.** But it is the other way; start from blockchain and then try to understand bitcoin.

Why Blockchain

It is another question that must be addressed first before going into the details of the technology. To say technology is revolutionary; obviously, it must have a lot of advantage over existing technologies. Here are some advantages of blockchain over existing systems of different domains. Blockchain is:

- **Decentralized**
- **Distributed**
- **Secure and Faster**
- **Transparent and Immutable**

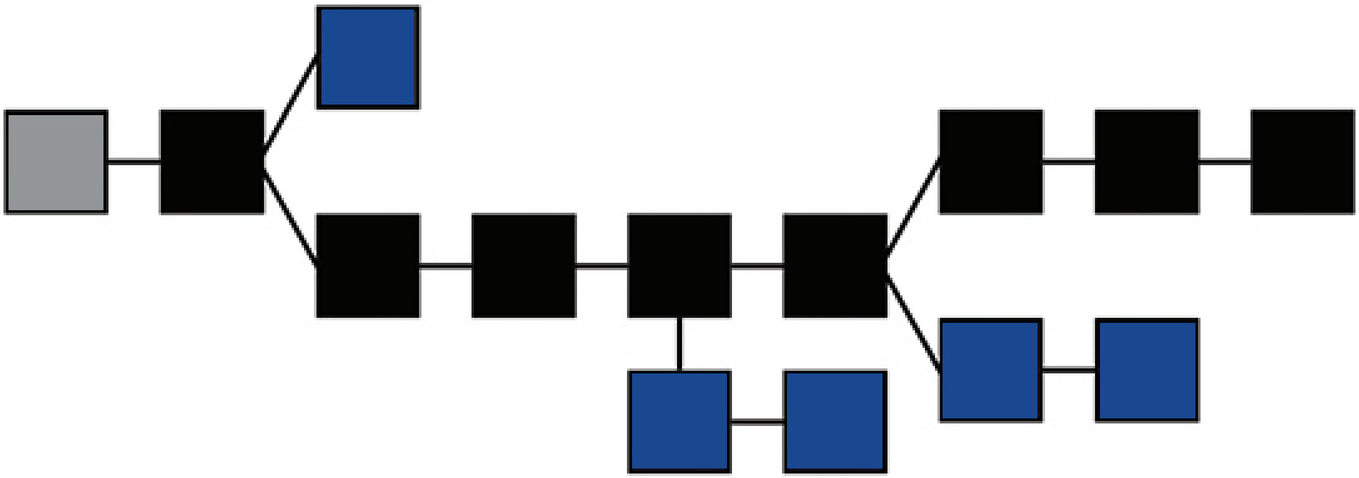
The features can be understood well if we look the data structure, data distribution, data validation (Authentication of a piece of data in blockchain) and other related terminologies of blockchain.

The Structure of Blockchain

According to IBM, blockchain is a shared, distributed ledger that facilitates the process of recording transactions and tracking assets in a network. The asset may be a tangible asset like property, house, vehicle or an intangible asset like digital currency, intellectual property rights, etc. Basically, it stores Data, and records its movements in a distributed environment. Let's look into its details.

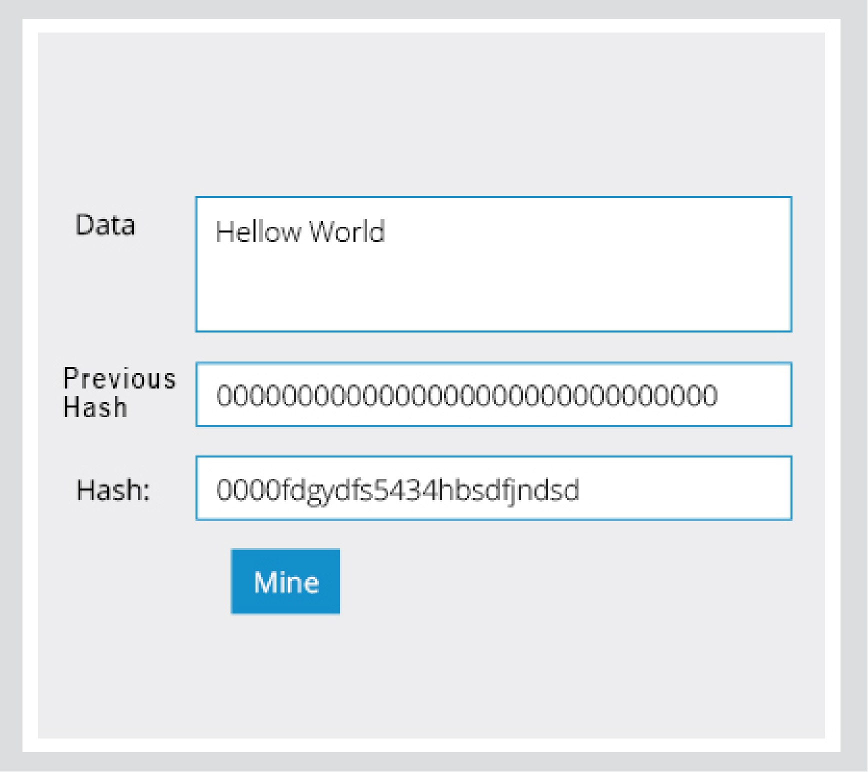
It is a distributed database or a public registry that keeps details of assets and its movements/transactions across a P2P network. Each transaction will be secured through cryptography and later all the transaction history will be grouped and stored as blocks of data. Then the blocks are linked together with cryptography and secured from modification. The whole process will create an unforgeable, and immutable record of the transactions that happened across the network. Additionally, this blocks of records are copied to every participating computer in the network, so everyone will have access to it. The great advantage of blockchain is that it can store any kind of asset, its ownership details, history of the ownership and location of assets in the network. Whether it is the digital currency bitcoin, or any other digital assets like a certificate, personal information, a contract, title of ownership of IP, even the real-world objects.

The powerful feature of Blockchain is that we can create a shared reality across non-trusting entities. That is all of these participating nodes in the network do not need to know each other or trust each other because each has the ability to monitor and validate chain for themselves. The irony is that the mutual distrust among participant is the thing which keeps the blockchain secure and verified.



Data Structure of Blockchain

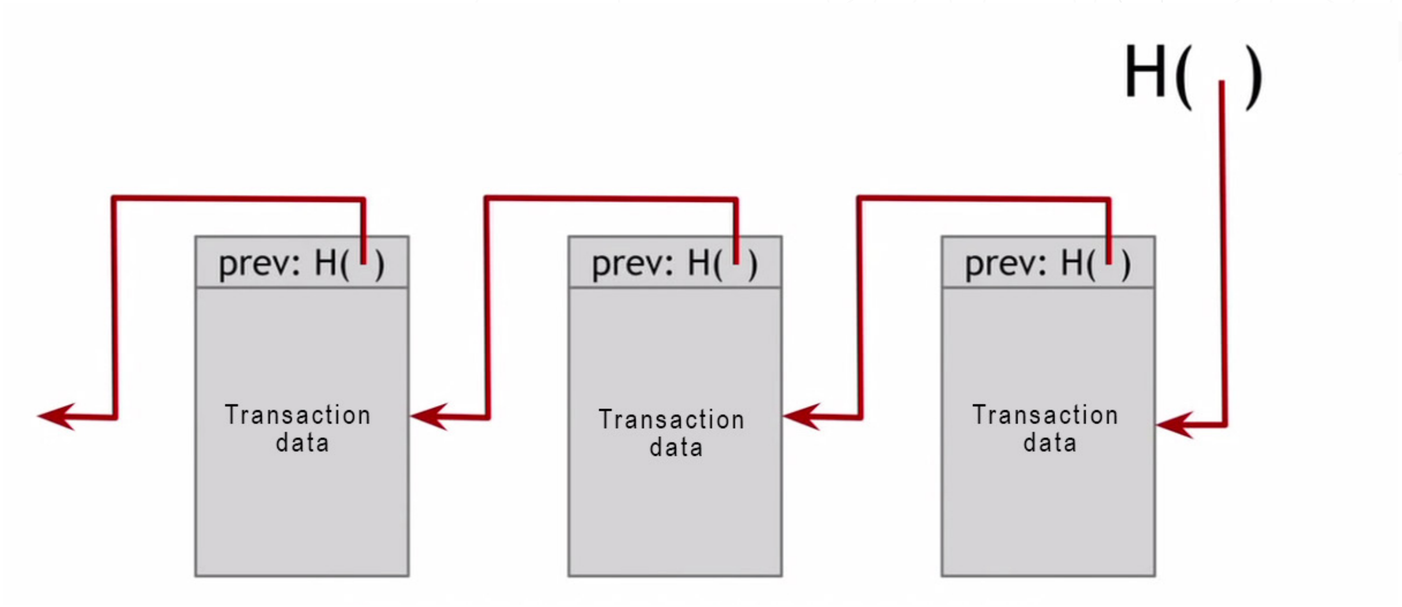
The data in blockchain is stored as individual blocks, that's why it is called Blockchain. Just like a linked list, the Blockchain is a collection of blocks linked together. So what does the block actually contain? Each block in a blockchain will have the following fields.



Data	Hellow World
Previous Hash	0000000000000000000000000000000
Hash:	0000fdgydfs5434hbsdfjndsd

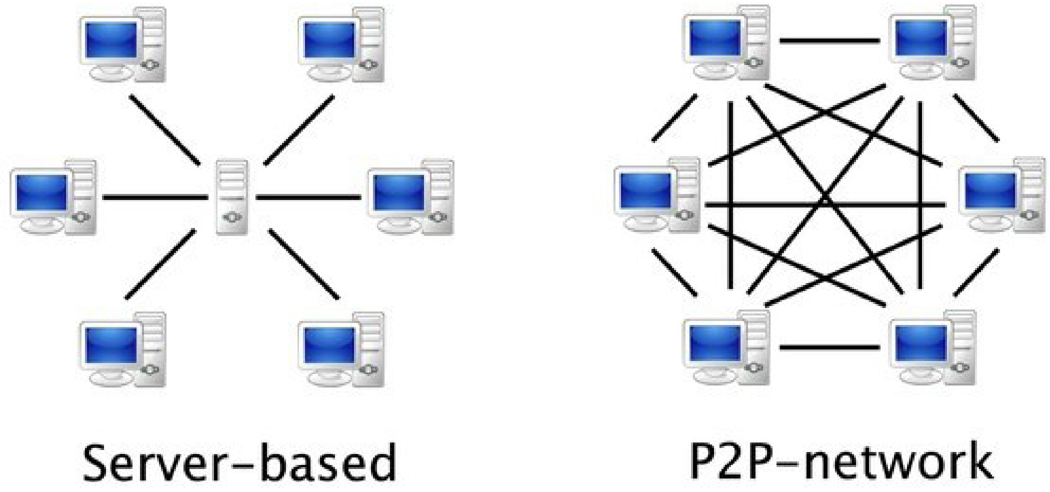
1) Data: Stores the data
2) Previous hash: Stores the hash of the previous block
3) Hash: Hash value for the current block which can be used to refer this block

As far as the user is concerned the Data field is the most important thing. The actual data (like transaction details, asset details etc.) are stored in this field. Previous hash will store the hash values of the previous block (consider it as a link to the previous block), the blocks are connected through this value.



Data Distribution in Blockchain

We saw that blockchain has its own unique Data storage structure, the data distribution in a blockchain has also a different approach. **They don't follow the widely adopted client server model rather the Peer to Peer model. The peer to peer data distribution approach gives the reason behind unfettered nature of Blockchain; there is no central authority to control.**



Unlike the client-server model, In P2P network the data is stored in all the participant nodes in the network. All the individual nodes will have the copy of the entire 'Blocks' and a single change in a particular block will be updated in all the nodes.

But here is the problem, in Client-Server model the data is stored in DB after verification of a central authority; but in P2P network there is no central authority, then how does the authenticity of data assured? The answer is the validation process and consensus mechanism of the blockchain network

Block Validation

As we described above; the asset and its transactions are stored as connected blocks in blockchain. Only the valid transactions are added to the blockchain. Technically saying, Blockchain validation is simply the process of finding the block hash. In a blockchain, all the blocks are added to the blockchain after validation only. Whenever a transaction takes place in the blockchain it will be added to a block; sometimes one transaction per block and sometimes several transactions per block. It depends on the block size and the nature of the network. When a transaction is added to the block, it must undergo a validation process before it is being added to the blockchain as a valid block. The hash value for the block can be calculated using some algorithms (like sha 256). The hash value has certain properties too. The main thing is that the hash value should be collision-free i.e. no two blocks should have the same hash value. Since each block is represented using the hash value it should be identical. The second property is that the hash values should be irreversible. This means the block data could not be retrievable from the hash value

Block Validators

Block validators are the nodes which participates in the process of block validation. The validators are rewarded for their effort, (In fact they are rewarded for the computational power they spent). Different blockchain protocols adopt different methodologies for selecting the validator from available pool of nodes. Some of the methods are described below.

PoW (Proof of Work)

In PoW, the mining challenge is open to all. All the miners compete each other to add the next block. A fixed reward is given to the miner who finds the solution first. In fact, the node with more computational power usually wins the race. Bitcoin uses the PoW algorithm.

PoS (Proof of Stake)

It is a common alternative of PoW. Here, the validators are chosen based on the fraction of coins they own in the system. The nodes with more number of coins have more chance to be selected than the node with lesser number of coins. In PoS the reward is in the form of transaction fee, new coins are not created for paying the validators. Presently, Blackcoin, NXT and Peercoin blockchains uses the PoS algorithm. Ethereum is also planning to shift to this method by 2018.

Proof of Activity

PoA is a hybrid approach and it is introduced to overcome some of the problems in PoS and PoW. In this method, the mining begins with PoW and at some point the process is switched PoS. Presently, ‘Decred’ is the only coin that is using a variation of proof of activity.

Proof of Elapsed Time

In this method, the network uses a lottery functions for implementing consensus. A lottery algorithm is used for finding the leaders from a set of nodes. So the validators are selected randomly from the pool. Hyperledger Sawtooth blockchain uses PoET method. .

Proof of Burn

In this method, the aspiring validators increase their stake in the system by sending their coins to an irretrievable location (thus the name burn). The validators are selected randomly, but those who has more stake in the system has high probability to get selected. Over the time the earned stake decays and the nodes has to burn more currency to increase their stake. The only coin that uses proof of burn mechanism is slimcoin.

At this stage we can't say which method is more efficient. Each method has its own advantages and disadvantages. Many other methods are also being introduced to attain maximum productivity on a blockchain.

Blockchain So far

Initially, it was about Bitcoin; following the trend, many other cryptocurrencies also came into the market. While some of them found their fortune, some other cryptocurrencies lagged behind. However, soon the blockchain technology found its real potential and spread to many other unpredicted domains. Healthcare Industry, Enterprise software development, financial domains like Banking, Insurance and so on; today the blockchain is drastically changing existing technology frameworks of almost all domains. According to prominent statistics websites, the blockchain market is expected to grow $20 billion by 2024.

Banking and payments

All the banking and payment systems are now moving towards blockchain. Bitcoin-like cryptocurrencies can control the payment systems without any geopolitical restrictions. ABRA is an example of bitcoin-based remittance.

Cyber Security

In blockchain, data is verified and secured using cryptography. This will restrict all unauthorized changes and hacks in the system. It removes the middlemen from the system so no one can make any unauthorized changes.

Supply chain

The blockchain can revolutionize the supply chain by providing better transparency, accountability and feedback mechanism along the supply chain. Any product can be tracked completely using the blockchain supply chain management. Each and every movement, as well as the condition of a product, can be recorded in the blockchain with IoT sensors. Blockverify and Provenance is a blockchain based supply chain management system.

Online Data Storage

Data on the centralized server like Onedrive, Google Drive etc. are vulnerable to the single point of failure. Blockchain allows distributed data storage in a more secure and robust way. Storj is such an encrypted cloud storage facility

Networking and IoT

The blockchain technology can be applied in Networking and IoT to create a decentralized network of IoT devices. This eliminates the need for a central location to handle the IoT devices.

Insurance

The global insurance market is based on trust management. Blockchain is the new way of managing the trust. Blockchain ensures trust by mutual distrust between participants. 'Aeternity' is an example of blockchain based insurance management system.

Government

Applying blockchain technology in government systems will reduce bureaucratic hurdles, red-tapism, and increases efficiency and transparency of government operations. Dubai government has already started to implement the technology.

Crowdfunding

It is a popular method of fundraising, for new startups and projects. In blockchain based crowdfunding platforms trust is built through smart contracts and online reputation systems, which eliminates the need for a central party who charges high fees for this service. New projects can release their own tokens that can later be exchanged for products, services or cash.

Multimedia and entertainment

Now blockchain has entered into the entertainment field where the third party interference is too much. The blockchain implementation in this fields will remove the middleman from the scenario. Online music is one of the entertainment areas where blockchain has already started their implementation
Eg; Mycelia & Ujo music

Real estate

It another important area where blockchain implementation will make a drastic change. The current real estate system is facing a lot of ownership and transfer issues. The blockchain implementation of this field can control the entire real estate systems with shared ledgers.

E.g; In India, the Andhra Pradesh state government has started implementing the complete land registration through blockchain.
There are much more other areas on the list. Like Voting, Healthcare, Forecasting, Transportation, Energy management, etc. Not only blockchain applied solutions but industry-specific blockchain development frameworks, blockchain management software, DApp and Digital Asset management software etc. also emerged along with blockchain. And many more tools are being introduced as it grows. All these tools and frameworks are making blockchain development and management easier than before. So the development and deployment of blockchain applied solution have become easier than before. In the upcoming chapters, we will discuss some of the prominent blockchain development frameworks, Blockchain development projects, Management tools and other related tools.

Cryptocurrency

In the first section, the blockchain and its structure have been discussed. Before we going to explain one of the famous blockchain (or Blockchain protocol) the Bitcoin, it would be better having a look into the terms Cryptocurrency.

The idea of ‘cryptocurrencies’ has been on the discourse since 1998 itself. The first known attempt for creating a digital cryptocurrency was B-Money and Bit Gold, but both never came into reality. Cryptocurrencies are the digital or virtual currencies working on the cryptographic principles. As the name indicates, it doesn’t have any physical existence or they are not tangible. They merely exist as a set of programming codes. Yet provides high security and usability than many existing currencies.

Cryptocurrency works on blockchain technology, we have already seen how blockchain works. In the case of cryptocurrency, the ledger keeps the track of cryptocurrency that is generated and transacted across the network. Every individual in a particular blockchain will have a unique account Id/address. The cryptocurrency is always associated with this accounts (Currency is Debited and Credited to this account).

People can manage their account through the application called wallets. Through the wallets, anyone can make the transaction to anyone on the network (both the sender and receiver must have an account). The transactions are verified by nodes and added to the blockchain ledger. So the immutable and encrypted ledger of blockchain is the backbone of cryptocurrency.

Suppose initially, my wallet has credited with 100 units of cryptocurrency. From there onwards every movement of every unit of currency will be recorded in the public ledger, every participating node in the network can watch the past as well as the present of each unit of currency in the system. Thus it will be a more transparent monetary system.

Other notable features of blockchain are also applicable to cryptocurrency; the encryption mechanism, peer to peer network, and no central authority/central server to control. Each cryptocurrency will be working on a blockchain protocol. One of the most famous cryptocurrency is bitcoin which relies on the bitcoin blockchain. And ether is another fast-growing cryptocurrency which runs on Ethereum protocol. While comparing with the traditional currencies, the cryptocurrencies provide highly anonymous nature for participants. The only visible identity of a user will be his account ID, rest everything will be encrypted. The participants will not have any idea about the real identity of a user. There are many advantages as well as disadvantages for cryptocurrency which will be discussed in the next chapter.

“

Satoshi Nakamoto

An unknown person or a group of people who first proposed and developed the Bitcoin. With nearly 980,000 bitcoins in hand, he is considered to be one of the richest person in the world. After initial involvement and support Nakamoto handed over the control of network and source code to community members and disappeared.

”

Bitcoin

Bitcoin is the first Cryptocurrency as well as the first blockchain implementation in the world. We have already discussed what cryptocurrency is. In this section, let us explore little deep into the topic with the most famous cryptocurrency, Bitcoin. The historical aspects of its creator and all have already pinpointed at many places. However, for the sake of continuity let's have a glance. Based on the conceptual framework put forward by some researchers in late 90's Satoshi Nakamoto introduced bitcoin in 2009. It does follow the exact structure of a typical Blockchain with P2P shared network, Distributed ledgers, and cryptographically protected data.

Bitcoin Working

So how someone can use the Bitcoin service? May the people are already familiar with the method. It is simple and we don't need any technical knowledge or programming skills to use Bitcoin. The first thing we have to do is create an Account in Bitcoin blockchain. For that, the simplest way is to

create a digital wallet. There is a number of wallet service providers like coinbase and BitCore. While creating an account the user has to provide a 'Key' (similar to a password). Using this key the wallet will generate a valid bitcoin Private key- Public Key pair. The public key will be visible to all and it is the visible account ID of the user. On the other hand, the user keeps the private key by himself, it is the access key to his account. If a person loses his private key he loses access to his account and his money.

Buy Bitcoin

The easiest way to own Bitcoin is to buy them from a bitcoin exchange. There are a number of online bitcoin exchanges which exchange normal currency to bitcoin. People can exchange their normal currency for bitcoin and move it to their wallet. Another method to own bitcoin is to participate in Bitcoin mining.

Transactions

Sending bitcoin from one account to another is called as a transaction. It is usually done through wallets. The wallet app will provide an interface where we can input the account Id of the recipient and the amount we wish to transfer. Once we have made the transaction, the miners will verify the transaction and add to the blockchain ledger if it is a legitimate one. In Bitcoin, the transactions are cost-free. Usually, a transaction validation time is about 10 minutes in bitcoin, but if we give a small transaction fee we can speed up the process.

Bitcoin Mining

The mining is the most important as well as the interesting topic in bitcoin. This is the process by which new transactions are validated and added to the so-called 'blockchain'. This demands dedicated mining hardware and thus, not all nodes are involved in mining. Those nodes who are participating in mining process is known as 'miners'.

When a new bitcoin transaction happens in the network that is broadcasted on the network. The miners listen to this broadcasting and engage in transaction verification. Once the transactions are verified they are added to a block.

So what do miners actually do?

Here, the mission is to find a hash value for the new block. The miner who finds the hash value first is rewarded with some bitcoins called block reward. Now it is 12.5 BTC. The reward is halved every 210,000 blocks or roughly every 4 years.

Finding hash value is not a big deal. Every node can do that. Therefore, a difficulty level is associated with it to make the nodes compete with each other. The difficulty level is a measure of how difficult is to find the hash. Difficulty level shrinks the set of hash values that a block can have. Without

difficulty level, the hash can have any of the value within the super gigantic set of 2^256 possibilities (since the length of hash = 256 bits). By associating a difficult level, the target set is reduced considerably. The difficulty level is specified in terms of a number of zeroes, which means the miner has to find a hash value which starts with a specified number of zeroes. The nodes keep finding different hash values and checks whether it satisfies the required difficulty level. Since the data of a block remains same, the hash is always same. Therefore, the only possibility to try out different hash values is by associating a nonce with the content of the block. The nonce is an arbitrary string of 32-bit length.
i.e. H(block + nonce)
Being a small target set, the probability of finding success is reduced. The miners keep changing the nonce in a brute force manner and the corresponding hash is computed each time. This is the real game and the computational power of nodes really matters here because the miners have to try out large combinations of 'Nonce'. The node which equipped with dedicated hardware and high computational power has a greater chance to win this game and get the block reward. Those who find hash first will broadcast the block along with the nonce. By receiving this, others stop mining and validate whether the received hash satisfies the specified difficulty level. If yes, the nodes show their acceptance by adding it to the blockchain.

Value of Bitcoin

The value of bitcoin has drastically increased and touched new heights in the last couple of months. So a general question that may arise in anyone's mind is 'who determines the value (or more economically speaking exchange rate) of bitcoin. As we know there is no central bank or any other designated agency to control it; then how the value is determined, or who determines it? The answer lays in the basic economics, which is demand and supply. Following is the simplest model to determine the value of bitcoin.

T : Total bitcoin transaction/second
D : Duration that a BTC needed by a transaction
S : Supply of the bitcoin
P : Price of the bitcoin
We have
S/D=Bitcoins available per Second
T/P= Bitcoins needed per Second

According to demand-supply rule, when the supply of the bitcoin increases the demand decrease consequently the price will also decrease. And when the demand increases the supply of bitcoin will also decrease, consequently the price of the bitcoin will also increase.

At an equilibrium state, where the supply S over D, is equal to the demand T over P. We can deduce the price P as

$$S/(\ D)=T/P$$

Equilibrium state:-

$$P=TD/S$$

That is at equilibrium, the price should be equal to T times D divided by S.

This is the very basic equation to calculate bitcoin exchange rate. **The value of the bitcoin basically depends on the demand and supply. However, there are many other factors including public perceptions, mining difficulty level, energy consumption for mining process etc.** that are taken into consideration while calculating the actual exchange rate. So that there will be some slight variations in exchange rate across the different market. It is evident that a single authority can’t control the value of bitcoin, rather it is determined strictly based on the user transaction.

Community, Politics and Regulations

Along with the enormous possibilities it opened, the Bitcoin (or the cryptocurrencies as a whole) poses potential threats also. The latest discourses on crypto currencies are mostly related to this aspect, especially that from government authorities and financial institutions. The cryptocurrencies can bring a lot of benefits to existing economic systems as well as the society. But an unfettered and anonymous economic regime also raises many other questions like security, illicit usage, black money etc. The discussion is still going on and both sides are upholding their own version. Here are some of the advantages as well as disadvantages of the cryptocurrencies.

Advantages

Transaction Speed

Cryptocurrencies offer very fast transaction which is far more superior than the Present banking transaction speed. Bitcoin takes a maximum of 10 minutes for validating a transaction and it is about 10 seconds in Ethereum.

Anonymity

Cryptocurrency transactions are fully anonymous and it is not possible to identify who had done this transaction or to whom this transaction is made. The participants will be using only the network address of the sender and receiver. No identity of those participants will be published in the shared ledger.

No restriction on payments

It is the most noticeable advantage of cryptocurrency. There is no restriction on transactions. The user can send the currency at anytime from anywhere to everywhere. That means no time boundaries like bank holidays.

Less /No transaction fees

The cryptocurrency transactions are normally free. Or the fee is much less than present financial transaction charges. In bitcoin, anybody can do transactions without paying any transaction fees. The user also has the option to offer transaction fees for speeding up their transaction. That is if a person is providing a transaction fee, more miners will come to validate the transaction; hence the transaction gets validated fast.

Immutable transactions

Cryptocurrencies are one of the most secure currency systems available today. It has the 'immutable' property; i.e. If one transaction had occurred in the blockchain based cryptocurrency, it is irreversible. So the chances of fraudulent transactions are nearly impossible.

Government can't De-monetize

Most of the cryptocurrencies work as a decentralized system and its exchange rate is fixed dynamically according to the demand-supply factors. No government regulation or anything can't stop such independent cryptocurrencies. The only thing that a government can do is restrict the conversion of it to normal currency. However, they can't stop the transactions in cryptocurrencies.

Secure Payment information

Cryptocurrency transactions don't use any identity of the users. They will only use the wallet address of the sender and receiver, all other information is securely hashed and no one can retrieve it back. When someone sends a cryptocurrency to another person/entity, none of the personal information will be shared with them. Only the particular amount of bitcoin will be transferred from one account to another account.

No Inflation

Most of the cryptocurrencies have a fixed number of currencies in their exchequer. In case of bitcoin, it is 21 million. Once the entire thing has mined there won't be any more new bitcoins. So there is no chance for inflation.

Disadvantages

Less Acceptance

Even though the demand for 'cryptocurrency' is steadily increasing, the point is that many governments have not given any official approval for 'cryptocurrency' transaction. And its usage is now limited some specific domains only.

Moreover, the ‘cryptocurrencies’ are still far away from the common mass.

Inconsistent rate

It can consider either as an advantage or disadvantage. Although there is a strict demand supply rule to define the exchange rate of cryptocurrencies, present market trends indicate an uncommon surge in the exchange rate of cryptocurrencies, especially that of Bitcoin. But it is believed soon that it will attain the normal pace.

Government Ban

As we said government can’t control cryptocurrencies, but they can ban it and illegalize its transaction. Of course, it cast a shadow over such ambitious, unfettered movements.

Deflation can happen

Cryptocurrencies are generally limited in number and its exchange rate is basically depended upon the supply and demand. Since most of the cryptocurrencies have only a fixed number of currencies, the possibilities of deflation are greater than any other economic system. In case of bitcoin, if someone holds the bitcoin for a long time, then the supply will reduce and still the demand will increase and it will create deflation.

Key recovery is impossible

Since most of the cryptocurrencies don’t have a central authority, every individual is responsible for keeping their account safe. If anyone loses the wallet key, no one can help them get it back.

Supports Money Laundering/Black Market

The anonymity of the cryptocurrency makes it attractive to the black market and money launderers. Since the identity is not revealed anywhere misuses are reported several times. Famous two are the “silk road” website which provides illegal drugs and other illegal items payable by bitcoin and recent ‘Wannacry’ cyber-attack.

The discussion is still going on and most of the governments have not yet formulated any direct legal frameworks regarding this. Of course, the potentials of cryptocurrency can be used to develop a more transparent economic system, but the loopholes and security threats have to be taken care of before taking such big leaps. A potential technology like this can’t be avoided forever, so we can expect a fully legalized cryptocurrency based economic system soon.

Ethereum

Ethereum is an Open Source Blockchain platform which allows anyone to develop and deploy Blockchain based Applications. **Any kind of application including cryptocurrency, tokens, wallets, social apps etc. can be developed and deployed in a Distributed Environment of Ethereum.** In other words, rather than sticking with the cryptocurrency alone, Ethereum opened the possibilities of the 'blockchain' and 'distributed ledger' technology to other application domains. Ethereum is not a single network rather it is more like a protocol for internode communication. Actually, in Ethereum many networks exist alongside. The community Ethereum Network, Community test network and other private Blockchain networks like

- Private network
- Public test network
- Main Ethereum network

> **Ethereum**
>
> The inventor Vitalik Buterin has done what Tim Berners-Lee had done to the networks. World Wide Web(WWW) brought the individual networks under a single umbrella. Similarly, Ethereum incorporated all blockchain functionalities in a single network and avoided creation of individual blockchains for each purpose.

How to be the part of Ethereum?

Basically, there is two type of users in a typical Ethereum blockchain. The one who issues a DApp (or a smart contract) and others who participates in the contract.

Every user will have an account in Ethereum, they are called Externally Owned

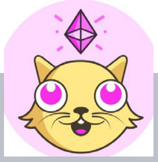
Accounts (EOA). Same way every DApp will have an account address in Ethereum known as Contract Accounts. User transaction is associated with these unique accounts. Users can make transactions with both other EOA
accounts as well as Contract Accounts.

DApp

DApp is the 'Decentralized Applications' running on the blockchain. They are the applications that run on blockchain without any centralized control. We can say bitcoin is a decentralized application that runs on Bitcoin blockchain. But it is the Ethereum blockchain that extended the scope of decentralized application and popularized the word DApp.

DApp uses the shared ledger instead of a server to record and store all the transactions. The DApps will have a set of backend codes as well as a user interface. In Ethereum, these backend codes will contain the smart contract and the front end will provide a user interface for the user to interact with the blockchain. Once the smart contracts are deployed on the blockchain, then the DApp will become accessible in the blockchain. Then any node in the blockchain network can use the DApp.

The DApps can be developed for any business use cases. Any application that is currently running on the client-server model can be implemented as a DApp. Some examples of Ethereum DApp:- Green Ether Project, splitcoin, The immortals



CryptoKitties

CryptoKitties is the first game in Ethereum blockchain. A participant can buy and sell crypto kittie token from the issuer. The transaction will be done in Ether. The game had witnessed an unprecedented demand from buyers and Ethereum network was flooded with the transactions.

Components of Ethereum

Smart contracts

Smart contracts are the nerves of Ethereum blockchain framework. All the operations in Ethereum are controlled with smart contracts. Smart contract is the digital version of contracts; which is executed automatically upon satisfying predefined conditions. Of course, they are lines of codes and it is used to exchange anything of value in a more secure and transparent way.In Ethereum, these smart contracts are written in solidity programming language. The smart contract will provide the direct contract execution between sender and receiver without a middleman.

Working of a Smart Contract.

- First, a contract account is created in Ethereum blockchain. The contract will have specific rules and actions based on that rules.

- The contract is then coded. In Ethereum, the smart contract is coded in Solidity; an Ethereum compatible high-level language.

- The coded contract is deployed in Ethereum network. The deployed contract will have a unique public-key address, the address is used to reach the contract in the network. Once the contract is deployed in, it can't be modified even by the Issuer.

Ether

Every collectively run network need some fuel to exist. Bitcoin is the fuels of bitcoin network and Ether is the fuel of Ethereum. Ether is the cryptocurrency of Ethereum network, and it is the backbone of transactions in Ethereum. Ethereum website put it in this way "Ether is a form of payment made by the clients of the platform to the machines executing the requested operations”. Similar to the blockchain, the Ethereum network exists in a consistent state because of the computational and other resources spent by individual nodes, Ether is the reward provided to those individual nodes. As more people getting interested in Ethereum, the value of Ether is also surging on daily basis. Today, Ether is the most demanded cryptocurrency after Bitcoin.

The initial supply and rate of issuance of Ether was determined during the presale took place in 2014. Other than the initial supply (which is about 72 million) new ether coins are issued whenever new blocks are created. But this issuance method will possibly change when ‘Ethereum’ adopts new consensus algorithm.

Ethereum Clients

Ethereum Clients are the tools used to connect to the Ethereum blockchain for developmental or mining purposes. Some of the Ethereum clients are listed below.

- Geth — Geth is an Ethereum client working in GO language. Geth has a command line interface (CLI) tool that communicates with the Ethereum Network and acts as the link between the different nodes in the network.

- Eth — C++ Eth is a powerful Ethereum client which is more focused on miners.

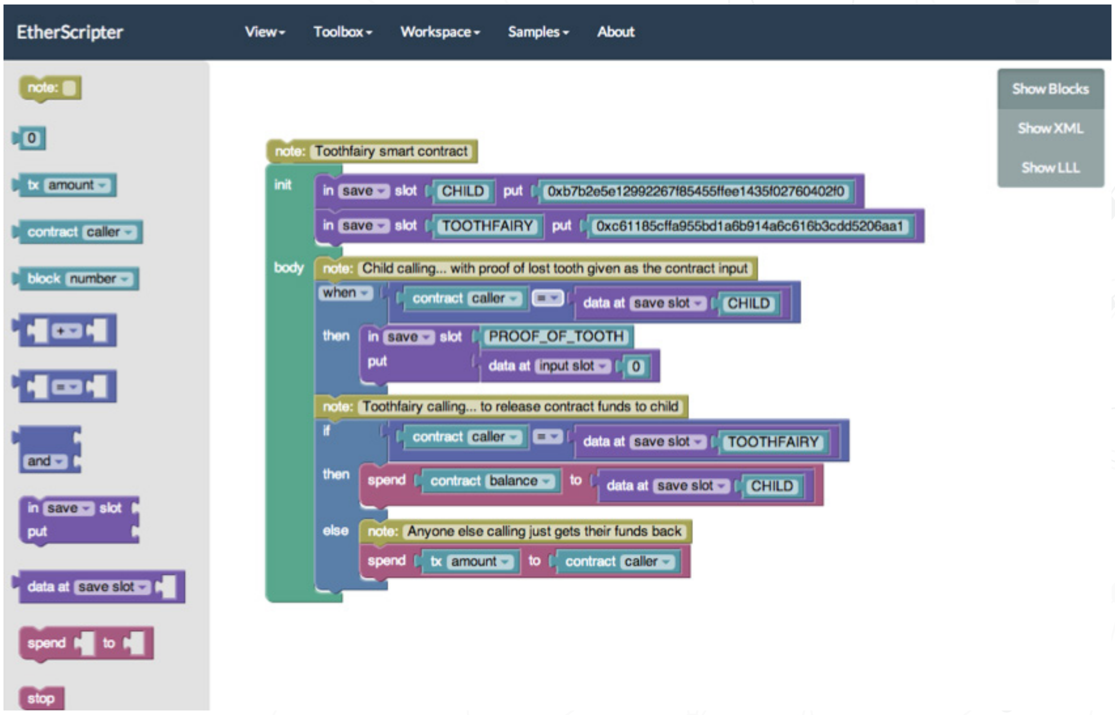
- Pyethapp — this client is useful for DApp development using python. 'Pythapp' is also an excellent choice for research and academic purpose in Ethereum blockchain.

EVM

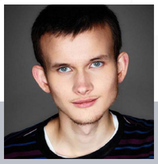
The EVM is the engine behind the whole Ethereum blockchain. Smart contracts are run on the Ethereum Virtual Machine (EVM) - the decentralized, consensus-driven computer which distinguishes Ethereum from earlier Blockchains. This Virtual Machine runs its own language of bytecode. For this reason, several languages for writing contracts have been developed. Of these, the most popular one is Solidity. Solidity is a JavaScript-like language developed specifically for writing Ethereum Smart Contracts. The Solidity compiler 'sol-c' turns this code into Ethereum Virtual Machine bytecode, which can then be sent to the Ethereum network, as a transaction to be given its own address. Every participating node will have an EVM installed in it.

Etherscripter

Etherscripter is a visual smart contract builder tool in Ethereum. It provides a GUI for creating smart contracts in simple steps. Etherscripter provides a simple drag and drop interface where the corresponding backend codes in Serpent, LLL, and XML will be generated automatically. Using Etherscripter even a non-programmer can create smart contracts.



```
note: Toothfairy smart contract
init
    in save slot CHILD put 0xb7b2e5e12992267f85455ffee1435f02760402f0
    in save slot TOOTHFAIRY put 0xc61185cffa955bd1a6b914a6c616b3cdd5206aa1
body
    note: Child calling... with proof of lost tooth given as the contract input
    when contract caller = data at save slot CHILD
    then in save slot PROOF_OF_TOOTH
         put data at input slot 0
    note: Toothfairy calling... to release contract funds to child
    if contract caller = data at save slot TOOTHFAIRY
    then spend contract balance to data at save slot CHILD
    else note: Anyone else calling just gets their funds back
         spend tx amount to contract caller
```



> ## Vitalik Buterin
>
> Vitalik is the creator of Ethereum. He first discovered blockchain and cryptocurrency technologies through Bitcoin in 2011, and was immediately excited by the technology and its potential. He co-founded Bitcoin Magazine in September 2011, after intensive researches about blockchain he wrote the Ethereum white paper in November 2013. He now leads Ethereum's research team, working on future versions of the Ethereum protocol.

.

Hyperledger

Hyperledger is a collaborative effort from different industry leaders to frame an open source, Cross-Industry Blockchain aided technologies. The movement basically aims to develop the distributed ledgers that can support enterprise-level business transactions. The entire project is developed on the open source platform. Even though the project is hosted and driven by the free folk of the internet 'Linux Foundation', technology giants like IBM, Intel, Samsung and many more others already became part of the project.

The project was announced in December 2015 by Linux foundation, and soon it became popular as leaders from different business domains like banking, healthcare, finance, supply chain, IoT, manufacturing etc. joined the movement. As of now with 170+ members, the project is the largest blockchain technology consortium and it is entirely funded by its members. Linux Foundation does not stipulate a single blockchain standard for the participants, rather they choose a community-driven approach to develop blockchain technologies. By early 2016, the project began accepting proposals for incubation and later a number of different business blockchain frameworks and tools were accepted for incubation under this project.
Under the project following frameworks have been unveiled so far.

- **Iroha**
- **Fabric**
- **Sawtooth**
- **Burrow**
- **Indy**

Another important thing to point out is the difference between Hyperledger and

Digital Tokens

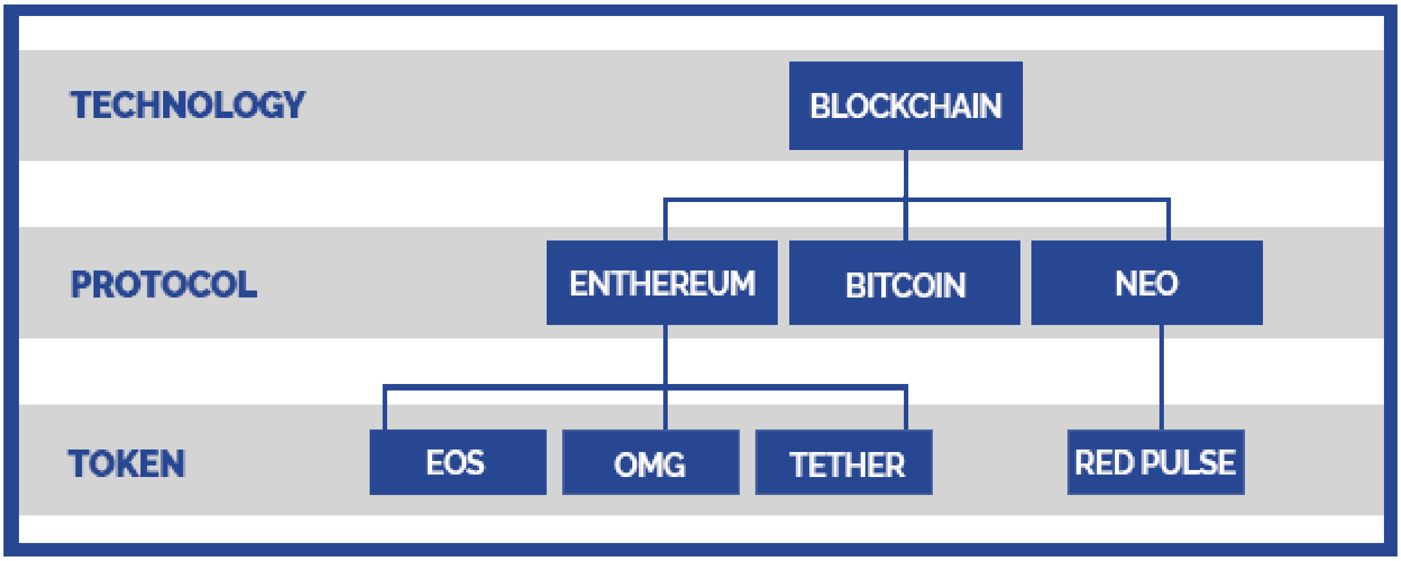
Digital tokens or simply Tokens, are another trending blockchain based application which is shaking the market. So what are they? Tokens are a slight variant of cryptocurrencies; they are a digital asset which is built on top of cryptocurrency of a blockchain network. The token can be used to provide a right, to pay for a service or to transfer data, as an incentive, as a gateway to extra services or any other purposes. In other words, a token can be used in whichever way the developer/ the developing organization decides.

The tokens can be mainly classified into two; Utility tokens and Equity tokens. Utility tokens or user tokens will provide some future access to a product/ service to the user.

Equity tokens are a subcategory of security tokens that represent ownership of an asset, such as debt or company share etc. Equity tokens can be considered as an investment.

The tokens will never be used as a cryptocurrency rather it is a digital asset which is less liquid than cryptocurrency. So whenever a token is created, its value also will be defined. In some cases, the tokens are refundable, that is we can exchange the token with a cryptocurrency. Like the cryptocurrency, the tokens are also managed with wallets.

See the below image to get a better understanding about tokens.



There are different blockchains exist today, like Ethereum and Bitcoin. We can collectively call them as blockchain protocols. Every blockchain based application including cryptocurrency and Tokens will be working on any of these protocols.

How Tokens are created.

Currently, most of the tokens are created on the Ethereum blockchain. Token creation using Ethereum blockchain is a simple process. In Ethereum, token creation is nothing but the creation of a smart contract. A Developer can simply create tokens using the development standards like ERC 20. Presently, there are more than 10000 tokens are available in the ERC20 token standard. The ERC 20 has a predefined structure for smart contract creation. For developing the tokens one just have to add necessary codes to the structure. Once the token is completed, it can be deployed in Ethereum network

ICO (Initial Coin Offering)

ICO (Initial Coin Offering) is similar to IPO (Initial Public Offering) in the case of stock markets. The ICO is generally used for crowdfunding. A company/an entity will offer a new Cryptocurrency/Token to investors against any other cryptocurrency or fiat currency. The investor can keep the token and exchange it in future for a service/product/anything that is assured by the issuer. The return on any token depends upon the smart contract of the token (which is of course defined by the issuer). Like the shares in IPO, the tokens issued through ICO is tradable also. The value of the token changes corresponding to its demand, the token holder can trade the tokens in such situation.

ICO has become a popular way of fundraising for startups, charity, and many other programs. Unlike any other written contract, the smart contracts of a token can't be modified by the issuer over the time. So ICO offers an assured return for investors if the issuer finds their fortune.

Many companies and startups accumulated massive funding in recent past through ICO. Some examples are,

OmiseGO

OmiseGO (OMG) token was issued by the Omise aimed at raising funds for the creation of a decentralized platform for the exchange of fiat money and crypto-currencies. Initial coin offering helped the project to raise $19 million. Any user of Omise GO will be able to conduct financial transactions such as payments, remittances, payroll deposit, B2B commerce, supply-chain finance, loyalty programs, asset management and trading, and other services, in a decentralized and inexpensive way.

EOS

EOS is token issued by the company ‘Block. one’ to conduct the Proof of Stake mechanism in blockchain development. Basically, EOS token holders vote for the block producers, which mine blocks and decide on major events in the EOS ecosystem.

Tether

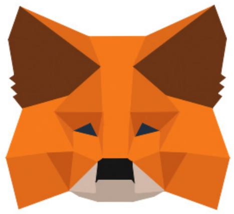
A method to maintain a one-to-one reserve ratio between a cryptocurrency token, called tethers, and its associated real-world asset, fiat currency.

SECTION-2

Cybrosys Limited Edition

BLOCKCHAIN

E-BOOK



MetaMask

MetaMask is a web browser add-on which enables anyone to run the Ethereum DApps without running the Ethereum full node. An Ethereum full node installation will take a lot of memory as well as time; so Metamask is a tool that eliminates the overburden of this hectic installation task. Initially, Metamask was available only for Google Chrome, but now it is available for Firefox and other popular web browsers.

MetaMask add-on for chrome can be added from chrome web store or from 'metamask.io' website. This MetaMask add-on provides a user interface for interacting with the blockchain. The user can connect to the Ethereum main network or 'testnet' or he may create his own private network and run DApps on the blockchain.

In normal case, a web3.js (the JavaScript API for Ethereum DApps) must be installed in the local system to interact with the Ethereum DApps. Web3.js is a collection of libraries used to interact with local or remote Ethereum node using Http or IPC connection. But MetaMask will inject the web3.js to each page for accessing the Ethereum blockchain by itself. This approach eliminates the effort of web3.js installation in the local system.

After adding the Metamask, the user can interact with Ethereum blockchain as normal. The user can create an account, access Ethereum DApps, or deploy once own DApp. MetaMask retrieves data from the blockchain and allows the users to manage the data securely.

The Metamask provides a vault account for each user, this vault secures, stores and tightly controls access to tokens, password, certificates, API keys and other elements in blockchain apps. The vault account act as a second level encryption for the user account.

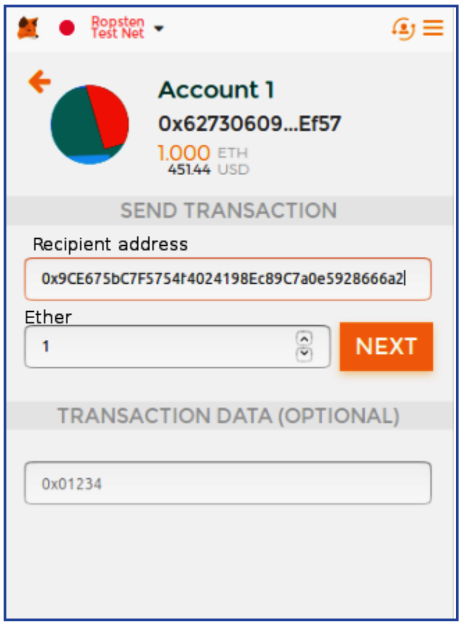
Wallet Seed

The Metamask will provide a group of 12 words known as “wallet seed” while installing it. It is the user credential and it must be stored somewhere safe. The users can also create passwords for their account. The wallet seed or the password is necessary to log in to the MetaMask. The vault account will encrypt the user metadata and securely store it in the browser itself.

MetaMask Transactions.

The Metamask user interface has a default buy and send option for buying and sending Ether. The user can access his wallet, buy or send ether, check his balance and transactions from this interface. When the user executes a transaction from the Metamask it will send the transaction to the respective blockchain network. Then the corresponding validation and confirmation will occur in the blockchain as usual. In the case of ‘testnet’ and main network, the user can see the transaction details and confirmations in the ‘Etherscan.io’.

Here is an example of a transaction of Ether through Metamask.
For sending Ether to an account you have to specify the recipient address and the amount to be transferred in the provided interface.


Ropsten Test Net

Account 1
0x62730609...Ef57
1.000 ETH
451.44 USD

SEND TRANSACTION

Recipient address: 0x9CE675bC7F5754f4024198Ec89C7a0e5928666a2

Ether: 1

TRANSACTION DATA (OPTIONAL)

0x01234

Before linking your transactions to the blockchain, the web3.js will ask your permission and the transaction will be submitted only after your approval.

Once the user submits the transaction, it will be sent to the blockchain for validation. The transactions will be broadcasted to the nodes and once the validation is completed you can see the transaction details in the etherscan window. The window will display all the information regarding that transaction i.e.; block number, hash value, sender and recipient address, number of confirmations, gas units, transaction cost, nonce etc.

GNOSIS, Maker(MKR), Token Factory, CryptoKitties etc. are some DApps that supports Metamask. Any developer can submit DApp in Ethereum with Metamask support so that the user doesn’t need to install the full Ethereum node for accessing the particular app. The Metamask is the very useful tool for accessing Ethereum in low bandwidth networks. Let’s hope the tool will expand the reach of ‘Ethereum’ to more people.

Fabric. Often people are confused with these two terminologies. The fabric is one of the protocols under the hyperledger project and Hyperledger is the name of the project itself, not a technology.

So if we recall the concepts discussed so far, Bitcoin is a blockchain network which offers only one kind of application that is bitcoin itself. Ethereum is a more versatile blockchain platform where anyone can develop and deploy the blockchain based application. Hyperledger is a project which aims to develop different enterprise level distributed ledger technologies.

The tools that are currently under hyperledger project

- Hyperledger cello
- Hyperledger composer
- Hyperledger explorer
- Hyperledger quilt

The project along with its participating technologies are often referred as 'Hyperledger Umbrella'

Objectives of the Hyperledger Project.

- To enable its member organizations to build robust, enterprise-level applications, platforms and hardware systems based on blockchain technology
- To advance the use of blockchain technology in business by developing a cross-industry level, open-source development library
- To facilitate building custom distributed ledger solutions for its members.
- To integrate independent blockchain protocols.

Mist

Mist is an Electron framework based application which is used for the management of Ethereum wallet and Ethereum applications. More specifically, it is a hybrid desktop application with a web interface to manage Ethereum DApps (Decentralized Apps). Mist is similar to a web browser like Chrome or Firefox, but a Web 3.0 edition. Mist is an all in one software to manage all the assets and contracts of an individual in the Ethereum Blockchain. Mist can browse DApps, manage contracts, manage ether and other digital assets etc. It acts like a window to the Blockchain network and access different Apps and Services provided in the network. Mist is also the official wallet of Ethereum blockchain to manage 'Ether'. Mist is developed and maintained by Ethereum team and it is still in beta stage, hence it may have problems. It is recommended to use the latest and updated version of the official Git repository of Mist.

Mist installation is similar to any other software. Download the 'Mist' executable file appropriate to your operating system from Git repository and install it like any other application. Since Mist is a client based application, it has to download and save the entire Ethereum Blockchain to the local system. As of now, there is at least 5GB of data and it will increase as the chain grows. So the downloading and installation process is a big task.

Mist wallet

Mist wallet is the official wallet of Ethereum Blockchain so the Ethereum currency Ether can be stored and managed with it. Since Mist can serve both as a wallet and asset management tool it is easy to engage in Ethereum transactions using mist. Another advantage of using Mist is the security. The wallet is purely designed and developed by Ethereum itself not by any third party. Mist wallet provides two type of service namely 'Simple Wallet' and 'Multisignature wallet'. In Both services, the users can manage their funds, ethers, as well as the contract in simple steps. But the 'Multisignature wallet' can provide some extra layer security to its user.

An important thing to note while using a Mist wallet is about syncing process. Always ensure that the entire Blockchain is synced with the local system. Incomplete syncing may invite critical problems during the transaction.

Truffle

Truffle is a blockchain development framework dedicated to Ethereum blockchain platform. This open source framework was developed by Consensys with an objective to simplify the blockchain and DApp development in Ethereum platform. Truffle is a well-equipped framework which makes Ethereum platform more user-friendly and interactive. Following features of truffle make it the best choice for developments in Ethereum.

Features of Truffle

1) Automated tools

- Truffle provides built-in smart contract creation tools, which will perform compilation, linking, deployment and binary management of the smart contracts. It offers a development environment with a testing framework and digital asset pipeline. The truffle reduces the complexities of team-based development, testing, deployment, and migration activities in an Ethereum project. In total, smart contract creation and testing which are two tedious tasks in blockchain development have become easier with truffle.

2) Scriptable

- Along with those automated tools available, the truffle is scriptable too. Which means the developer is allowed to add more scripts to the project during the development. For running these scripts a ‘script runner’ is also available in truffle, which can run both external and internal scripts.

3) Networking

- As the blockchain is growing day by day extensions and migrations have become a frequent requirement in the development environment. The truffle framework is fully capable of integrating an infinite number of networks to it and will completely support migrations and extensions of networks. The configurable build pipeline in truffle supports tight integration within the network.

4) Package support

- Truffle provides two package management tools, Ethpm & Npm. Ethpm is the Ethereum Package Manager and Npm is Node Package Manager These package includes several modules and set of predefined codes which can be utilized while scripting.

5) User interaction

- Truffle has an interactive console for the direct contract communication. Through this console, the developer can create, compile and test the smart contracts. The console also manages the communications between the user and smart contracts.

Another interaction environment available in truffle is the browser portal. The developer can use the default account in truffle to use the browser environment. It is possible to see the local transactions, pending requests etc. from there.

Development-Truffle boxes

Before installing a truffle environment owe must install an Ethereum client like ‘Geth’, ‘WebThree’, Parity’ or any other. Truffle is installed as a separate project above Ethereum client and it can be deployed on it without any extra configuration.

After installing the Ethereum client, create a project directory for truffle using the command

mkdir projectfolder’

To initialize the project (i.e. truffle) navigate to the created directory cd projectfolder

And run following command.

init truffle

If the initialization is complete then the following project structure will be created

1. **Contracts**
2. **Migrations**
3. **test**
4. **truffle.js**

By default, there will be some sample contracts and set of sample projects in truffle environment to get acquainted with the truffle environment.

Truffle Box

In truffle, each individual project is called truffle boxes. The truffle box will contain required modules, front end views, solidity smart contract libraries of a project etc..

As mentioned earlier, Truffle comes with some sample Truffle Boxes available. The user can unbox and run them in the test environment.

Creating a Truffle Box

Users can create their own truffle boxes in the truffle environment. They can either create truffle from scratch or can add an existing project to the box and develop from it. If the project is starting from scratch a 'blueprint truffle box' is readily available in the truffle. The 'blueprint truffle box' will contain all the necessary configuration files and common values for a truffle box.

For developing truffle boxes the user need a

1. **Github repository,**
2. **Configuration file,**
3. **Optionally small and large images for the boxes listing.**

The truffle configuration file name is *truffle-box.json*. It contains 3 attributes ignore, commands and hooks

Ignore: An array attribute contains the list of files to be ignored while unboxing.
Commands: Object attributes of a key-value pair, contains the list of files to be compiled or migrated. After unboxing these files are visible to users.
Hooks: Object attribute which contains a list of commands need to be executed when unboxing.
The blueprint contains all the basic components needed for developing a truffle box. The developer can delete the default sample files and images from the box and can create new files. The configuration file is also customizable.

Community truffle box

Truffle is already providing several official truffle boxes developed by the truffle developers. Along with that a vibrant truffle community is also actively contributing new truffle boxes. Any individual can contribute to the community by sending the developed truffle details and GitHub repository details to truffle community. The Box will undergo a screening and if it is compatible with truffle then it will be published as a community truffle box

Embark

Embark is a framework used for developing and deploying DApps (Decentralized Apps) using one or more decentralized technologies. The tools and functionalities provided by the Embark make the DApp development process easy and productive. It reduces the interaction between the front end of application and smart contract so that the application can run faster on the network. The technology uses IPFS protocol to store and manage files across the decentralized network. 'Whisper' and 'Orbit' communication platforms streamline the communication process in Embark. The platform performs automatic deployment of smart contract and ensures the redeployment if the contract has undergone any changes. Embark currently integrates with Ethereum, decentralized storages like IPFS, and decentralized communication platforms like Whisper and Orbit. Using embark, it is possible to manage different chains like testnet, private net, livenet etc. And the smart contracts in solidity and serpent can be built and it is deployable with embark.

How to Install embark?

Prerequisites:

- geth (1.5.8 or higher)
- node (6.9.1 or higher)

Open terminal and run the below-given codes

```
npm -g install embark
```

To run it on a simulator instead of a real Ethereum node, run the code

```
npm -g install ethereumjs-testrpc
```

Creating a DApp with Embark

Generally, a typical DApp in embark consists of 2 sections.

1. Smart contracts

this section will contain the business logic of the DApp. It is written in either solidity or serpent

2. User Interface

the user interface through which the user interacts with the app

To create a DApp in embark simply run this code

```
        embark new <DApp name>
```

This code will initialize a predefined template and creates a directory structure

Solidity

Solidity is a high-level programming language, which is designed to work with the technology of the time -Blockchain. More specifically speaking, Solidity is designed to develop Smart Contracts in Ethereum Blockchain platform.

Smart Contracts

A contract in the sense of Solidity is a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum Blockchain. In each Contract, we can define State Variables, Methods, and Events etc. This contract can manage transactions between blocks in Blockchain network. Each block has a particular address in the form of a cryptographic key that generated by the result of some functions including hashing of adjacent blocks. This creates a strong relationship between adjacent blocks. So that manipulation or any other form of hacking in nodes or blocks are not easy or not even possible. Solidity is one of the many languages that can be used to develop EVM (Ethereum Virtual Machine) understandable bytecode. There are many built-in classes and Libraries in Solidity which support hassle free smart contract development. You can use the IDEs like Remix, Visual Studio (With Solidity extension), Ether atom, IntelliJ IDEA plugin (both with Solidity extension) to develop.

Following are some of the features of solidity which are very similar to common high-level languages like Java and C++.

Statically typed Language

Though it is having a structure of JavaScript, unlike JavaScript it is a Statically Typed language. For example, you must declare the type of a variable like in C++ and Java before it is used. Otherwise, a compile-time error will be generated

Contract and Interfaces

‘Contract’ is a unique data structure of Solidity language, it helps to create and manage contracts easily. Contracts can be inherited by child Contracts and can create complex contract structures.

Function Modifier

It is similar to function override in other OOP languages. Suppose you want to execute a function in a different way when a condition is met. In this case, you can use function modifier feature to change the behavior of the function. Generally useful in inherited Contracts to override the parent function behavior.

Events

Events are used to write information from contract to Blockchain log. The event is similar to a function which takes the data as the argument and writes to Blockchain client's log.

Access Specifies

This is similar to the access specifies in other OOP languages like private and public. In Solidity the name and access rights have some change. 'Owned' and 'mortal' are two access specifies in Solidity. There are some more options available to extend security.

Explicit Type conversion

You can perform explicit conversion of different data types. Such conversions are generally checked at compile time, but there are exceptions also.

Memory Arrays

Dynamic arrays can be directly allocated to memory.

Libraries

You can access the vast built-in Contract Libraries, which can be used to develop your custom contract.

Import

With 'import' keyword you can import other source files to your contract.

These are some very basic features you can find in Solidity, refer more and understand the simplicity of Solidity.

A sample contract in solidity.

```
pragma solidity ^0.4.0;
contract SimpleStorage
{
            uint storedData; // State variable storedData declared as uint
(Unsigned integer)
            function set(unit x)
            {   storedData=x;
              }
            function get() constatnt returns (uint) {
              return storedData; }
}
```

This represents a contract in Blockchain.

Here, each block has the accessibility to ‘set’ and ‘get’ the values

If we need the restrictions like ‘only the owner of the contract can set the value’ then we can provide accessibility only for the owner of the contract or block, an example is below.

```
pragma solidity ^0.4.0;
contract SimpleStorage
{
            address owner; // Here ‘Owner’ holds the address of the current
block.
            function SimpleStorage() // The constructor  runs only once and set
the address of particular contract to the ‘Owner’
            {
            owner=msg.sender
            }
            uint storedData;
            function set(unit x) {
            if(owner!=msg.sender) // when an address other than the ‘Owner’ try
to access the ‘Set’ function, the program deny access
            {
                return “Sorry you can’t modify this data”
            }
            else
                storedData=x;
              }
            function get() constatnt returns (uint) {
            return storedData;
            }
}
```

Like this, we can modify the permissions of different blocks in a Blockchain

HYPERLEDGER
FABRIC

Hyperledger Fabric

Hyperledger Fabric is a blockchain framework implementation initially developed by Digital Asset and IBM and now hosted by Linux Foundation under the hyperledger project. Fabric joined the hyperledger project for incubation in the early 2016 and after 1 year of incubation, it became the first project get into the ‘active’ state. On July 11, 2017, the hyperledger Technical Steering Committee announced their first production-ready distributed ledger codebase, Hyperledger Fabric V1.0.

The Fabric platform is intended as a foundation for developing blockchain applications, products or solutions. The fabric is a Private and Permissioned system which delivers a high degree of confidentiality, resilience, flexibility, and scalability. It adopted a modular architecture and supports pluggable implementations of different components like consensus, membership services etc. Like other blockchain technologies, Fabric has a ledger and smart contracts. **The smart contract in the fabric is known as chaincode** and it is in the chaincode the business logic is embedded. The following features impart a high degree of security and privacy for the fabric framework.

Channels: The private blockchains built on fabric protocol. Each channel consists of only the authorized partners

Visibility settings: It is possible to restrict who can see the input data using visibility settings

Data Encryption: The data can be Hashed or Encrypted before calling the chaincode

Data Access Restriction: By embedding access controls into the chaincode logic, it is possible to restrict data access to certain roles in the organization.

File encryption: Ledger data at rest can be encrypted using file system encryption, and data-in-transit can be encrypted using TLS protocol.

Fabric v/s Ethereum

Both the Fabric and Ethereum are two blockchain technologies available today. But there are several differences in Structure, Mode of Operation and many other aspects of both technologies. It is worth to understand the difference between these two.

Governance:

As discussed earlier, Fabric is governed by the Linux Foundation and Ethereum is hosted and governed by Ethereum Developers community.

Platform:

Ethereum is a generic blockchain platform for any kind of application whereas Fabric is a modular blockchain platform which is customized mainly for supporting different business domain

Targeted Crowd:

Ethereum is targeted towards applications that are distributed in nature, on the other hand, the fabric is meant for business domains including banking, health, goods delivery etc.

Network Structure:

In Ethereum, network partitioning is not possible. Therefore, the transactions are visible to everyone in the network including the competitors. Obviously, it is not suitable for the business environment. Fabric offers a provision to create private networks, called channels. Each channel consists of only the authorized partners. In addition to that, there is ‘Endorsing Node’ which are designated to approve a transaction

Mode of operation:

Ethereum is a permission-less system, which means any node is allowed to participate in the network. On the other hand, Fabric being a private and permissioned system, the members who are enrolled through MSP (Membership Service Provider) can only participate in a network

Consensus:

In Ethereum consensus is based on PoW (Proof of Work) scheme. All participants have to agree upon a common ledger and all participants have the access to all entries ever recorded. This may affect the transaction processing as the network grows. In Fabric, the consensus can be achieved in different ways. The process of endorsing is based on an endorsing policy. Nodes can take a certain role called Endorsing Peers (endorser), who are responsible for endorsing transactions based on the policy. All this results in a better performance and security.

Smart Contracts:

In Ethereum, business logic is included in smart contracts written in Solidity language. In fabric, smart contract is implemented using chaincode and it can be written in languages like Go, Python etc.

Currency:

In Ethereum, there is a built-in cryptocurrency called Ether. Digital tokens for custom use-case can be also be built on top of ether. In fabric, since the consensus is not reached through mining, there is no need of built-in cryptocurrency. However, it is possible to develop a currency or a digital token for specific use-case.

Even though a full-fledged application development process has not yet taken place in Fabric, the initial results and the features it provides indicate that the Fabric is a promising technology for future enterprise level application.

Hyperledger Iroha

Hyperledger project has attracted the attention of many MNCs to co-develop blockchain frameworks suitable for different fields. Under the hyperledger project, 5 frameworks have been unveiled so far and Iroha is one of them. It is one of the blockchain platform implementation inspired by hyperledger fabric. It was initially developed by Japanese startup Soramitsu, Hitachi, NTT Data and Israeli startup Colu and now it is hosted by Linux Foundation. Iroha joined the hyperledger project in October 2016 and became the 3rd project under the hyperledger umbrella. On May 2017, the hyperledger Technical Steering Committee changed the state of Iroha from 'incubation' to 'active'.

Iroha provides a development environment for C++, web, and mobile application developers so that the developers can contribute not only to Iroha but to the whole Hyperledger Project. It allows the developers to create reusable components in C++ and that can also be called from other languages like Go. Iroha includes some common use cases like deploying new currencies, sending messages, etc. in its core framework.

The consensus methodology adopted in Iroha is Sumeragi which is a new chain-based Byzantine Fault Tolerant consensus algorithm. Sumeragi is found to be one of the fast executing consensus algorithms. The robust libraries of reusable components provided by Iroha are highly useful in custom project development in it. Some of them are.

- Sumeragi consensus library
- Ed25519 digital signature library
- SHA-3 hashing library
- Iroha transaction serialization library
- P2P broadcast library
- API server library
- iOS library
- Android library
- JavaScript library
- Blockchain explorer/data visualization suite

Components of Iroha

- Iroha is composed of the following:
- Iroha core
- Iroha Native iOS Library
- Iroha JavaScript Library
- Iroha Native Android Library

Iroha core includes the distributed ledger infrastructure, data membership services, consensus algorithm, peer-to-peer network transmission, data validation, chaincode infrastructure etc.

Features of Iroha

- Simple construction
- Modern, domain-driven C++ design
- Emphasis on mobile application development
  It includes libraries for android, iOS and JavaScript
- Creation and management of custom complex assets
- User account management

HYPERLEDGER
SAWTOOTH

Hyperledger Sawtooth

Hyperledger Sawtooth is blockchain development platform initially developed by Intel and now hosted by Linux foundation under its hyperledger project. Similar to other hyperledger projects sawtooth is also a highly modular open source platform for blockchain development. **The sawtooth is mainly proposed for the insurance claim processing, IoT assisted supply chain management, and international remittance.** Sawtooth already developed some versatile DApps in the above-mentioned areas.

Components of sawtooth

Every distributed ledger must contain the following basic components:

- A data model:-To represent the current state of the ledger.
- The language of transactions : To change ledger state.
- Protocol:-To makes consensus among participants.

The versatility, security, and simplicity of a hyperledger depend upon the features of these three components. The sawtooth has a well-defined structure for these components**. In sawtooth, the data model and language of transactions are combined together as a 'transaction family'.** So each transaction family will have a specific data model for defining the ledger state and a language of the transaction for changing the state. **The user can create their own transaction family according to their ledger requirements.**

Some transaction family models are

Validator registry

- Contains the methodology for registering ledger services

Integer Key

- Used for testing deployed ledgers

Settings

- For storing on-chain configuration settings

Identity

- Handles on chain permissioning for participants and validators. i.e.; for managing identities for the list of public keys.

The user can create the transaction families based on any of the above models. There are some readily available transaction families for some specific areas.

Smallbank
Used for performance analysis and benchmarking
Seth
Used for creation and execution of smart contracts
Blockinfo
Used for storing information about the configurable number of historic blocks.
XO
To play tic-tac-toe
Supply chain
For tracking objects
track & trade
For tracking ownership, custodianship

Consensus in Sawtooth

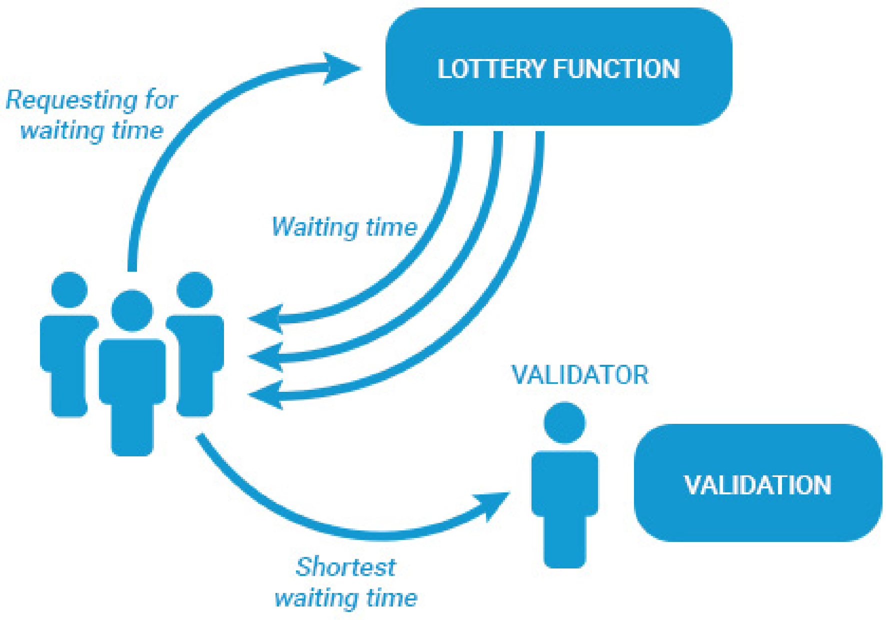
Sawtooth follows some special protocols which will ensure a universal agreement on transaction validations. This vigorous consensus method differentiates sawtooth from other platforms. In sawtooth blockchain, there will be clients and validators. **The validators are selected randomly for each transaction. It employs an algorithm to choose the validators, then the selected validators will engage in transaction validation process.** Sawtooth provides two consensus implementation method they are **Dev**-mode and **PoET.**

Dev-Mode

It is a simplified random leader algorithm used for consensus implementation. This is useful for developers and testers of sawtooth blockchain apps.

PoET (Proof of Elapsed time):

Which is a production-grade protocol for large networks, were consensus-making is a bit heavy task. In this method, the network uses a lottery functions for implementing consensus. A lottery algorithm is used for finding the leaders from a set of participants in a blockchain. In the PoET algorithm, all the validators will request for a wait time to an enclave function. Once they got the wait time, the validator with the shortest wait time will consider as the leader for validating that transaction. The function is broadcasted to all eligible participants in the blockchain to maintain fairness in the selection process; so that the leadership will be distributed randomly among these validators.



DApps in sawtooth

The hyperledger sawtooth already implemented some DApps. The Main apps are

1) Seafood supply chain traceability.

2) Bond asset settlement.
3) Marketplace digital asset management.

Seafood supply chain traceability

This application will provide the traceability and accountability of seafood from the harvest to the end customer. Basically, it works on IoT, the physical objects are traced through the blockchain network. The sawtooth app will trace the entire journey of a seafood from the ocean to the dining table with the help of IoT sensors. It can trace the ownership, possession and other parameters like location, temperature, humidity, motion, shock, and tilt etc.

The customers will also get the opportunity to trace the 'journey' of their food. They can verify the quality themselves through the blockchain network.

Bond Asset Settlement

This will provide a secure and trustworthy solution for the bond settlement. Sawtooth provides bond asset settlement application with highly secure smart contract facility. The app provides a user interface to create, buy, sell and settle the bonds.

Marketplace Digital Asset Exchange

This application provides an infrastructure for transferring digital assets. The user can easily create participants and asset, and can easily manage all the asset transactions.

Maybe it is too early to make a definite conclusion on future of sawtooth. However, by considering the features it provides and the developments happened so far, we can definitely say that the sawtooth has the potential to make game-changing applications.

Cello

Perhaps, the complexities, as well as the benefits of a blockchain is clear to most of the aspiring learners. And most of the minds are stuck up with its complexity more than the enormous possibilities. But many tools are being created daily to transform the blockchain management process as simple as any other software handling. Hyperledger cello is one of the tools in this category which makes blockchain management functions easier for normal users. The Cello acts as a middle layer in between the Blockchain and the blockchain infrastructures.

The Cello toolkit is essentially helpful for Implementing Blockchain as a Service (BaaS), as it eases the effort required for blockchain management such as create, manage, and terminate etc.

How it works?

Cello creates a layer in between the Blockchain and the blockchain infrastructure and provides automatic multi-tenant chain services. Operators can create, manage and perform other blockchain operations in simple steps by the dashboard provided by the cello.

Features of Cello

- Cello can manage the different stages of blockchain such as create, delete, terminate etc.
- Can create customized blockchain networks
- Supports various kinds of virtual machines and containers
- Supports heterogeneous system architecture
- Automatic monitoring and screening of blockchain are possible.

Cello operator dashboard

The Hyperledger cello helps blockchain developers in various aspects. **The main attraction of Hyperledger cello is the quick creation of Private Blockchain The developers can quickly create blockchain and provide BaaS from scratch without doing the complex programs.** The customization feature allows the user to decide the basic features like size of the chain, consensus mechanism, hosts etc. Through the dashboard the operator can manage a number of blockchains on top of virtual clouds, container clusters, bare metals etc. (e.g., Swarm, Docker, Kubernetes,). The tool support functionalities like adjusting the chain numbers, check system status, scale resources and so on.

The hyperledger cello can bring great flexibility and scalability in Blockchain network management. Currently, most of the cello services are available only for hyperledger fabric. It will expand its services to other hyperledger frameworks like Sawtooth, Embark etc.-

Comparison of Bitcoin, Ethereum and Hyperledger

So far we have discussed the popular blockchains like Bitcoin and Ethereum and the Blockchain development framework Hyperledger. We also peeked into some of the enterprise level blockchain frameworks under Hyperledger project. To get the complete picture at a glance let’s have a look at the comparison chart. Remember again that Bitcoin and Ethereum are two blockchains while Hyperledger includes a lot of Blockchain development frameworks. So it will be a general comparison.

	Bitcoin	Ethereum	Hyperledger Frameworks
Cryptocurrency Based	Yes (Bitcoin)	Yes (Ether)	No
Permissioned	No	No	Yes (In general)
Auditable	Yes	Yes	Yes
Immutable Ledger	Yes	Yes	Yes
Modularity	No	No	Yes
Smart Contract	No	Yes	Yes
Consensus Protocol	PoW	PoW	Various
Major application domains	Cryptocurrency	DApps including cryptocurrency and more.	Enterprise blockchain development

Multichain

Multichain is a free and open source blockchain platform to create private/permissioned blockchain networks. Multichain is an extended version of the bitcoin core software. The bitcoin engine provides security and control over peer to peer communications to Multichain.

Language support

The thing which makes Multichain more powerful is its support to 5 high-level languages, they are Python, JavaScript, and Ruby, Php, and C #. Multichain provides a simple API and a command line interface for the application development. The developer can download Multichain packages for all those languages from Github repository and start development. Multichain doesn’t use cryptocurrencies and smart contracts. So that financial transactions are not possible with it.

Compared to other blockchain platforms average block time is very low in Multichain and it is about 2 seconds. But the speed increases the chance of hash collisions.

Security

The main feature of a Multichain is that the visibility of blockchain activities are kept private within the chosen participants. Only those selected participants can see the activities in the blockchain. It uses a set of collective admins, i.e.; a set of identifiable entities are defined as miners and all the validation/mining task will be done by them only. But the mining process doesn’t involve the proof-of-work (PoW) scheme. Many blockchain networks including bitcoin use PoW scheme, in which a is a piece of data is created to verify the transaction which

is difficult to produce but easy for others to verify
In Multichain, all the transactions between two participants are secured with a handshaking mechanism. In which a handshaking message and acknowledgment message is used to make sure that the exact participants are available on the communication channel.
Following steps will take place between two participants before starting a transaction

- Each node submits its identity as a public address in a permitted list.
- Each node will verify that the other nodes address is there in its permitted list
- Each node will send a challenge message to the other node.
- After receiving the challenge message each node will send back a signature of the challenge message providing their ownership of the private key corresponding to the public key.

If the node who sent the challenge message received a signature message then the transaction between those two participants will start. And if a satisfying signature is not received then the peer to peer connection between those two nodes will disconnect.

Mining

Multichain introduces a new parameter called 'mining diversity' for defining the mining process. The mining diversity is used for defining the participation of minors in the mining process. In Multichain, mining is done in a "Round-robin schedule". The minors can create valid blocks in a round robin fashion so that all the minors can participate in the mining process equally.

The mining diversity is defined as
"0<= Mining Diversity <=1"

The '1' represents that every permitted miner will participate in the round robin rotation and 'zero' represents no restrictions in mining. All miners can equally participate in the mining process and the validity of the block can be verified through different steps.
In a Multichain platform, there is no transaction cost or block rewards by

default. But we can define those parameters in the params.dat file. The params.dat file contains several parameters for defining the blockchain behavior. There is an agricultural supply chain application already available on the Multichain platform. The app can control the entire supply chain system starting from the farmer to the customer. Each stage can be tracked through blockchain and the approach will help to increase the quality, reach and profitability of the product.

HydraChain

HydraChain is an open source blockchain platform, which is developed by Brainbot technologies and Ethereum project. HydraChain is an extension of the Ethereum Blockchain platform which provides support to create private/permissioned Blockchain networks. The supporting language for HydraChain is python. As an extension of Ethereum, HydraChain is fully compatible with all the API level and contract level protocols in Ethereum. There are several well-defined tools in Ethereum for creating smart contracts and DApps, (Decentralized Apps). You can reuse all those tools in HydraChain also. So it will be easy for those who know Ethereum to move on to HydraChain.

Smart contracts and HydraChain

Solidity/Serpent based smart contracts can co-exist in the same chain with the Python based smart contracts. Yes...! They are interoperable. Smart contracts created using HydraChain is independent of EVM (Ethereum virtual machine) as it is developed in Python programming language. The EVM provides a runtime environment for the contracts. EVM will execute all the untrusted codes and can provide security by restricting the accessing of each other's state. But Python based smart contracts will bypass the EVM so that the contract execution is fast. And we know, python is an easy to use language, less time consuming and easier to debug too. In fact, you don't need to go for a new language like solidity for developmental purpose.

How blocks are added?

Basically, HydraChain is providing the permissioned network creation services. So the validation is a great concern here. There will be a registered accountable validators in the network who is responsible for the validation of the blocks and transactions. In a HydraChain network, all the blocks are not allowed to enter the network without validation. That means a block will be added to the network only when the validators sign that the block is required. So once a block is entered into a network it is persistent. There are no reverts.

The HydraChain keeps a limitation in creating the blocks. HydraChain will create a new block if and only if there is a pending transaction. Whenever the Blockchain is unable to hold a transaction then it will create new block and validator has to sign it as a required block. The block will be added to the network after validation. As the validators are registered, a KYC is used for the participants, to make sure that the transaction are take place between registered participants only.The HydraChain platform provides a customizable nature in different components of Blockchain like transaction cost, gas limits, genesis allocation and block time. All these components have an inevitable role in a blockchain.

Transaction cost

The transaction cost in a HydraChain is the cost required for executing the computational steps in a transaction. In the HydraChain, you have the provision to configure the cost as per your requirement. The transaction cost can be calculated using an equation i.e.;

Transaction cost = Gas unit * Gas price.

Gas limits

The gas units are the basic units for an executed transaction in a Blockchain. A transaction can be divided into several ‘opcode’ and each opcode will have a specific number of gas units based on the type of the opcode.A ‘zerostep’ opcode has ‘0’ gas units, ’quickstep’ opcode has 2 gas units and a ‘faststep’ opcode has 5 gas units and so on. When a transaction is executing in a blockchain first it will extract the opcodes from the transaction, then the number of gas units will be identified from it, now the gas price will multiply with the total gas units, and that will be the transaction cost for that particular transaction. In HydraChain you can customize the gas limit also. Where the gas limits are the maximum number of computational steps in a transaction. For example: If a blockchain has a gas limit 50 then each transaction should have maximum 50 gas units.

Genesis Allocation in HydraChain

The genesis allocation is related to the hashing and mining capacity of a blockchain. The HydraChain has the power to customize its genesis allocation.

Usually, a Blockchain provides two types of mining methods they are direct mining and indirect mining. In Ethereum you will be using the cryptocurrency ‘ether’ for financial transactions. While creating a HydraChain you have the provision to decide whether to perform direct mining or indirect mining. Cryptocurrencies like ETH and ETC can be mined directly. And for other currencies like bitcoin, zeta coin etc. you can define an indirect mining with the help of sha256 hashing algorithm. In indirect mining, first you have to mine the Ethereum anyway, at payout time the ETH will exchange with other currencies of your choice at the latest exchange rate. This customization feature in mining will provide a facility to do financial transaction with all currencies.

Block time

One more customization is possible in HydraChain that is block time. Block time is the time delay between the validations of two blocks. An average block time in a Bitcoin Blockchain is 10 minutes that means it will take 10 minutes for the addition of a new block to a Blockchain.

Installation

HydraChain can be downloaded and installed from GitHub. It also provides easy deployment of networks. For the network deployment, several docker file templates are already available in HydraChain. These templates can be used while we creating new networks.

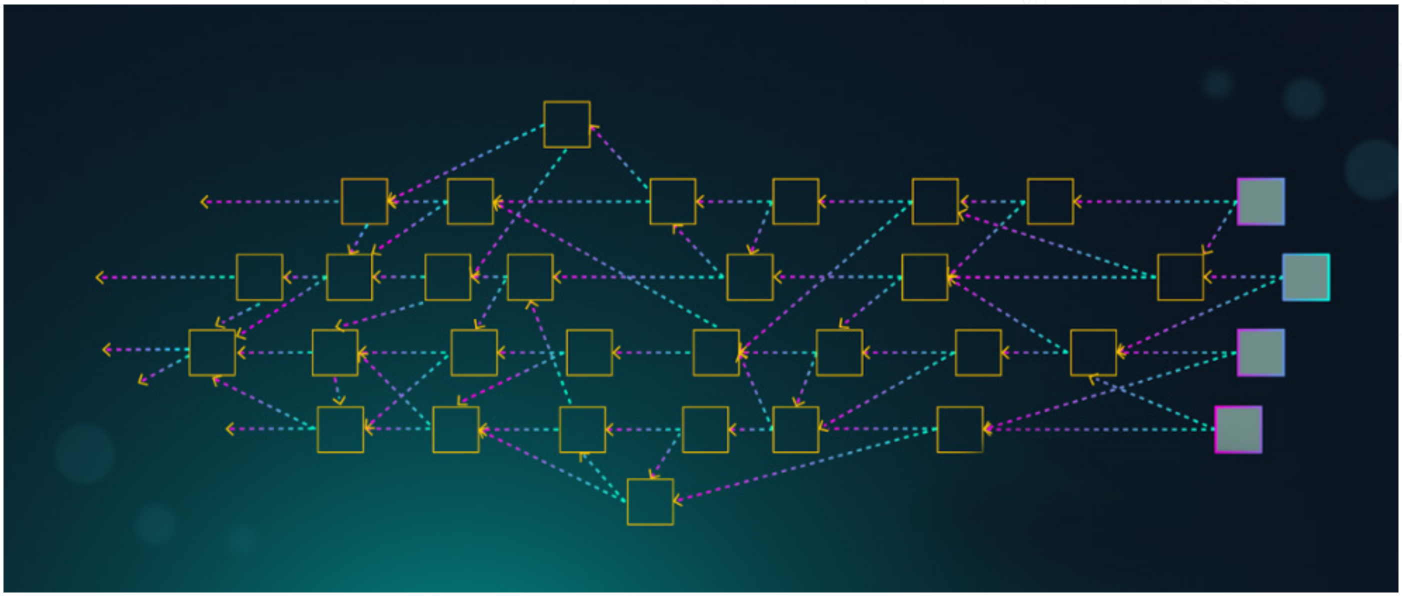
IOTA

IoT (Internet of Things) has made immense progress from conceptual to deliverable aspect in the last couple of years. We started our digital era with sharing of files only, now it has turned into a stage where we can share anything as a digital product. From wearable gadgets to vehicles to home appliances, the objects that are connected to the internet are increasing exponentially. In the beginning, we were unable to share these physical entities through the internet now we are in the 4th industrial revolution where we can transfer anything in the world through internet.We know IoT is the network of physical devices like gadgets and home appliances, vehicles etc., But what is IOTA?

In simple words, IOTA is blockchain technology which enables the digital transaction of IoT products across the network. The Blockchain technology is known as the future internet where IOTA (Internet of Things tAngle) is known as future Blockchain. Yes!!! These things are making the real technological revolution in the industry.

IOTA- a ‘Block’ less Blockchain

IOTA provides a Blockchain for the Internet of Things. But the interesting thing is that there are no blocks in IOTA. Then how Blockchain is created? Then answer lays in another concept known as DAG (Directed Acyclic Graph) which is a directed graph without cycles. In IOTA there is no ledger as in normal blockchains instead they use a DAG called Tangle for the transaction management.



In Tangle, the vertices of the graph represent the nodes/physical devices and the directed edges represent the transaction from one device to another. In short, IOTA network is a lightweight tangle which is scalable to any extent for adding any number of transactions and DAG is the backbone of it.

No Transaction Fees?

In most Blockchain networks the transaction cost is often a matter of concern. But IOTA is not charging any fee for the transaction. The IOTA is mainly designed to perform Nano transactions and these Nano transactions will be executed without any transaction fee. The IOTA can create both private and permissioned networks of IoT, and can manage the transactions with Tangle.

No miners?

In case of a Blockchain mining is a vital element, but IOTA is an exception to it. There is no miners or mining process in IOTA network. So how the transactions are verified in IOTA? The tangle just needs a verification only. And the verification is done by the node who generated the transaction, with the help of a validation algorithm. But the node will be able to proceed this transaction only after verifying two other random transactions in the network using the same validation algorithm.Instead of the Bitcoin protocol, IOTA uses the GHOST(Greedy Heaviest Observed Subtree) protocol, which is a modified version of bitcoin protocol itself. GHOST just modify the bitcoin protocol by creating a tree instead of a blockchain.

Weighted graphs

In IOTA, the priority of the transaction is measured with the help of 'weight' associated with each transaction. DAG is a weighted graph and the weight of each transaction is proportional to the amount of work that the issuing node invested into it. And obviously, the transactions with higher weight will get higher priority than the transactions with lower weight

Developing IOTA

The supporting languages for IOTA are Python and JavaScript. The library packages for the languages are available on the IOTA website itself. The developer may use an IOTA sandbox environment or install IOTA core client for developing the IOTA network. In IOTA network, every object is considered as a service. Or in other words, the physical existence of a 'thing' will be converted as a 'service' in the tangle. This conversion is done with the help of IoT sensors. An IoT sensor is nothing but a simple hardware device attached to the physical entity, which will detect and digitize all the movements of that particular entity. The digitized data will be used in our IOTA network for the management of these entities. All machine to machine communication are controlled using this IoT sensors in IOTA network.

Security

IOTA offers a high level of security for both transactions and assets. The data transfer through the tangle will be in encrypted form and fully protected from external attacks. IOTA uses the masked massaging technique ensure the security of data transfer. In Masked messaging service, the data is encrypted with quantum proof security which makes the data broadcasting also easy. Starting from the weight calculation to restricting an external attack, IOTA employs several mathematical equations which are capable of detecting any small changes in the graph. This highly mathematical approach ensures the protection of data from any kind of external attacks.

The combined advantage of blockchain and IoT has already brought many application areas to IOTA. As the IoT and blockchain is expanding rapidly, more existing services may come under this technology in near future.

Corda

Corda is a distributed ledger platform specially designed for the financial sector. It is an open source platform that can be used to build apps for financial institutions on top of it. It is a permissioned private network designed to record, manage and synchronize contracts and other shared data between partners. Corda is governed by R3 consortium which is a collaboration of 70+ finance institutions. According to R3, Corda is a distributed ledger technology and isn't a blockchain. In fact, R3 provides a platform for developing and deploying distributed apps for different financial use cases. The distributed apps created with Corda is known as CorDapps. DemoBench is a standalone desktop application provided by Corda to configure and launch local Corda nodes. It is a useful tool for training sessions and development of CorDapps.

Corda has many similarities as well as differences with many existing blockchain/distributed ledger technologies. Corda allows the creation of immutable records for financial events. But unlike other blockchains, the transactions are done privately in Corda. Corda smart contracts can be written in Java or any other JVM language like kotlin (a java derived language).

And most importantly, Corda is not tied to any particular consensus algorithm and it doesn't have its own cryptocurrency. It uses the "Notary" infrastructure for 'sequencing of transactions' and validating the transactions. And it does not broadcast a transaction globally for validation purpose. A Corda network may have multiple 'Notaries' and they validate the transactions using different algorithms. The ultimate objective of Corda is to remove costly friction in business transactions by avoiding businesses intermediaries. Since it is only focusing on finance domain, its architecture is simple than that of Ethereum or Fabric. This approach gives performance and security advantage for Corda over other enterprise-level blockchain frameworks. Just like many other distributed technologies, Corda is also in its infant stage and it is hard to make a conclusion on its prospects.

Elements Project

The Elements Project is a protocol level technology which is used to extend the functionalities of bitcoin blockchain. It is an open source community driven project which intends to create new extensions to the Bitcoin and build more bitcoin-based applications.Elements project uses sidechain technology to easily integrate the new application to bitcoin blockchain. Side chains are the separate blockchains having all the components of a normal blockchain including the smart-contract. It exists along with the main blockchain with a back and forth transaction compatibility.

Some of the deployed Elements.

The community members has developed and deployed many Elements already and some of them are on the way of deployment. The well-known side chains of bitcoin including Alpha, Gem are the product of elements project. Some other promising elements are listed here,

1. Asset Issuance

Physical assets are mostly being converted to digital assets nowadays. Since asset has high value than any other documents it needs more protection and security. Asset Issuance helps to issue assets asset including deposits vouchers, shares, currencies, deposits, coupons bonds, etc. with better security.

2. Confidential Transactions

It is a highly sought extension of the bitcoin blockchain. Confidential Transactions hide the amount of bitcoin transferred between two parties from a third party.

3. Segregated Witness

Using Segregated Witness we can reduce the space that used by every transaction in the block. The application separates the properties of a transaction that reflects on the ledger from the data that needed for the validation. This reduces the memory required for a block to store a transaction detail.

4. Relative Lock Time

It allows setting a timeout for the transactions. You can set a particular time period to complete the transaction if it is not completed within the time the transaction will be canceled. It can augment the transaction security of bitcoin blockchain.

5. Schnorr Signature Validation

This extension provides a new way for making signatures for validation and providing new methodologies for multi-signatures

6. New Opcodes

It introduces new opcodes to bitcoin includes DETERMINISTICRANDOM and CHECKSIGFORMSTACK. Furthermore, it re-enabled several scripts in side-chain that are previously available in Bitcoin blockchain.

7. Signature Covers Value

The value can be used for validating the transaction fast. Since the signature on a transaction will be invalidated if the inputs have been spent. This will help you to validate the transaction very easily.

8. Deterministic Pegs

This extension offers a cross-chain facility which simplifies the token transfer between two blockchains.

9. Signed Blocks

The Signed Block is a useful extension to bitcoin. This will allow the user to sign the blocks cryptographically. It is helpful to verify the creator of a block in future.

Other than these listed Elements there are many other extensions available under elements project. More elements are being created by community participant. The project has helped the bitcoin blockchain to work with more customized features and augmented its application domains.

Chain

Chain Core

Chain Core is a blockchain management software developed by Chain Inc. in 2014. The software is designed to manage the permissioned blockchain networks. The chain core can manage any number of independent blockchains or it acts as a blockchain client for different permissioned blockchains. Chain core keeps the copy of the ledgers of multiple blockchains and updates these ledgers during the validation of transactions. The validation and consistency in Chain core are ensured by a Federation of block signers. Here any digital assets including digital currencies, securities, bonds etc. are issued in a common format and represented using any units of value guaranteed by the trusted issuer

There are two editions of Chain core available. A Free Open Source Developer edition and an Enterprise Edition. The Developer edition can be used to test and make prototypes. The Enterprise Edition is essential to develop and deploy the original product based on this prototype.

The leading financial service firms like Visa, Citi group etc. are working with Chain core to develop their blockchain infrastructure.

There are basically three operations available in the software.

1. Create a blockchain.

This option is for creating a new blockchain. The chain core act as a block generator as well as a block signer in the created network. The core provides a Url and a blockchain id for the created network. The id and Url are useful when another core is going to join this network.

2. Connect to an existing Blockchain network

This option enables a core to connect to an existing network. A user must have a blockchain url, a blockchain id, and an active access token for managing the transactions and digital assets.

3. Connect to the test blockchain network

This option is basically oriented to beginners. They may join the blockchain network of chain core and test the blockchain network by making basic operations like account creation, transaction, digital asset management etc.

Development & Security

The chain core application can be developed with Java, node.js, or Ruby. The respective packages and APIs are available in respective repositories. Chain core uses HSM (Hardware Security Module) for a production environment. Compared to other platforms this approach provides a better security standard for the digital assets.

Chain core uses private & public key pairs for the locking and unlocking of assets. Assets are always loaded with a control program. The transactions are verified by running these control programs along the data (public key). If it produces a valid result then the transaction is declared as valid. Using multiple keys for transactions will improve the level of security.

Ivy and Ivy Playground

Ivy is the high-level programming language developed by Chain for creating smart contracts in Chain core. The Ivy playground is an additional tool to create, compile and load the smart contract that can be run along the core.

As the number of blockchains is being created for different purposes, the importance of a tool like the core is evident. The security features like HSM and simplicity in blockchain management makes core an appropriate option for blockchain management

CoCo Framework

Coco (Confidential Consortium) is an open source blockchain framework designed by Microsoft. Microsoft announced the ‘Coco’ in August 2017 in their whitepaper ‘Coco Framework Technical Overview’. The source code of the Coco framework is planned to publish in Github by 2018. Coco is not just a standalone blockchain protocol like Bitcoin or Ethereum rather it provides a platform for building trusted networks using any of the existing protocols. Of course, Coco is designed to be compatible with any existing blockchain protocols such as Ethereum.

Specialties of CoCo

In their whitepaper, Microsoft points out some of the problems with existing systems and how Coco solve the issues.The main drawbacks they pointout of existing systems are

Low transaction throughput

The average processing rate of the public Ethereum network is only 20 transactions per second, which is far behind to meet the requirement in an enterprise environment. Other blockchain networks also fail to meet the enterprise level transaction rate.

High latency

The average latency of a public Ethereum network is about 10-20 seconds and it is 10-15 minutes in bitcoin network. Such high latency will create a bottleneck effect in a business environment.

Lack of confidentiality

In a public blockchain networks, everyone is allowed to see every transaction. This is definitely not a welcome thing in the business environment where the competitors may also be the part of the network.

Lack of effective governance

Public blockchain networks are often self-governed or collectively governed by the users. The model is not suitable for many environments, especially for business level networks.

Low computational efficiency

As the network grows, the computational power required for mining also grows. Thus the energy required is very huge. The annual energy consumption of the bitcoin network is about 15 TWh !!!.

Many attempts were made to overcome these issues and new blockchain platforms like Fabric, Corda etc. also came into existence. But some of these are designed only to meet the requirements of a particular business domain. Some others provided an enterprise level control and security by employing complex algorithms but compromises on performance.Furthermore, whenever a new protocol is introduced to accommodate a feature, the user has to leave behind the technology that he expertise. And it will take some time to understand and work with the new system.

Benefits of CoCo

According to Microsoft, the COCO framework eliminates most of the drawbacks of the existing systems and it offer

- Acceptable throughput and latency for meeting enterprise needs
- Richer, flexible yet simpler confidentiality models
- Network policy management and distributed governance
- Facilitate non-deterministic transactions
- Reduced energy consumption

Coco achieves these performance indices through the use of Trusted Execution Environments (TEEs) like Intel's SGX or Windows Virtual Secure Mode (VSM).

This approach enables Coco to create a trusted networks of nodes and the distributed ledgers are run top of these.

The introduction of Microsoft coco framework is expected to make a big leap. As said earlier, the Coco is not a standalone blockchain protocol. Actually, it provides a foundation for building blockchain networks on top of it. Thus with Coco, it is possible to develop blockchains in any protocol and can integrate different blockchain technologies into a single project to satisfy different enterprise needs. And coco provides many additional features to ease and enhance the development process. In conclusion, from the information available so far, Coco has the potential to be the cradle of blockchain based enterprise applications

TIERION

Tierion

The importance and vicinity of Blockchain are increasing on daily basis. More existing platforms and services are shifting towards the Blockchain technology by perceiving the advancement it makes. Consequently, different tools and associated services are also emerging in the background. Tierion is such an associated platform which can be used to create a verifiable database of any data on Blockchain. Or it is a Proof engine for data verification. Developers use Tierion to check integrity and timestamp of data or file or any process. The platform offers API and Developer tools to anchor data into a distributed ledger.

The capabilities of Tierion can be utilized by financial institutions, Insurance firms, etc. for safeguarding their critical data from unauthorized modifications. With Tierion, they can track each and every modification being made to the property titles, contracts, digital assets etc. Chain point, an open source protocol and distributed service developed to anchor data into the Bitcoin and Ethereum Blockchain, is the backbone of Tierion. The company is presently working with the Blockchain development projects of Philips and Microsoft to expand the application of it to ore areas.

Features of Tierion

Following are some of the features of Tierion which makes it an advanced tool for data verification.

Digital Receipts: The digital receipts issued by Tierion is a timestamp proof of a transaction took place.

Audit Trail: Tierion generates audit trials for data which are cryptographically verifiable. The trial will track a data from the origin onwards.

Immutable Records: Properly tracked data guarantees the immutable record keeping.

Secure Customer Data: They create verifiable customer data and reduce KYC and compliance cost.

Hash API: With Tierion 'Hash API' developers can anchor records with minimum cost.

Data Collection: Tierion is also used collect data from the web and mobile applications.

Integrate with other Apps: Zapier helps Tierion to integrate with other apps such as Gmail, Twitter, SalesForce etc.

Chainpoint

It contains all the information needed to verify the data without intermediaries. Chainpoint is the main component of Tierion which creates the timestamp proof of a Blockchain transaction. The initial version of Chainpoint was introduced in June 2015, and later versions Chainpoint 2.0 and 3.0 released in August 2016 and August 17 respectively. The ultimate proof from a Chainpoint or a 'chainproof' is a trail of operation cryptographically linking your data to one or more Blockchain Chainpoint Proof Creation Steps

Following steps are involved in Chainpoint proof creation.

- The user submits the hashed data to a Chainpoint.
- Chainpoint returns a hash_id (UUID) with a timestamp to the user.
- Chainpoint combine the submitted hash with UUID to obtain a new hash.
- The same hash is combined with a 'NIST Beacon' and a new hash is

created.
(This will make sure that the chain proof is created after the generation of hash_id)

- The new hash is sent to aggregation service.
- Aggregation service aggregates the hashes into Merkle trees.
- Then the Merkle root of the Merkle tree is sent to the Chainpoint calendar. (Various Chainpoint servers are kept in agreement to create a Chainpoint calendar. In fact, Chainpoint calendar is a Blockchain.)
- Calendar data is organized as blocks, and they are stored in a normal database called CockroachDB.Calendar blocks are then anchored to Bitcoin or Ethereum Blockchain.
- Now the Chainpoint starts to monitors the Blockchain. On each anchoring, if the transaction receives an adequate number of 'Validation', validated blocks are added to the calendar.
- Each validated blocks contains data to create the final Chainpoint proof.
- To finalize the proof, Chainpoint appends the partial proof with final data. And the final proof is created.

Benefits of Tierion

The major benefit of Tierion is that it eliminates the role of the third party in data verification. Anyone with the proof issued by Tierion can verify the entire transaction path of a data. It has a highly scalable architecture and better performance standard. The time stamp accuracy is achieved with Network Time Protocol (NTP) and National Institute of Standard and Technology (NIST) server. Immediate anchoring is another feature, Chainpoint anchors the data whenever a new hash is submitted to Chainpoint service. And indeed it is a cost-effective solution.

BIGCHAIN DB

BigchainDB

The BigchainDB is a scalable distributed database which can be used for the blockchain technology. In a normal case, blockchain itself is the database. As in the case of bitcoin and many blockchain applications the blocks is providing the storage facility too. There are no additional databases. But the BigchainDB provides an alternative to this method. The BigchainDB will work as a distributed database with all characteristics of a blockchain

The BigchainDB was first introduced as a distributed database and later the characteristic of blockchain technology has added to it. Now BigchainDB has the features of both traditional blockchain (like bitcoin) and the distributed database and it supports both private and public networks. BigchainDB is a NoSQL(Non-SQL) database which provides a storage mechanism and data retrieval models other than the tabular relations used in relational databases.

The commonly used NoSQL types are Key-value stores, Document database, Wide column stores and Graph stores. Each of these NoSQL databases adopts different methods of data storage. The developer can select any of the above models according to the requirement and use case.

Why BigchainDB?

Normal blockchain networks like bitcoin suffer from several problems like low throughput, high latency, low storage capacity etc. In a bitcoin network, the latency before a single confirmed write is about 10 minutes and throughput is only a few transactions per second. The storage capacity is also not promising as it is still pegging at a few dozen GB. But in BigchainDB the throughput is about 1 million writes per second and latency is also significantly lower. The storage capacity of BigchainDB is that of a distributed database. Which means the capacity will increase as the number of nodes increases.

The BigchainDB has the following features.

- Decentralized control:-No central server for managing the database
- Immutability:-Once a change is made to the database it is immutable.
- Creation & Movement of Digital assets:- Digital assets can be created or manage the BigchainDB

BigchainDB vs Normal Blockchain and Distributed Database

	Traditional Blockchain	**Distributed Database**	**BigchainDB**
Throughput	Low (few transaction per second)	High (increase with nodes)	High (increase with nodes)
Latency	High (10 min)	Low	Low
Storage capacity	Low	High (increase with nodes)	High (increase with nodes)
Query capability	No	Yes	Yes
Rich permissioning	No	Yes	Yes
Decentralized Control	Yes	No	Yes
Immutability	Yes	No	Yes
Creation & Movement of Digital assets	Yes	No	Yes
Event Chain Structure	Merkle tree	-	Hash Chain

Models in BigchainDB

Three models namely Transaction model, Block model, and Vote model are the backbone of BigchainDB. These models give it the advantages of Blockchain as well as the normal database.

Transaction Models in BigchainDB

The basic component of BigchainDB is transactions. Every data stored in it will the details of the individual transaction. Two types of transaction models are used in BigchainDB

1) Creation Transaction
2) Transfer Transaction

The “Creation Transaction” is used to initialize the details of an asset in the blockchain and the “Transfer Transaction” is used to transfer ownership of the asset. A transaction in a JSON document will have the following structure
Id: Is the primary key. It will be the hash value of that particular transaction,
Version: It is the version number of that transaction model,
Fulfillments: Each fulfillment is a pointer to the unspent assets. It will point to the ownership of an asset,
Conditions: List of conditions that should be fulfilled by the transfer transactions,
Operation: String representation of the operation to be performed,
Timestamp: Transaction creation time in UTC. Provided by the user,
Hash: It is the hash value of the serialized payload,
Payload: It can be any JSON document. For a transfer transaction, it will be empty. All the transactions in the BigchainDB will be stored in the above mentioned structure only.

Block Models in BigchainDB

The blocks are also represented as JSON documents in the following structure,

```
{
“id”: “<hash of block >”,
“block”: {
“timestamp”: “<block -creation
timestamp >”,
“transactions”: [“<list of
transactions >”],
“node_pubkey”: “<public key of
the node creating the block >”,
“voters”: [“<list of federation
nodes public keys >”]
},
“signature”: “<signature of block >”,
“votes”: [“<list of votes >”]
}
```

Id: The primary key. It is the hash of the serialized block,

Timestamp: It is the time of creation of a block. It is given by the created the node,

Transactions: The list of transactions included in the block,

Node-pub key: The public key of the node, that created the block,

Voters: It is the list of public key of federation nodes existed in the system when the node is created,

Signature: Signature of the block by the node who created the block, It is generated by serializing the block data and using the private key,

Votes: list of votes given by the voters.

A vote has the following structure:

```
{
“node_pubkey”: “<the public key of the voting node
>”,
“vote”: {
“voting_for_block”: “<id of the
block the node is voting for >”,

    “previous_block”: “<id of the
block previous to this one >”,
“is_block_valid”: “<true|false
>”,
“invalid_reason”:
        "<None|DOUBLE_SPEND|TRANSACTIONS_HASH_MISMATCH|
        NODES_PUBKEYS_MISMATCH",
        "timestamp": "<timestamp of the
        voting action >"
        },
        "signature": "<signature of vote >"
        }
```

Node_pubkey: It is the public key of the voting node.

Voting_for_block: Id of the block for which a node is voting

Previous_block: Id of the previous block

Is_block_valid: Vote for the block it can be true or false. I.e. positive or negative vote

Invalid_reason: Reason for invalidating or voting ’false’.

Timestamp: Time at which voting action takes place.

Signature: Signature for the vote.

Among many other blockchain related technologies BigchainDB is a unique one as it changes the very data storage mechanism of the blockchain. From the initial assessment, it is a promising technology, especially for handling the huge amount of data. It has the potential to leverage the blockchain technology in the domains like Big data analysis AI etc.

Conclusion

The blockchain technology has undergone many changes since its onset in 2009. From a mere cryptocurrency image, it spread to a lot of other unexpected domains. As we mentioned earlier, entrepreneurs are venturing to the new use cases of blockchain and are coming with assuring results. After the first blockchain-Bitcoin, many new protocols have emerged; Ethereum is prominent among it. Today, high-level developments are taking place in Ethereum. DApps, ICO, and Tokens are the major area where Ethereum blockchain is highly utilized. Along with it, Hyperledger project is trying to give full-fledged blockchain protocol suits for enterprise level applications, and so far they have made successful results. The pace of developments taking place in blockchain is also faster than many existing technologies. New protocol suites, development IDEs, implementation environment, blockchain management suits and allied ecosystem are introduced frequently. Hopefully, most of these developments are in a positive direction and they rectify the problems in existing frameworks.

On a wider perspective, the blockchain may bring the next level of evolution on the course of the human race by radically changing the socio-economic and political establishments. Yet some others say it is too early to make such auspicious hopes as the technology is still in its infancy. But it is sure that if we could develop the technology as we envisaged at the beginning, the radical changes are quite certain.

As we mentioned in the beginning, this book is just a beginners guide, however, we included as much as we can to clarify the misconception exists among many of us. So read further to be updated with blockchain and its developments.

CYBROSYS™
Technologies

BLOCKCHAIN®
EXPERT

BLOCKCHAIN