R.M.K
GROUP OF ENGINEERING INSTITUTIONS

R.M.K
GROUP OF INSTITUTIONS

R.M.K
GROUP OF
INSTITUTIONS

Please read this disclaimer before proceeding:

R.M.K
GROUP OF
INSTITUTIONS

22AI001 AI in BLOCK CHAIN

Department : **CSE(Cyber Security)**

Batch / Year : **2022 – 2026 / IV**

Created by : **Mr. RAJESH KUMAR K, AP/CSE(CS)**

Date :

Table of Contents

Course Objectives

Course Objectives

		L	T	P	C
22AI001	**AI IN BLOCK CHAIN**	**3**	**0**	**0**	**3**

OBJECTIVES:

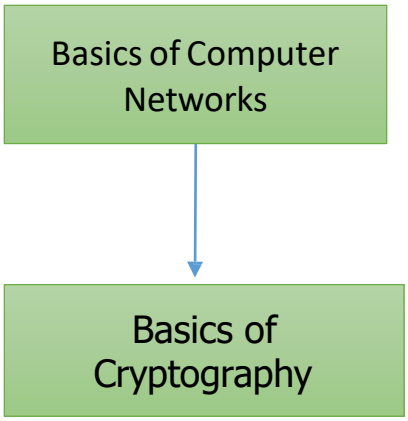
The Course will enable learners to:

- To acquire knowledge in Blockchain Technologies
- To Understand how block chain and AI can be used to innovate.
- To explain Cryptocurrencies and AI.
- To develop applications using blockchain.
- To understand the limitations and future scope of AI in Blockchain

.

PRE REQUISITES

Prerequisites



Syllabus

OPEN ELECTIVE (Offered to Other Departments by ADS)

22AI001	AI in BLOCK CHAIN	L	T	P	C
		3	0	0	3

OBJECTIVES:

- To acquire knowledge in Blockchain Technologies.
- To understand how block chain and AI can be used to innovate.
- To elaborate Cryptocurrencies and AI.
- To develop applications using blockchain.
- To understand the limitations and future scope of AI in Blockchain.

UNIT I	INTRODUCTION TO BLOCKCHAIN	9

Overview – Blockchain vs Distributed Ledger Technology vs Distributed Databases – Public vs private vs permissioned blockchains – Privacy in blockchains – Blockchain platforms - Hyperledger – Hashgraph, Corda – IOTA - Consensus Algorithms – Building DApps with blockchain tools.

UNIT II	BLOCKCHAIN AND ARTIFICIAL INTELLIGENCE	9

Introduction to the AI landscape - AI and Blockchain driven Databases – Centralized vs Distributed data – Blockchain data – Big data for AI analysis – Global databases – Data Management in a DAO - Benefits of combining blockchain and AI – Aicumen Technologies -Combining blockchain and AI to humanize digital interactions.

UNIT III	CRYPTOCURRENCY AND AI	9

Bitcoins – Ethereum - Role of AI in cryptocurrency – cryptocurrency trading – Making price predictions with AI – Market making – future of cryptocurrencies.

UNIT IV	DEVELOPING BLOCKCHAIN PRODUCTS	9

Development Life Cycle of a DIApp – Designing a DIApp – Developing a DIApp – Testing – Deploying – Monitoring – Implementing DIApps.

UNIT V	LIMITATIONS AND FUTURE OF AI WITH BLOCKCHAIN	9

Technical Challenges – Business Model Challenges – Scandals and Public perception – Government Regulation – Privacy Challenges for Personal Records – Convergence of AI with Blockchain – Future – Enterprise.

TOTAL: 45 PERIODS

OUTCOMES:

At the end of this course, the students will be able to:

CO1: Acquire knowledge in Blockchain Technologies.
CO2: Understand how block chain and AI can be used to innovate.
CO3: Elaborate Cryptocurrencies and AI.
CO4: Develop applications using blockchain.
CO5: Understand the limitations and future scope of AI in Blockchain.
CO6: Elaborate the various applications of AI in Blockchain.

TEXT BOOKS:

1. Ganesh Prasad Kumble, Anantha Krishnan, "Practical Artificial Intelligence and Blockchain: A guide to converging blockchain and AI to build smart applications for new economies", Packt Publications, 2020.
2. Melanie Swan, "Block Chain: Blueprint for a New Economy", O'Reilly, 2015.

REFERENCES:

1. Daniel Drescher, "Block Chain Basics", Apress; 1st edition, 2017.

Course Outcomes

Course Outcomes

CO#	COs	K Level
CO1	Acquire knowledge in Blockchain Technologies	K1
CO2	Understand how block chain and AI can be used to innovate.	K2
CO3	Explain Cryptocurrencies and AI.	K2
CO4	Develop applications using blockchain.	K4
CO5	Understand the limitations and future scope of AI in Blockchain.	K4

Knowledge Level	Description
K6	Evaluation
K5	Synthesis
K4	Analysis
K3	Application
K2	Comprehension
K1	Knowledge

CO – PO/PSO Mapping

R.M.K
GROUP OF
INSTITUTIONS

CO – PO /PSO Mapping Matrix

CO #	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PS0 3
CO1	3	3	2	1	1	-	-	-	2	-	-	2	3	2	-
CO2	3	2	2	2	2	-	-	-	2	-	-	2	3	2	-
CO3	3	3	2	2	2	-	-	-	2	-	-	2	3	2	-
CO4	3	2	2	2	2	-	-	-	2	-	-	2	3	3	-
CO5	3	2	2	2	2	-	-	-	2	-	-	2	3	2	-
CO6	2	2	1	1	1	-	-	-	2	-	-	2	3	2	-

Lecture Plan

Unit I

R.M.K
GROUP OF
INSTITUTIONS

Lecture Plan – Unit 1 -Introduction

Sl. No.	Topic	Number of Periods	Proposed Date	Actual Lecture Date	CO	Taxonomy Level	Mode of Delivery
1	Blockchain Overview	1			CO1	K2	Chalk & Talk
2	Blockchain vs Distributed Ledger Technology vs Distributed Databases	1			CO1	K2	Chalk & Talk
3	Public vs private vs permissioned blockchains	1			CO1	K2	Chalk & Talk
4	Privacy in blockchains	1			CO1	K2	Chalk & Talk
5	Bitcoins	1			CO1	K3	Chalk & Talk
6	Ethereum	1			CO1	K3	Chalk & Talk
7	Hyperledger	1			CO1	K2	Chalk & Talk
8	Blockchain platforms	1			CO1	K3	Chalk & Talk
9	Consensus Algorithms – Building DApps with blockchain tools.l	1			CO1	K3	Chalk & Talk

Activity Based Learning

Unit I

R.M.K
GROUP OF
INSTITUTIONS

Activity Based Learning

CRYPTO WORD SEARCH

G V E K I D R I H O D L C K W I D
W U T B G C C Y K A C G V H F Y R
L B H Q G C O I N R R E C K O C I
X L E E I M U Z F P Y O Z E R Q P
K O R H C F B A V F P Z P L K B P
F C E N O Z V F E W T C O V Y I L
Y K U F L W X Y J F O S U F H T E
P C M B M I S B E A R I S H B C B
P H X K M A R K E T C A P S X O O
O A T O K E N K X M Q L O H Q I O
L I C B R U A L T C O I N V E N D
A N B I N I N D B N W F P P M P V
M Z M C W A L L E T Z U C L O M O
B K T O C M I N I N G D Z Y O H M
O B U L L I S H A M U B A I N V U
N A V C R Y P T O C U R R E N C Y
I D E C E N T R A L I Z E D Z B R

coin
ico
altcoin
bitcoin

token
bearish
fork
ethereum

decentralized
bullish
lambo
ripple

marketcap
wallet
moon
hodl

R.M.K GROUP OF INSTITUTIONS

Lecture Notes – Unit 1

R.M.K
GROUP OF
INSTITUTIONS

1.1 Overview of Blockchain

Software system is considered as a composition of layers

Layers of a Software System

The following two ways of partitioning a system are used throughout this book:

- Application vs. implementation
- Functional vs. nonfunctional aspects

Application vs. Implementation

- Mentally separating the user's needs from the technical internals of a system leads to a separation of the application layer from the implementation layer.
- Everything that belongs to the application layer is concerned with the user's needs.

Example: listening to music, taking photos, or booking hotel rooms.

Everything that belongs to the implementation layer is concerned with making these things happen. Elements of the implementation layer are technical by nature.

Example: converting digital information into acoustic signals, recognizing the color of a pixel in a digital camera, or sending messages over the Internet to a booking system).

Functional vs. Nonfunctional Aspects

Distinguishing between what a system does and how it does what it does leads to the separation of functional and nonfunctional aspects.

Examples of functional aspects are sending data over a network, playing music, taking photos, and manipulating individual pixels of a picture.

Examples of nonfunctional aspects are a beautiful graphical user interface, fast-running software, and an ability to keep user data private and save. Other important nonfunctional aspects of a system are security and integrity.

- Integrity means that a system behaves as intended, and it involves many aspects such as security and correctness.
- As a rule of thumb, one can say that functional aspects are similar to verbs, while nonfunctional aspects are similar to adverbs.
- Identifying functional and nonfunctional aspects as well as separating application and implementation layer can be done at the same time, which leads to a two-dimensional table. Table 1-1 illustrates the result of mentally layering a mobile phone in this way.

Functional aspects of the application layer are the most obvious elements of a system, because they serve obvious needs of the users. The nonfunctional aspects of the implementation layer are rarely seen as major elements of the system. They are typically taken for granted.

Table 1-1. Example of Mentally Layering a Mobile Phone

Table 1-1. Example of Mentally Layering a Mobile Phone

Layer	Functional Aspects	Nonfunctional Aspects
Application	Taking photos	The graphical user interface looks beautiful
	Making phone calls	Easy to use
	Sending e-mails	Messages are sent fast
	Browsing the Internet	
	Sending chat messages	
Implementation	Saving user data internally	Store data efficiently
	Making a connection to the nearest mobile connector	Saving energy
		Maintaining integrity
	Accessing pixels in the digital camera	Ensure user privacy

Integrity

Integrity is an important nonfunctional aspect of any software system. It has three major components

- **Data integrity:** The data used and maintained by the system are complete, correct, and free of contradictions.
- **Behavioral integrity:** The system behaves as intended and it is free of logical errors.
- **Security:** The system is able to restrict access to its data and functionality to authorized users only

Most software failures, such as losses of data, illogical behavior, or strangers accessing one's private data, are the result of violated system integrity

Software architecture and its relation to the blockchain

A Payment System

Let's apply the concept of layering to a payment system. Table 2-1 shows some of the user's needs as well as some of the nonfunctional aspects of both the application and the implementation layers

Table 2-1. Aspects and Layers of a Payment System

Layer	Functional Aspects	Nonfunctional Aspects
Application	Deposit money	The graphical user interface looks beautiful
	Withdraw money	Easy to use
	Transfer money	Transfer of money is done fast
	Monitor account balance	System has many participants
Implementation	?	Available 24 hours a day
		Fraud resistant
		Maintaining integrity
		Ensure user privacy

Two Types of Software Architecture

There are many ways to implement software systems.The two major architectural approaches for software systems are centralized and distributed.

In centralized software systems, the components are located around and connected with one central component. In contrast, the components of distributed systems form a network of connected components without having any central element of coordination or control.

Figure 2-1 depicts these two contrary architectures.

Table 2-1. Aspects and Layers of a Payment System

Layer	Functional Aspects	Nonfunctional Aspects
Application	Deposit money	The graphical user interface looks beautiful
	Withdraw money	Easy to use
	Transfer money	Transfer of money is done fast
	Monitor account balance	System has many participants
Implementation	?	Available 24 hours a day
		Fraud resistant
		Maintaining integrity
		Ensure user privacy

- The circles in the figure represent system components, also called nodes, and the lines represent connections between them.
- The important point is the existence of these two different ways of organizing software systems.
- On the left-hand side of Figure 2-1, a distributed architecture is illustrated where components are connected with one another without having a central element.
- It is important to see that none of the components is directly connected with all other components. However, all components are connected with one another at least indirectly.

- The right-hand side of Figure 2-1 illustrates a centralized architecture where each component is connected to one central component.
- The components are not connected with one another directly. They only have one direct connection to the central component.

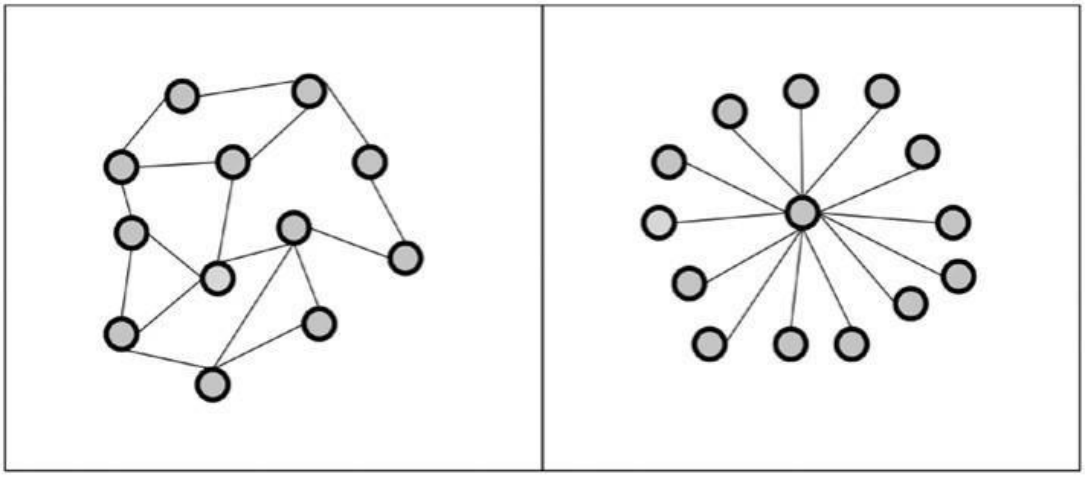


Figure 2-1. Distributed (left) vs. centralized (right) system architecture

The Advantages of Distributed Systems

The major advantages of a distributed system over single computers are:

- Higher computing power
- Cost reduction
- Higher reliability
- Ability to grow naturally

Higher Computing Power

The computing power of a distributed system is the result of combining the computing power of all connected computers. Hence, distributed systems typically have more computing power than each individual computer.

Cost Reduction

Since distributed systems consist of many computers, the initial costs of distributed systems are higher than the initial costs of individual computers.

However, the costs of creating, maintaining, and operating a super computer are still much higher than the costs of creating, maintaining, and operating a distributed system. This is particularly true since replacing individual computers of a distributed system can be done with no significant overall system impact.

Higher Reliability

The increased reliability of a distributed system is based on the fact that the whole network of computers can continue operating even when individual machines crash. A distributed system does not have a single point of failure. If one element fails, the remaining elements can take over. Hence, a single super computer typically has a lower reliability than a distributed system.

Ability to Grow Naturally

The computing power of a distributed system is the result of the aggregated computing power of its constituents. One can increase the computing power of the whole system by connecting additional computers with the system. As a result, the computing power of the whole system can be increased incrementally on a fine-grained scale.

The Disadvantages of Distributed Systems

The disadvantages of distributed systems compared to single computers are:

- Coordination overhead
- Communication overhead
- Dependency on networks
- Higher program complexity
- Security issues

Coordination Overhead

- Distributed systems do not have central entities that coordinate their members.
- Hence, the coordination must be done by the members of the system themselves.
- Coordinating work among coworkers in a distributed system is challenging and costs effort and computing power that cannot be spent on the genuine computing task, hence, the term coordination overhead.

Communication Overhead

- Coordination requires communication. Hence, the computers that form a distributed system have to communicate with one another.
- This requires the existence of a communication protocol and the sending, receiving, and processing of messages, hence, the term communication overhead.

Dependencies on Networks

- Any kind of communication requires a medium. The medium is responsible for transferring information between the entities communicating with one another.
- Computers in distributed systems communicate by means of messages passed through a network.
- Networks have their own challenges and adversities, which in turn impact the communication and coordination among computers that form a distributed system.

Higher Program Complexity

- Solving a computation problem involves writing programs and software.
- Due to the disadvantages mentioned previously, any software in a distributed system has to solve additional problems such as coordination, communication, and utilizing of networks.
- This increases the complexity of the software.

Security Issues

- Communication over a network means sending and sharing data that are critical for the genuine computing task.
- However, sending information through a network implies security concerns as untrustworthy entities may misuse the network in order to access and exploit information.
- Hence, any distributed system has to address security concerns.

Distributed Peer-to-Peer Systems

- Peer-to-peer networks are a special kind of distributed systems. They consist of individual computers (also called nodes), which make their computational resources (e.g., processing power, storage capacity, data or network bandwidth) directly available to all other members of the network without having any central point of coordination.
- The nodes in the network are equal concerning their rights and roles in the system.
- Furthermore, all of them are both suppliers and consumers of resources.
- Peer-to-peer systems have interesting applications such as file sharing, content distribution, and privacy protection.

Mixing Centralized and Distributed Systems

- The graphic on the left-hand side of Figure 2-2 illustrates an architecture that establishes a central component within a distributed system.
- On first glance, the components seem to form a distributed system. However, all of the circles are connected with the larger circle located in the middle.
- Hence, such a system only appears to be distributed on a superficial view, but it is a centralized system in reality.

- The graph on the right-hand side of Figure 2-2 illustrates the opposite approach.
- Such a system appears to be a centralized system on first glance, because all the circles in the periphery only have one direct connection to a large central component.
- However, the central component contains a distributed system inside. The components in the periphery may not even be aware of the distributed system that lives within the central component.

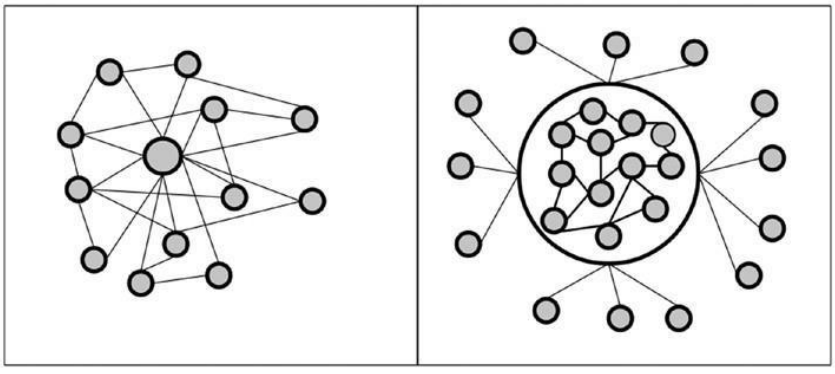


Figure 2-2. Mixing distributed with centralized architecture

Identifying Distributed Systems

The emergence of hybrid architectures makes it hard to identify distributed systems clearly. If you are in doubt whether or not a system is distributed, look for a single component (e.g., a database, a name or user registry, a login or logoff component, or an emergency switch-off button) that could terminate the whole system. If you find such a component, the system under consideration is not distributed.

Note: If one single component exists, e.g., a single switch-off button that can bring down the whole system, then the system is not distributed.

The Purpose of the Blockchain

The architecture is only a means to an end when it comes to implementing a system.

Hence, a payment system, as was proposed in Table 2-1, can be implemented as a distributed or centralized system.

Each of the two architectural concepts has its own advantages and disadvantages and their own specific way of doing things. Choosing a specific architecture has consequences on how you will achieve the functional and nonfunctional aspects of a system.

In particular, both architectural concepts have very different approaches to ensure integrity. And this is the point where the blockchain enters the picture. **The blockchain is a tool for achieving integrity in distributed software systems. Hence, it can be seen as a tool to achieve a nonfunctional aspect of the implementation layer.**

Note The purpose of the blockchain is to achieve and maintain integrity in distributed systems.

Peer-to-peer systems

- The definition of a peer-to-peer system
- Architecture of peer-to-peer systems
- The link between peer-to-peer systems and the blockchain

The Definition of a Peer-to-Peer System

Peer-to-peer systems are distributed software systems that consist of nodes (individual computers), which make their computational resources (e.g., processing power, storage capacity, or information distribution) directly available to another.

Architecture of Peer-to-Peer Systems

- Peer-to-peer systems are distributed computer systems by construction since they are made of individual nodes that share their computational resources among others.
- However, there are also peer-to-peer systems that still utilize elements of centralization.

Centralized peer-to-peer systems maintain central nodes to facilitate the interaction between peers, to maintain directories that describe the services offered by the peer nodes, or to perform look-ups and identification of the nodes.

Purely distributed peer to-peer systems do not have any element of central control or coordination. Hence, all nodes in those systems perform the same tasks, acting both as providers and consumers of resources and services.

An example of a centralized peer-to-peer system is Napster, which maintained a central database of all nodes connected with the system and the songs available on these nodes.

The Link Between Peer-to-Peer Systems and the Blockchain

Purely distributed peer-to-peer systems may use the blockchain in order to achieve and to maintain system integrity.

Hence, the link between purely distributed peer-to-peer systems and the blockchain is its usage for achieving and maintaining integrity in purely distributed systems.

The Potential of the Blockchain

- The relation between purely distributed peer-to-peer systems to the blockchain is that the former uses the latter as a tool to achieve and maintain integrity.
- Hence, the argument that explains the excitement about and the potential of the blockchain is: Purely distributed peer-to-peer systems have a huge commercial potential as they can replace centralized systems and change whole industries due to disintermediation.
- Since purely distributed peer-to-peer systems may use the blockchain for achieving and maintaining integrity, the blockchain becomes important as well.
- However, the major fact that excites people is the disintermediation. The blockchain is only a means to an end that helps to achieve that.

Trust and Integrity in Peer-to-Peer Systems

- Integrity is a nonfunctional aspect of a system to be safe, complete, consistent, correct, and free of corruption and errors.
- Trust is also the firm belief of humans in the reliability, truth, or ability of someone or something without evidence, proof, or investigation.
- Trust is given in advance and will increase or decline based on the results of interactions on an ongoing basis.
- With respect to peer-to-peer systems, this means that people will join and continue to contribute to a system if they trust it and if the results of interacting with the system on an ongoing basis confirm and reinforce their trust.
- Integrity of the system is needed in order to fulfill the expectations of the users and reinforce their trust in the system. If the trust of the users is not reinforced by the system due to a lack of integrity, the users will abandon the system, which, as a result, will eventually cause it to terminate. Achieving and maintaining integrity in purely distributed systems depends on a variety of factors, some of the most important are:
    - Knowledge about the number of nodes or peers
    - Knowledge about the trustworthiness of the peers

The chances of achieving integrity in a distributed peer-to-peer system are higher if the number of nodes as well as their trustworthiness is known. However, the worst circumstances for achieving integrity in a distributed peer-to-peer system are given when the number of nodes and their trustworthiness is unknown.

This is the case when running a purely distributed peer-to-peer system on the Internet that is open to everyone.

Two major integrity threats in peer-to-peer systems:

- Technical failures

• Malicious peers

Technical Failures

- Peer-to-peer systems are comprised of the individual computers of its users who communicate via a network.
- All hardware and software components of a computer system as well as any component of a computer network have the immanent risk of failing or creating errors.
- Hence, any distributed system has to face the problem that its components may fail or may produce wrong results by chance.

Malicious Peers

- Malicious members are the second integrity threat in peer-to-peer systems.
- This source of untrustworthiness is not a technical problem, but rather a problem caused by the goals of the individuals who decide to exploit the system for their own purposes.
- Dishonest and malicious peers comprise the most severe threat to the peer-to-peer system, because they attack the foundation on which any peer-to-peer system is built: trust.

The Core Problem to Be Solved by the Blockchain

The core problem to be solved by the blockchain is achieving and maintaining integrity in a purely distributed peer-to-peer system that consists of an unknown number of peers with unknown reliability and trustworthiness.

Definition of Blockchain

The blockchain, the term is used as follows:

• As a name for a data structure

• As a name for an algorithm

• As a name for a suite of technologies

• As an umbrella term for purely distributed peer-to-peer systems with a common application area

A Data Structure

- In computer science and software engineering, a data structure is a way to organize data regardless of their concrete informational content.
- When used as a name for a data structure, blockchain refers to data put together into units called blocks.
- These blocks are connected to one another like a chain, hence the name blockchain. The ordering is an important detail, which will be used extensively.
- Additionally, the chaining of the data blocks in the data structure is achieved by using a very special numbering system, which differs from the page numbering in ordinary books.

An Algorithm

- In software engineering, the term algorithm refers to a sequence of instructions to be completed by a computer.
- These instructions often involve data structures. When used as a name for an algorithm, blockchain refers to a sequence of instructions that negotiates the informational content of many blockchain-data-structures in a purely distributed peer-to-peer system.

A Suite of Technologies

- When used to refer to a suite of technologies, blockchain refers to a combination of the blockchain-data-structure, the blockchain-algorithm, as well as cryptographic and security technologies that combined can be used to achieve integrity in purely distributed peer-to-peer systems, regardless of the application goal.

An Umbrella Term for Purely Distributed Peer-to-Peer Systems with a Common Application Area

Blockchain can also be used as an umbrella term for purely distributed peerto-peer systems of ledgers that utilize the blockchain-technology-suite.

Various technical definitions of blockchains

Definition 1: The blockchain is a purely distributed peer-to-peer system of ledgers that utilizes a software unit that consist of an algorithm, which negotiates the informational content of ordered and connected blocks of data together with cryptographic and security technologies in order to achieve and maintain its integrity

Definition 2: Blockchain is a decentralized consensus mechanism. In a blockchain, all peers eventually come to an agreement regarding the state of a transaction. Blockchain is a distributed shared ledger.

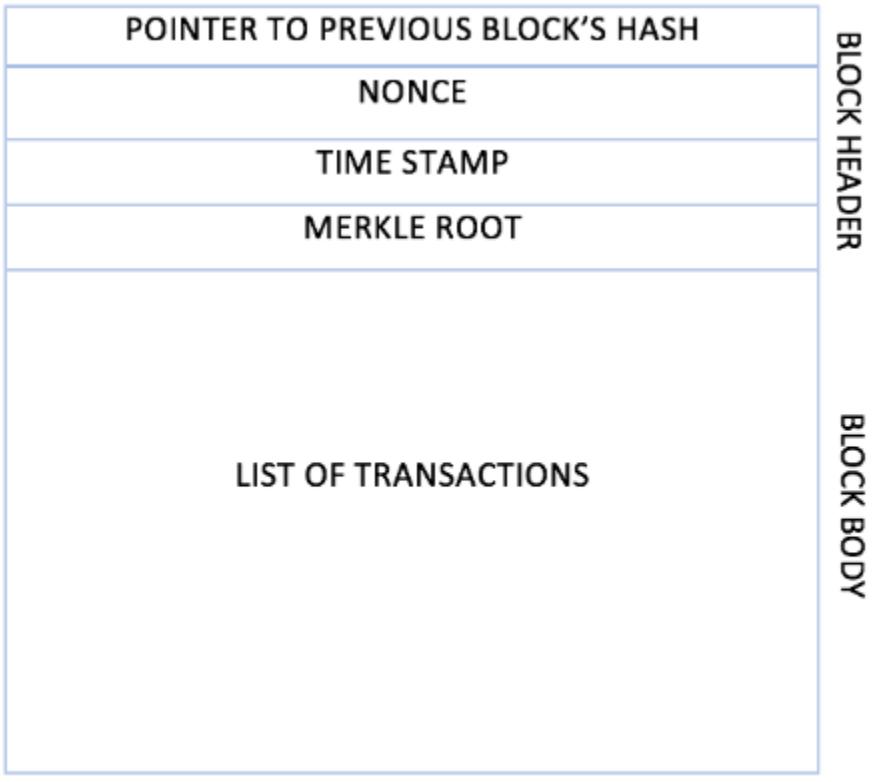
Definition 3: Blockchain can be considered a shared ledger of transactions. The transaction are ordered and grouped into blocks. Currently, the real-world model is based on private databases that each organization maintains whereas the distributed ledger can serve as a single source of truth for all member organizations that are using the blockchain.

Definition 4: Blockchain is a data structure; it is basically a linked list that uses hash pointers instead of normal pointers. Hash pointers are used to point to the previous block.

The generic structure of a block

- A **block** is merely a selection of transactions bundled together and organized logically.
- A **transaction** is a record of an event, for example, the event of transferring cash from a sender's account to a beneficiary's account. A block is made up of transactions, and its size varies depending on the type and design of the blockchain in use.
- A reference to a previous block is also included in the block unless it is a genesis block.

- A **genesis block** is the first block in the blockchain that is hardcoded at the time the blockchain was first started.
- There are just a few attributes that are essential to the functionality of a block: the block header, which is composed of pointer to previous block, the timestamp, nonce, Merkle root, and the block body that contains transactions.
- There are also other attributes in a block, but generally, the aforementioned components are always available in a block.
- A **nonce** is a number that is generated and used only once. A nonce is used extensively in many cryptographic operations to provide replay protection, authentication, and encryption. In blockchain, it's used in PoW
- **Merkle root** is a hash of all of the nodes of a Merkle tree. Merkle trees are widely used to validate the large data structures securely and efficiently. In the blockchain world, Merkle trees are commonly used to allow efficient verification of transactions. This means that verifying only the Merkle root is required to verify all transactions present in the Merkle tree instead of verifying all transactions one by one.



Generic elements of a blockchain

ADDRESSES

- Addresses are unique identifiers that are used in a transaction on the blockchain to denote senders and recipients.
- An address is usually a public key or derived from a public key.
- While addresses can be reused by the same user, addresses themselves are unique. A single user may not use the same address again and generate a new one for each transaction.

TRANSACTION

A transaction is the fundamental unit of a blockchain. A transaction represents a transfer of value from one address to another.

BLOCK

- A block is composed of multiple transactions and some other elements such as the previous block hash (hash pointer), timestamp, and nonce.

PEER-TO-PEER NETWORK

- As the name implies, this is a network topology whereby all peers can communicate with each other and send and receive messages.

SCRIPTING OR PROGRAMMING LANGUAGE

- This element performs various operations on a transaction. Transaction scripts are predefined sets of commands for nodes to transfer tokens from one address to another and perform various other functions.
- Turing complete programming language is a desirable feature of blockchains.
- Turing Complete refers to a machine that, given enough time and memory along with the necessary instructions, can solve any computational problem, no matter how complex.

VIRTUAL MACHINE

- A virtual machine allows Turing complete code to be run on a blockchain (as smart contracts). Virtual machines are not available on all blockchains; however, various blockchains use virtual machines to run programs, for example Ethereum Virtual Machine (EVM) and Chain Virtual Machine (CVM).

STATE MACHINE

- A blockchain can be viewed as a state transition mechanism whereby a state is modified from its initial form to the next and eventually to a final form as a result of a transaction execution and validation process by nodes.

NODES

- A node in a blockchain network performs various functions depending on the role it takes. A node can propose and validate transactions and perform mining to facilitate consensus and secure the blockchain.
- Nodes can also perform other functions such as simple payment verification (lightweight nodes), validators, and many others functions depending on the type of the blockchain used and the role assigned to the node.

SMART CONTRACTS

- These programs run on top of the blockchain and encapsulate the business logic to be executed when certain conditions are met.
- The smart contract feature is not available in all blockchains but is now becoming a very desirable feature due to the flexibility and power it provides to the blockchain applications.

Features of a blockchain

A blockchain performs various functions. These are described below in detail.

DISTRIBUTED CONSENSUS

Distributed consensus is the major underpinning of a blockchain. This enables a blockchain to present a single version of truth that is agreed upon by all parties without the requirement of a central authority.

TRANSACTION VERIFICATION

Any transactions posted from nodes on the blockchain are verified based on a predetermined set of rules and only valid transactions are selected for inclusion in a block

PLATFORMS FOR SMART CONTRACTS

A blockchain is a platform where programs can run that execute business logic on behalf of the users. As explained earlier, not all

blockchains have a mechanism to execute smart contracts; however, this is now a very desirable feature.

TRANSFERRING VALUE BETWEEN PEERS

Blockchain enables the transfer of value between its users via tokens. Tokens can be thought of as a carrier of value.

GENERATING CRYPTOCURRENCY

This is an optional feature depending on the type of blockchain used. A blockchain can generate cryptocurrency as an incentive to its miners who validate the transactions and spend resources in order to secure the blockchain.

PROVIDER OF SECURITY

Blockchain is based on proven cryptographic technology that ensures the integrity and availability of data.

IMMUTABILITY

This is another key feature of blockchain: records once added onto the blockchain are immutable. There is the possibility of rolling back the changes but this is considered almost impossible to do as it will require an unaffordable amount of computing resources.

UNIQUENESS

This feature of blockchain ensures that every transaction is unique and has not been spent already. This is especially relevant in cryptocurrencies where much desirable detection and avoidance of double spending are a key requirement.

SMART CONTRACTS

Blockchain provides a platform to run smart contracts. These are automated autonomous programs that reside on the blockchain and encapsulate business logic and code in order to execute a required function when certain conditions are met. This is indeed a revolutionary feature of blockchain as it allows flexibility, programmability, and much desirable control of actions that users of blockchain need to perform according to their specific business requirements.

HOW BLOCKCHAINS ACCUMULATE BLOCKS

1. A node starts a transaction by signing it with its private key.
2. The transaction is propagated (flooded) by using much desirable Gossip protocol to peers, which validates the transaction based on pre-set criteria. Usually, more than one node is required to validate the transactions.
3. Once the transaction is validated, it is included in a block, which is then propagated on to the network. At this point, the transaction is considered confirmed.
4. The newly created block now becomes part of the ledger and the next block links itself cryptographically back to this block. This link is a hash pointer. At this stage, the transaction gets its second confirmation and the block gets its first.
5. Transactions are then reconfirmed every time a new block is created. Usually, six confirmations in the bitcoin network are required to consider the transaction final.

1.2 Blockchain versus distributed ledger technology versus distributed databases

It is a digital system used for storing the transaction of assets, even when the data is stored at multiple places simultaneously. It might sound like a traditional database, but is different because of the fact that there is no centralized storage place or administration functionality. Meaning, every node of the ledger processes and validates every item, and this way, contributes to generating a record of each item and building a consensus on each item's veracity.

Benefits of distributed ledger technology

1. The use of DLT ensures that the data is 100% tamper proof till the database ledger is distributed.
2. It offers a highly secure and trustworthy experience by distributed ledger technology companies.
3. A decentralized private distributed network enhances the robustness of the system and assures continuous operation without any interruption.

A distributed ledger is a type of database that keeps multiple copies of information in different locations, which it can update consistently. This allows anyone who has access to the copy of the shared ledger to verify the same data (information). An important requirement for this type of database is maintaining integrity and availability. The three basic requirements are:

Consistency
Availability
Partition tolerance

Based on some of the user- and application-level features and heuristics, we can observe the following differences:

Feature	Blockchain	DLT	Distributed Database
Immutability	The information persisted in blockchains cannot be removed or updated without a new identifier to the target data.	Although most DLTs are pro-immutability, there are a few exceptions where immutability is not a design constraint.	Most distributed databases are not immutable due to design limitations.
Logical execution	Smart contracts can be used to enforce business logic on data from a blockchain.	DLTs offer the execution of logic on the data within them, as well as on user inputs.	User-defined functions and stored procedures are normal approaches that are used here.
Accessibility	Data in a public blockchain is stored in the form of a transaction or account states in a block and is visible and accessible with middleware.	Data is private in a DLT and may, in some cases, be encrypted in the DLT entry. Data can only be accessed by participating stakeholders.	Data is persisted within the distributed data clusters spread across the globe for faster access, using traditional client-server techniques.
Verifiability	All the transactions are verified before a change is made to the state of an account.	Most DLTs do not offer verification algorithms or modules as a design restriction to applications.	The verifiability of data is not offered as the state of accounts is not persisted in a specific structure.
Incentivization	Most blockchains use several economic models to incentivize their stakeholders.	Stakeholders in a DLT group host the nodes and are self-incentivized to run their business more confidently.	The company manages the data for sustainability and so no extra incentivization can be observed.

Comparing the technologies with examples

The following scenario is provided to aid your understanding of the core differences between the preceding three implementations.

Imagine that you plan to create a new digital platform for stock photography. If you want to invite photographers all over the world to use the platform and allow them to upload their work and be incentivized with their royalties automatically paid off by the consumers, you'd want to use blockchain to offer public access and incentivization and to transfer the royalties directly from the consumer to the photographer, thereby eliminating the need for a third party performing the duty payment collection, guaranteeing the return of royalties but with a service fee. However, if you want your platform to form a private

consortium of photographers, with their art exclusively available to a limited audience, and to handle royalties in conjunction with other means, you would use a DLT. Finally, if you intend to use your platform to exhibit art by an eligible set of photographers that are accessible across the globe, with or without royalties (which is handled offline), you'd form a cluster of nodes that host this data and logic to handle access and payments. So, you would use distributed databases

1.3 Public Vs Private Vs Permissioned Blockchain

Public Blockchain

- These blockchains are also known as permission-less ledgers.
- These blockchains are open to the public and anyone can participate as a node in the decision-making process.
- Public blockchains were designed and developed with a focus on ensuring that any number of interested parties can execute the business logic and access transactional information.
- Similarly any interested party can also verify and validate the transactions incoming to the network as well as be rewarded for the process.
- Users may or may not be rewarded for their participation. These ledgers are not owned by anyone and are publicly open for anyone to participate in.
- All users of the permission-less ledger maintain a copy of the ledger on their local nodes and use a distributed consensus mechanism in order to reach a decision about the eventual state of the ledger.

Private Blockchain

- Private blockchains are implemented to ensure that access to business information is limited and only accessible to a limited set of participating stakeholders.

- Private blockchains as the name implies are private and are open only to a consortium or group of individuals or organizations that has decided to share the ledger among themselves.

Permissioned blockchains

- Permissioned blockchains are hybrid implementation of both public and private blockchains.
- A permissioned ledger is a blockchain whereby the participants of the network are known and already trusted.
- Permission blockchains are implemented if data is to be accessed by a specific stakeholder. This is achieved by leveraging private networking as well as the encryption of transactional user data.
- Permissioned ledgers do not need to use a distributed consensus mechanism, instead an agreement protocol can be used to maintain a shared version of truth about the state of the records on the blockchain.
- There is also no requirement for a permissioned blockchain to be private as it can be a public blockchain but with regulated access control.

Comparing usage scenarios The following table shows how the three types of blockchain can be used in various scenarios. They are:

Attribute versus variant	Public blockchains	Private blockchains	Permissioned blockchains
Network barrier	Access to the network is not restricted. The details inside public blockchains are widely accessible to all users.	Access to the network is limited by an IP or a DNS. Only a few people with suitable credentials can join the network.	Access to the network is limited to verified participants. Only selected people can join the network with limited permissions to read, write, or both.
Restrictions	There are many different actions that the user can perform, such as develop a smart contract and use it, host a node as a validator, and so on.	Virtually, there are only two common roles for members in a private blockchain—facilitated nodes as validators and DApps users.	Based on the role of the members, the users may be able to deploy DApps, use DApps, validate transactions, or all three.
Encryption	Almost all of the user data in blocks is not encrypted as the general goal is to serve the information to a public audience.	Encryption may not be used if there is a trust quotient between the participating stakeholders.	Encryption is widely used as it involves various stakeholders in the networks with potential conflicts of interest.

The advantages of **public and permissionless** blockchain are as follows:

- There's no infraastructure costs for creating and running **decentralized applications** (**dApps**)
- There's no need for a trusted party or intermediary; there is no intermediary
- The network is open and transparent and offers anonymity
- The network offers trustlessness and immutability

The advantages of **public and permissioned** blockchain are as follows:

- No infrastructure costs for creating and running dApps
- No need for a trusted party or intermediary; there is no intermediary
- Scalable, fast, and lower cost

The advantages of **private and permissionless** blockchain are as follows:

- Cost of transaction is reduced
- No need for reconciliations
- Simplified document handling
- Reduced data redundancy

The advantages of **private and permissioned** blockchain are as follows:

- There's better governance than public blockchain
- The cost of transactions is reduced. There is no need for reconciliations.
- Document handling is simplified and data redundancy is reduced
- As participants are preapproved and identities are known, there is better privacy and security

The disadvantages of **public and permissionless** blockchain are as follows:

- **Scalability**: There is a limitation on the number of transactions that can be created, which can often reach to minutes at the peak period. Hence, such decentralized systems are not scalable.
- **Slowness and higher cost**: As the number of transactions increases, so does the cost of executing those transactions, which leads to the clogging of miners to execute high-value transactions, and so, the system becomes slow and costly.

- **Identity is anonymous**: Anonymous participants could be malicious.
- **Immutability is a challenge**: Although immutability of transactions and blocks is the major feature of public blockchain,

immutability of code (smart contract) is a challenge for the blockchain network. Blockchain considers smart contract deployment as a transaction and as they are transactions, they are immutable.

The disadvantages of **public and permissioned** blockchain are as follows:

- Identity is anonymous—participants, being anonymous, can be malicious
- Immutability is a challenge
- There's no finality and 51% attack (theory)
- It can lead to centralization

The disadvantages of **private and permissionless** blockchain are as follows:

- It still has an intermediary and hence it is not decentralized.
- It is centralized and hence it is not decentralized. However, it can be distributed.
- As participants are not preapproved, identities are not known, although malicious users cannot perform write transactions and can only read information.

The disadvantages of **private and permissioned** blockchain are as follows:

- **Not fully distributed**: It still has an intermediary and hence it is not fully distributed.
- **Consortium formation is a challenge**: Formation of a consortium needs like-minded enterprises to collaborate over common business problems.

1. 4 Privacy in blockchains

2. Blockchains add new values, such as transparency and provenance of information.
3. However, many people mistakenly believe that all transactions are publicly viewable in a blockchain. However, in reality, not all the blockchains necessarily facilitate transactions with public viewability:

Motivations:

- Several applications on blockchains are not just built for enterprise use cases. Many blockchain applications are now targeting mass consumer audiences.
- The internet, in recent years, has become a testbed for various approaches in preserving the privacy of users.
- Unlike any other trend or improvement on the current state of the internet, most blockchain projects aim to deliver a privacy-first mode of operation to users by leveraging pseudonymous cryptographic wallets without revealing the identity of the senders and receivers.
- Some examples of privacy-first blockchains include Monero, Quorum, and Zcash.

Approaches:

- Public blockchains have design limitations with respect to privacy. As global access to user data is one of the prominent objectives of a public blockchain, we see very few applications of cryptography in them. However, the emerging blockchains such as Zcash, Monero aim to offer untraceable, secure, and analysis-resistant transactional environments for users with their own cryptocurrencies.
- This is made possible by leveraging a Zero - Knowledge proof mechanism that prevents double spend of the same

cryptocurrencies, but at the same time preserves the fundamental values of blockchain. On the other hand, private and permissioned blockchains consider protecting the privacy of the participating stakeholders as high priority. One well-known private implementation is the Quorum blockchain, which was developed by JP Morgan Chase & Co. Quorum offers transaction-level privacy, yet at the same time offers network-level transparency on the actions by all the stakeholders in the network by using a privacy engine called Constellation.

- Constellation encrypts the transaction payload with a special key generated from the public/private key pair of the users involved in the transaction. It also facilitates the deployment and operation of private smart contracts within an existing network.

1.5 Bitcoin

- Bitcoin is a virtual currency on a peer-to-peer network with users and validators distributed across the network.
- With the help of the Bitcoin blockchain network, users can transfer cryptocurrency in a truly decentralized manner, without a need for either a central bank, a clearing house, or an intermediary.
- The transfer of Bitcoin between users is recorded in the form of a transaction, which is later verified, mined, and added to a canonical link of blocks.
- Bitcoin is believed to have been created by a group work working under the pseudonym Satoshi Nakamoto, with most of its features and functionalities derived based on existing techniques in cryptographic hashes, peer-to-peer network communication, and immutable data structures.

The following diagram illustrates how Bitcoin mining works in a single node, as well as in pool environments:

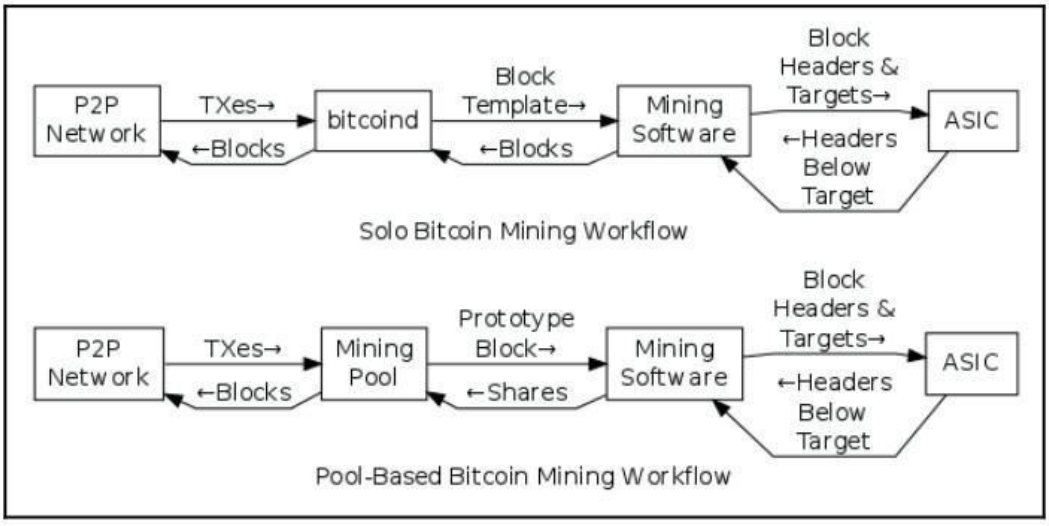


Fig 1.1: Two types of mining in the Bitcoin blockchain network

A brief overview of Bitcoin This section offers historical background on the Bitcoin cryptocurrency, along with factual information on its current state as well as the technical and architectural limitations perceived by experts in the market.

Motivation:

- One of the core motivations behind this cryptocurrency was that the currencies rolled out by central banks could not be trusted as they may not be backed by real collateral.
- This led to the adoption of a free-market approach to the production, distribution, and management of the money, with proof of work for every Bitcoin minted, thereby eliminating the need for central banks and other intermediaries.

Facts:

- The virtual currency was open sourced in 2009 with a maximum supply of 21 million Bitcoin that can be minted. Around 18.3 million Bitcoin has been mined to date, with at least three forks.

The following are the prominent Bitcoin forks:

- Bitcoin Cash (with larger block sizes)
- Bitcoin Gold (preserving GPU-based Proof of Work (PoW) mining instead of ASICs) and
- Bitcoin Adjustable Block-Size Cap (ABC) with 32 MB of blocksize)
- Bitcoin Satoshi's Vision (SV) with an increased block size of 128 MB At the time of writing this book, each Bitcoin was valued at around USD 6,806.00. The Bitcoin blockchain network incentivizes validating miners by charging users who transfer Bitcoin with a small fee, which is awarded to the winning block maker as per the PoW algorithm.
- **Criticism:** The cryptocurrency is alleged to be one of the prime choices of medium for illicit transactions. One of the major crackdowns of this sort of use came from a renowned online black market on the darknet, Silk Road. The FBI shut down the website in late 2013.

An introduction to Bitcoin

- Bitcoin is a completely decentralised, peer-to-peer, permissionless cryptocurrency put forth in 2009 by Satoshi Nakamoto.
- Bitcoin is the first blockchain application. Bitcoin is digital cash. It is a digital currency and online payment system in which encryption techniques are used to regulate the generation of units of currency and verify the transfer of funds, operating independently of a central bank.

- Bitcoin blockchain size is growing exponentially. Bitcoin is the first decentralized digital currency.
- Bitcoin is digital cash that is transacted via the Internet in a decentralized trustless system using a public ledger called the blockchain.

Mining:

- It is a mechanism to generate hash of the block.
- Mining involves creating a hash of a block of transactions that cannot be easily forged, protecting the integrity of the entire blockchain without the need for a central system.

Bitcoin Mining:

H(k) = Hash(H(k-1) || T || Nonce)

Here,

H(k-1) = Previous block hash

T = List of transactions

Nonce = Miners find this nonce as per the complexity(number of zeros at the prefix

Properties of Bitcoin:

- It's an international network of payments.
- It uses cryptography to control its creation and management, rather than relying on central authorities such as governments, banks, union territories, or intermediaries.
- It's not printed but is produced by people using software that solves mathematical problems.
- It is controlled and limited in supply, which arrests the hyperinflation problem. For example, whenever African countries were short of currency notes, they had to print more notes, which resulted in hyperinflation and brought the value of the currency down.
- Since the arrival of Bitcoin, the way Bitcoin programs are written means there will always be a maximum of 21 million Bitcoins available across the globe. The moment 21 million Bitcoins have

been mined, the program will not generate any more new Bitcoins. Hence, Bitcoins will be limited in supply and this will arrest the problem of hyperinflation.

History of Bitcoin

- Bitcoin was created in 2009 (released on January 9, 20096) by an unknown person or entity using the name Satoshi Nakamoto.
- The concept and operational details are described in a concise and readable white paper, "Bitcoin: A Peer-to-Peer Electronic Cash System."
- Payments using the decentralized virtual currency are recorded in a public ledger that is stored on many—potentially all—Bitcoin users' computers, and continuously viewable on the Internet.
- Bitcoin is the first and largest decentralized cryptocurrency
- There are hundreds of other "altcoin" (alternative coin) cryptocur- rencies, like Litecoin and Dogecoin, but Bitcoin comprises 90 percent of the market capitalization of all cryptocurrencies and is the de facto standard.
- Bitcoin is pseudonymous (not anonymous) in the sense that public key addresses (27–32 alphanumeric character strings; similar in function to an email address) are used to send and receive Bitcoins and record transactions, as opposed to personally identifying information
- Bitcoins are created as a reward for computational processing work, known as mining, in which users offer their computing power to verify and record payments into the public ledger.
- Individuals or companies engage in mining in exchange for transaction fees and newly created Bitcoins. Besides mining, Bitcoins can, like any currency, be obtained in exchange for fiat money, products, and services.

- Users can send and receive Bitcoins electronically for an optional transaction fee using wallet software on a personal computer, mobile device, or web application

Blockchain and Bitcoin

The block chain is a **shared public ledger** on which the entire Bitcoin network relies. All confirmed transactions are included in the block chain. It allows Bitcoin wallets to calculate their spendable balance so that new transactions can be verified thereby ensuring they're actually owned by the spender. The integrity and the chronological order of the block chain are enforced with cryptography.

A transaction is **a transfer of value between Bitcoin wallets** that gets included in the block chain.

- Bitcoin wallets keep a secret piece of data called a private key or seed, which is used to sign transactions, providing a mathematical proof that they have come from the owner of the wallet.
- The signature also prevents the transaction from being altered by anybody once it has been issued.
- All transactions are broadcast to the network and usually begin to be confirmed within 10-20 minutes, through a process called mining.

Mining is a **distributed consensus system** that is used to confirm pending transactions by including them in the block chain.

It enforces a chronological order in the block chain, protects the neutrality of the network, and allows different computers to agree on the state of the system.

- To be confirmed, transactions must be packed in a block that fits very strict cryptographic rules that will be verified by the network.
- These rules prevent previous blocks from being modified because doing so would invalidate all the subsequent blocks.
- Mining prevents any individual from easily adding new blocks consecutively to the block chain.
- In this way, no group or individuals can control what is included in the block chain or replace parts of the block chain to roll back their own spends.

Bitcoin wallet

What is a bitcoin wallet?

A bitcoin wallet is a software program that runs on a computer or a dedicated device that provides the functionality required to secure, send and receive bitcoin.

The wallet secures the cryptographic keys — essentially a very specialized type of password — that proves the ownership of a specific amount of bitcoin on the Bitcoin network.

Anytime a bitcoin transaction is executed, ownership of the bitcoin transfers from the sender to the recipient, with the network designating the recipient's keys as the new "password" for accessing the bitcoin.

- Bitcoin uses a system called public-key cryptography (PKC) to preserve the integrity of its blockchain. Originally used to encrypt and decrypt messages, PKC is now commonly used on blockchains to secure transactions. This system allows only individuals with the right set of keys to access specific coins.
- There are two types of keys required to own and execute bitcoin transactions: A private key and a public key. Both keys are strings

of randomly generated alphanumeric characters used to encrypt and decrypt transactions. On the bitcoin network, PKC implements one-way mathematical functions that are easy to solve in one way and almost impossible to reverse.

- The blockchain uses the one-way mathematical algorithm to create a public key from the private key. With this, it is practically impossible to regenerate the private key from the public key.
- To execute transactions, you are required to use your private key and public key to encrypt and sign your Bitcoin transactions. Also, you have to include the public address of the recipient. With this, only the recipient with the right private key can unlock or claim the transferred bitcoin.

What is proof of work?

- Proof of work is a mechanism used in the Bitcoin network to ensure that the network remains secure and that new transactions can be verified and added to the blockchain.
- It involves solving a complex mathematical problem to create a new block of transactions on the blockchain, which requires a significant amount of computational power.
- The process of solving the mathematical problem is known as mining, and the people or organizations that do this are known as miners. Miners compete to be the first to solve the problem and create the new block, and the first one to do so is rewarded with a certain number of Bitcoins.
- When a new block is discovered, the successful miner who found it through the mining process gets to fill it with 1 megabyte's worth of validated transactions. This new block is then added to the chain and everyone's copy of the ledger is updated to reflect the new data.

- This process helps to secure the network and ensures that new transactions are properly verified and added to the blockchain.
- Proof of work is an important part of the Bitcoin network and is one of the main ways that the network is able to maintain its security and integrity.

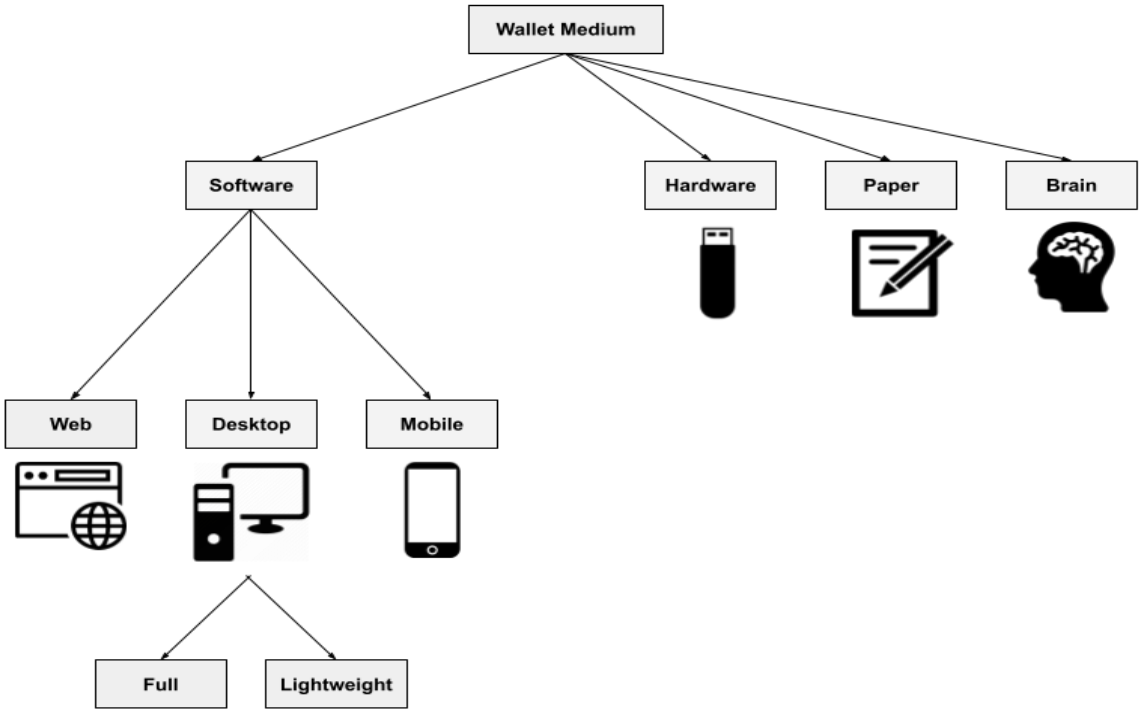
Types of wallets

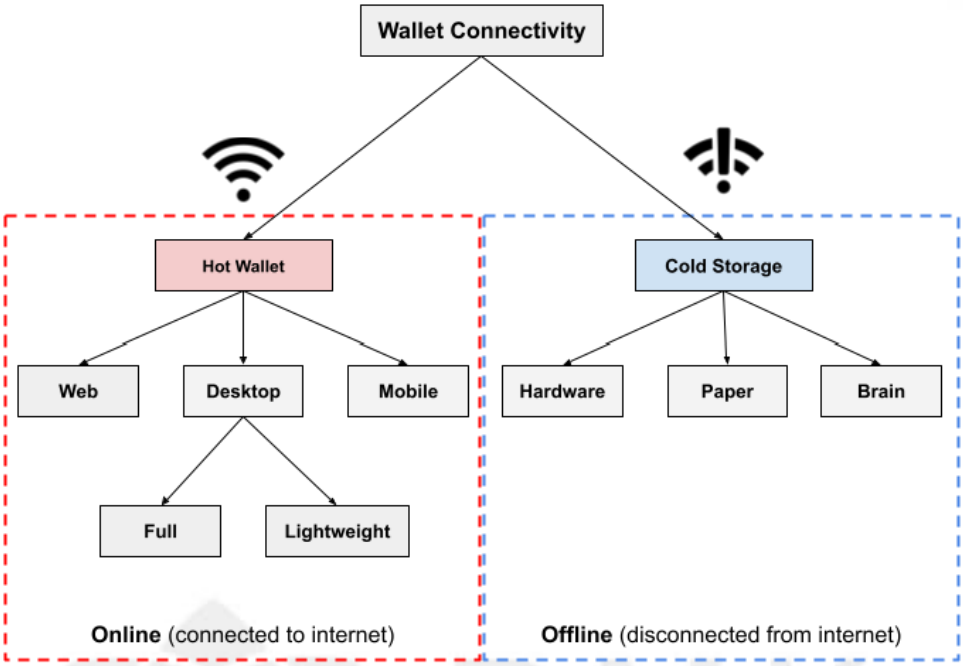
- A crypto wallet is used to interact with a blockchain network. The three major types of crypto wallets are hardware, software, and paper wallets.
- Based on their work, they can be further classified as cold or hot wallets.
- Software-based wallets are more accessible and more convenient, whereas hardware ones are the most secure.
- Paper wallets are printed out on paper and are now unreliable and obsolete.
- In reality, crypto wallets don't store the currency but act as a tool of interaction with blockchain, i.e., generating the necessary information to receive and send money via blockchain transactions.
- The information comprises pairs of private and public keys. Based on these keys, an alphanumeric identifier called address is generated.
- In essence, this address specifies the location to which coins can be sent to the blockchain. The address can be shared to receive funds, but private keys are to be never disclosed.
- The private key can be used on any wallet for accessing the cryptocurrency. As long as the private key is known, funds are accessible on any device.

- Also, coins are just transferred from one address to another, never leaving the blockchain.

Bitcoin wallets

- There are several types of wallet available, in which we can hold our Bitcoins safely. Each of these wallets has their own function and ways to operate.
- The different kinds of the wallet are as follows:





Software wallets

- A **software wallet** is a computer program or mobile app that holds private keys online.
- They connect to the Bitcoin network through trusted full nodes, centralized services (such as centralized exchanges), or are full nodes themselves.
- Bitcoin Armory is an example of a software wallet and is supposed to be the most stable and secure wallet of all.

There are three types of software wallets:

- **Desktop wallets** that are used on a desktop or laptop computer. Desktop wallets are programs which are run from your desktop or laptop computer. They provide a streamlined, easy-to-use interface for users to interact with their crypto holdings. desktop wallets only connect to the internet when necessary for completing transactions. They are most often non-custodial, which means the wallet owner bears sole responsibility for safeguarding their private keys.

Benefits

- Convenient method of securely conducting crypto transactions using a computer
- Easy to use and free to download
- Non-custodial (in most cases), meaning the user controls their private keys (and their crypto)

Drawbacks

- Like all "hot" wallets, being kept on an online device makes them potentially susceptible to hackers
- Can be vulnerable to computer viruses or malware
- Lack of portability compared to mobile wallets

Popular desktop wallets

- Electrum
- Exodus
- Atomic Wallet

Mobile wallets that work as a mobile "app" and are used on your smartphones

Web-based wallets that work as a browser extension and are used in your web browser.

Hardware wallets

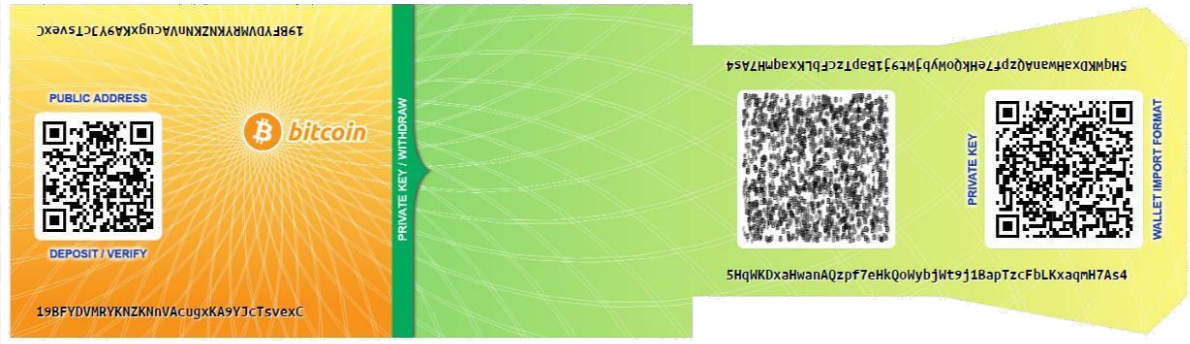
- A **hardware wallet** is a small, portable physical device (similar to a USB flash drive) that keeps your private keys isolated from the internet at all times.
- To transact, a computer and the device manufacturer's software application are required to authorize (or "sign") the transaction since the private keys are kept offline.
- Hardware wallets provide an extra layer of security than software wallets but most people don't use them since they're complicated and the devices themselves are expensive.

Paper wallets

- A **paper wallet** is literally a piece of paper with your address and private key printed or written on it.
- Paper wallets are created by downloading software, then running the software offline (disconnected from the internet) to generate a public/private key pair which you print out on a piece of paper.
- It is a physically printed QR coded form wallet. Some wallets

PROS	CONS

allow downloading the code to generate new addresses offline.



Brain wallets

- A **brain wallet** refers to a private key that is stored in the user's memory in the form of a **seed phrase,** which is a sequence of 12-24 words (often called a "mnemonic phrase").

Cold wallets

- The "**cold**" in a cold wallet or cold storage refers to the lack of connection to the internet.
- Because cold wallets are never connected to the internet, online theft of your bitcoins is impossible. A hacker would have to physically steal the device to gain access to your bitcoins.

More secure than hot wallets due to offline storage of keys.	If you lose your keys and don't and didn't back them up, recovery is impossible.
Most popular hardware wallets are supported by hot wallets	Transactions usually take longer.

Custodial Wallets

- A **custodial wallet** is controlled by a trusted entity, with the user typically having to access its contents via a web interface. These sites store private keys for you so you don't have to worry about them.
- Custodial wallets are also known as "**hosted wallets**". Wallets are considered "hosted" because a third party holds your crypto for you, similar to how a bank holds your money in a savings or checking (or current) account.
- Wallets provided by **centralized crypto exchanges (CEXs)** are a common example of custodial wallets. They hold your bitcoins and other cryptocurrencies in an account, and they manage and control the keys.
- This means that they have full control over your funds.

Non-Custodial Wallets

- A **non-custodial wallet** gives the user full control over their funds and the associated private keys or seed phrase.
- This sounds great but that also means that YOU are solely responsible for the security of your own private keys.
- While non-custodial wallets provide the software necessary to manage your crypto, the responsibility of keeping your private keys safe falls entirely on YOU.

- There is no third party or a "custodian" to keep your crypto safe.

1.6 Ethereum

Ethereum is a public blockchain that was designed by Vitalik Buterin in 2013 as an enhancement to the incumbent Bitcoin blockchain, by including transaction-based state management with business logic scripting using a special-purpose programming language and a virtual machine called the Ethereum Virtual Machine (EVM).

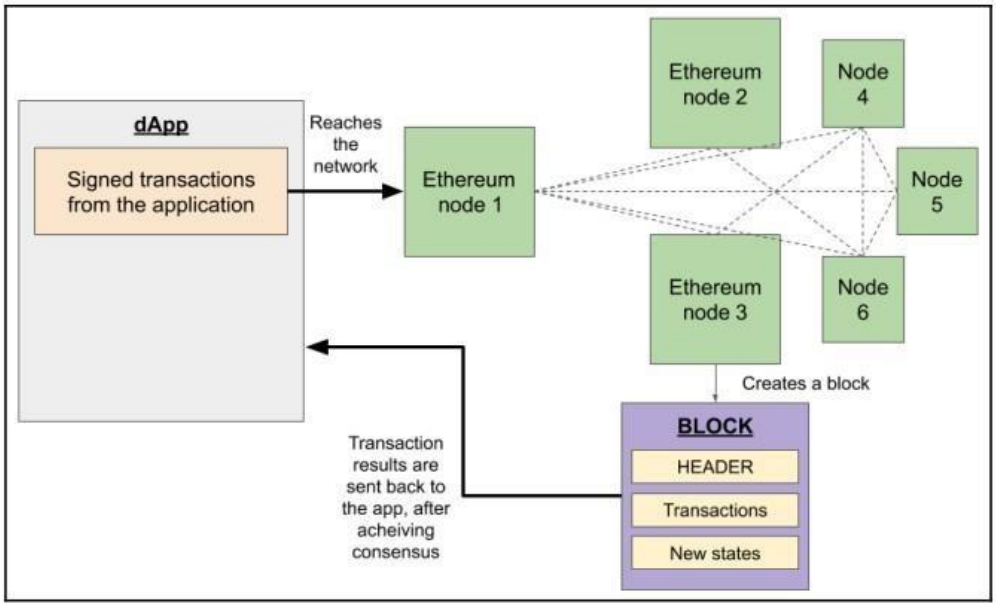


Fig 1.2: Block creation in Ethereum

A brief overview of Ethereum

Motivation:

The main motivation behind Ethereum was to support building decentralized applications on the powerful medium of blockchain. Unable to convince the Bitcoin community of the need for a scripting language, Vitalik and a like-minded group of people created Ethereum.

Facts:

- The project was open sourced with an initial release date of July 30, 2015.

- The research and development upgrades to the Ethereum network is managed under the Ethereum Foundation, financially supported by the initial crowd sale of the Ether (ETH) token from July to August 2014. Around 105 million ETH has been minted so far.
- Ethereum has one major fork called Ethereum Classic (the original Ethereum blockchain that denied the DAO hard fork and retained the original unaltered state of the Ethereum network).
- The Ethereum blockchain network also incentivizes the validating nodes by charging users who make transactions on DApps or transfer ETH with a small fee, which is awarded to the winning block maker.
- The rules of creating blocks and the acceptance of blocks are specified by consensus algorithms called PoW or Proof of Stake (PoS).

Criticism:

- The Ethereum community has had to face some of the earliest criticism due to the hard-fork decision taken by the team, thereby contradicting some of the ideology and values of blockchain, such as immutability and immunity from human political dynamics.
- The network was later criticized and heavily scrutinized by the regulatory authorities due to the alleged Ponzi schemes offered by the Initial Coin Offerings (ICOs) without a stable product or service.
- A hard fork is defined as a radical change made to the protocol, thereby rendering some of the previous blocks and its transactions invalid.

1.7 Hyperledger

Introduction to Hyperledger

Hyperledger is an open source project hosted by the Linux Foundation in collaboration with various industry leaders in finance, banking, supply chain, manufacturing, and other domains to create standard blockchain technologies.

Overview of the project

- Linux Foundation announced the Hyperledger project on February 9, 2016, with an initial 30 founding corporate members, including Accenture, IBM, DTCC, Intel, and R3, among others.
- At the time of writing, the Hyperledger governing board consists of 21 board members and around 200 corporate members around the globe. The project hosts a dozen code repositories of blockchain frameworks and tools.
- A few significant examples are mentioned in the following section.

Hyperledger Fabric

- Hyperledger Fabric is a blockchain framework initially developed by the IBM and Digital Assets members.
- Fabric is a DLT that aims to provide a modular architecture for developers to use only what is needed.
- The framework supports the execution of logic abstracted into containers called chaincode.
- Using Fabric is easily enabled by the plethora of documentation, tutorials, and tools available for deploying business networks without much hassle.

Hyperledger Sawtooth

- Hyperledger Sawtooth is a blockchain framework that offers enterprises a secure leadership election of nodes in the network, with special modes for executing instructions.

- Sawtooth offers a powerful, developer-friendly Software Development Kit (SDK) for a majority of languages to write and deploy smart contracts.
- Notably, Sawtooth is one of the early live projects to experiment with WebAssembly (WASM) as a virtual medium for the execution of smart contracts.

1.8 Blockchain Platforms

Other Hyperledger frameworks and tools

Some of the other notable projects incubated under the Hyperledger umbrella are as follows:

Hyperledger Indy:

A blockchain platform to specially handle decentralized identities from inside or external systems Hyperledger Grid: A WASM-based project for building supply chain solutions

Hyperledger Quilt:

A blockchain tool to connect blockchain realms of different protocols using the Interledger Protocol (ILP) specifications

Hyperledger Caliper: A blockchain benchmarking tool to assess the performance of a specific blockchain with specific parameters such as Transactions Per Second (TPS), transaction latency, resource utilization, and so on With this basic understanding of Hyperledger, let's now explore other blockchain platforms available to developers.

Other blockchain platforms

Hashgraph, Corda, and IOTA Hashgraph is a DLT with a superior consensus mechanism leveraging Directed Acyclic Graphs (DAGs). Notably, the implementation of this project is not fully open source. The algorithm was designed and published by Leemon Baird and was initially released in 2017.

Corda

- It is an open source DLT maintained by the financial services consortium R3.
- Corda offers a smart contracts platform to allow businesses to execute complex agreements, associating multiple variants of asset classes across various business domains, including supply chain, healthcare, and finance.

IOTA

- It is an open source DLT that offers payment automation and secure communication between IoT devices.
- This project is maintained by the non-profit IOTA Foundation. Quoted as one of the promising ICOs, the project has delivered impressive wallets, a data marketplace for sensor data, and payment channels for quicker transaction settlements using a new special data structure called Tangle, eliminating the need for miners and traditional canonical representations of transactional data in blocks

1.9 Consensus Algorithms

Byzantine Generals problem

In 1982 a thought experiment was proposed by Lamport et al. whereby a group of army generals who are leading different parts of the Byzantine army are planning to attack or retreat from a city. The only way of communication between them is a messenger and they need to agree to attack at the same time in order to win. The issue is that one or more generals can be traitors and can communicate a misleading message. Therefore there is a need to find a viable mechanism that allows agreement between generals even in the presence of treacherous generals so that the attack can still take place at the same time. As an analogy with distributed systems, generals can be considered as nodes, traitors can be considered Byzantine (malicious) nodes, and the

messenger can be thought of as a channel of communication between the generals.

This problem was solved in 1999 by Castro and Liskov who presented the Practical Byzantine Fault Tolerance (PBFT) algorithm. Later on in 2009, the first practical implementation was made with the invention of bitcoin where the Proof of Work (PoW) algorithm was developed as a mechanism to achieve consensus.

Consensus

Consensus is a process of agreement between distrusting nodes on a final state of data. In order to achieve consensus different algorithms can be used. It is easy to reach an agreement between two nodes (for example in client-server systems) but when multiple nodes are participating in a distributed system and they need to agree on a single value it becomes very difficult to achieve consensus. This concept of achieving consensus between multiple nodes is known as **distributed consensus.**

CONSENSUS MECHANISMS

- A consensus mechanism is a set of steps that are taken by all, or most, nodes in order to agree on a proposed state or value.
- Consensus mechanisms have recently come into the limelight and gained much popularity with the advent of **bitcoin and blockchain**. There are various requirements which must be met in order to provide the desired results in a consensus mechanism.

The following are their requirements with brief descriptions:

Agreement: All honest nodes decide on the same value.

Termination: All honest nodes terminate execution of the consensus process and eventually reach a decision.

Validity: The value agreed upon by all honest nodes must be the same as the initial value proposed by at least one honest node.

Fault tolerant:

The consensus algorithm should be able to run in the presence of faulty or malicious nodes (Byzantine nodes).

Integrity: This is a requirement where by no node makes the decision more than once. The nodes make decisions only once in a single consensus cycle.

TYPES OF CONSENSUS MECHANISM

There are various types of consensus mechanism; some common types are described as follows:

Byzantine fault tolerance-based:

With no compute intensive operations such as partial hash inversion, this method relies on a simple scheme of nodes that are publishing signed messages. Eventually, when a certain number of messages are received, then an agreement is reached.

Leader-based consensus mechanisms:

- This type of mechanism requires nodes to compete for the leader-election lottery and the node that wins it proposes a final value.
- Many practical implementations have been proposed such as **Paxos,** the most famous protocol introduced by Leslie Lamport in 1989.
- In Paxos nodes are assigned various roles such as **Proposer, Acceptor, and Learner.**
- Nodes or processes are named replicas and consensus is achieved in the presence of faulty nodes by **agreement among a majority of nodes.**
- Another alternative to Paxos is RAFT, which works by assigning any of three states, that is, **Follower, Candidate, or Leader,** to the nodes.
- A Leader is elected after a candidate node receives enough votes and all changes now have to go through the Leader, who commits

the proposed changes once replication on the majority of follower nodes is completed.

Consensus in blockchain

Consensus is basically a distributed computing concept that has been used in blockchain in order to provide a means of agreeing to a single version of truth by all peers on the blockchain network.

Consensus Algorithm

Consensus algorithms are the specific instructions programmed on computers in a network so that they have a common definition of objects and instructions to agree on changes. Crashes, failures, and Byzantine faults in computers led to a better approach in forming an agreement in a digital network and so consensus algorithms rose to great heights, well before the dawn of the internet.

Proof of work

- It is introduced by the founder of Bitcoin – Satoshi Nakamoto
- It is one of the arliest consensus algorithm.
- This type of consensus mechanism relies on proof that enough computational resources have been spent before proposing a value for acceptance by the network.
- It leverages a combination of cryptography, P2P network communications and a Merkle data structure to offer a distributed immutable and cumulative state of accounts
- The solution computed by the first node is verified by the remaining nodes and the block producer is broadcast in the network:
- This is used in bitcoin and other cryptocurrencies. Currently, this is the only algorithm that has proven astonishingly successful against Sybil attacks.

Merit:

- The PoW algorithm has been time tested in the Bitcoin blockchain network and there is not a single hack/compromise of the account states in the network leading to double spend.

Demerit:

- As the PoW algorithm needs to find a solution to a mathematical problem, significant CPU cycles are required to generate hashes and so it is an energy-intensive technique.

Proof of stake

- **Proof of stake** (**PoS)** is a new consensus algorithm designed and developed to address some of the trade-offs of the PoW algorithm.
- This algorithm works on the idea that a node or user has enough stake in the system; for example the user has invested enough in the system so that any malicious attempt would outweigh the benefits of performing an attack on the system. This idea was first introduced by Peercoin and is going to be used in the Ethereum blockchain. Another important concept in Proof of Stake (PoS) is coin age, which is a derived from the amount of time and the number of coins that have not been spent. In this model, the chances of proposing and signing the next block increase with the coin age.
- The block-producing node is determined by an application of mathematical function involving a few determining factors, such as the stake (for example, ETH), the age of the node, and the randomization of eligible node candidates

Merit: The PoS algorithm is energy-efficient as there are fewer computational requirements and it does not select a block-producing node based on a solution-verification model.

Demerit: Although the PoS algorithm is efficient in its block times and is environment-friendly, there have been criticisms relating to the algorithm's vulnerability to capitalist attacks on the network of the node

owner and tries to compete with other candidates with a stupendous amount of cryptocurrency at stake, higher than all the other candidates.

Proof of Burn (**PoB**)

- It is a consensus algorithm with an interesting approach to solving transition problems from one version of cryptocurrency to another in the blockchains.
- Through the PoB algorithm, the old cryptocurrency (or its preceding version) is burnt in order to reduce its supply and gradually increase the supply of the new cryptocurrency (or its succeeding version).
- This consensus algorithm is practiced in various forms, including a method wherein users can transfer the old cryptocurrency to an unspendable wallet address in exchange for new ones.

Merit:

- The PoB algorithm is convenient during the transition of cryptocurrencies and network upgrades if the system trusts the participating entities.

Demerit:

- The PoB algorithm is usually applicable in PoW-based blockchains and so has a limitation of applicability. This is due to the requirement of verifiable proofs and the ability to decay the burnt coins over time, which is naturally capable through PoW algorithms.

Delegated Proof of Stake

- **Delegated Proof of Stake** (**dPOS**) is a consensus algorithm developed and used by the Block.one EOS platform. Under dPOS,

the token holders reserve the right to nominate the validators (also called block producers).

- The selection of block producers is a continuous process and performs the duties of packaging user transactions into blocks with Byzantine fault-tolerant safety:

Merit: dPOS is **Byzantine Fault Tolerance** (**BFT**) -ready and scales easily in a public network environment.

Demerit: Although dPOS is efficient, it is prone to capitalistic efforts to supersede other minor token stakeholders.

Proof of authority

- As the name suggests, the **Proof of Authority** (**PoA**) algorithm facilitates a distributed consensus with a few eligible verifiable nodes preserving the right to add transactions to blocks, if some criteria is met. There are many variants of the PoA algorithm, with or without the reputations of the validating nodes used in the public, private, and permissioned blockchains:

Merit: The PoA algorithm is energy-efficient and not prone to capitalistic pitfalls as the validator nodes are authorized to add transactions to blocks based on their reputation. If the node is observed to malfunction, its reputation is severely affected and cannot proceed as a validator.

Demerit: The PoA algorithm is partially centralized as the authority of adding or rejecting transactions lies in the purview of very few nodes in the network.

Practical Byzantine fault tolerance

- **Practical Byzantine Fault Tolerance** (**PBFT**) is one of the replication algorithms brought to light by academic research.

Authored by Miguel Castro and Barbara Liskov in 1999 (http://pmg.csail.mit.edu/papers/osdi99.pdf), this algorithm was primarily aimed at solving the Byzantine faults caused by the arbitrary point of failures in the nodes of a network.

- PBFT algorithm is used by the Hyperledger Fabric blockchain framework.
- PBFT is a Byzantine fault tolerance protocol for state machine replication. The state machine replication is a method for avoiding the unavailability of the online services due to failures by duplicating the system states over multiple machines. Due to the system replication nature, the system needs to keep all its states in sync by using the PBFT consensus algorithm.

Proof of elapsed time

- **Proof of Elapsed Time** (**PoET**) is a consensus algorithm developed and used by the Hyperledger Sawtooth blockchain framework. The PoET algorithm ensures security and randomness involved in the leadership of validator nodes with special CPU instructions available in most of the advanced processors featuring secure virtual environments:
- Introduced by Intel, it uses Trusted Execution Environment (TEE) to provide randomness and safety in the leader election process via a guaranteed wait time. It requires the Intel SGX (Software Guard Extensions) processor in order to provide the security guarantee and for it to be secure.

Merit:

PoET allows anyone with eligible hardware to participate as a validator node, allowing legitimate ways of verifying the leader election.

Demerit:

The cost of affording specialized hardware does not come cheap. So, there have been criticisms highlighting this as an unfair bar to enter the network.

1.10 Building DApps with blockchain tools

One of the main causes of the mainstream adoption of blockchain is the developer-led wave of evangelism for the technology.

This has been observed in the form of frameworks and tools at developer's disposal.

Blockchain toolchains and frameworks

The following list introduces several blockchain toolchains and frameworks that are popular with both developers and the associated solution community:

Truffle:

The Truffle framework was developed by ConsenSys as an open source project, offering a pipeline for the development, testing, and deployment of smart contracts targeted on the EVM.

Embark:

- The Embark framework was developed by Status as an open source project, offering a debugging and integration environment for Ethereum smart contract developers.
- Notably, Embark offers tighter integration with IPFS for the decentralized storage of contract data.

Hyperledger Composer:

This is an open source effort from the Linux Foundation, which offers tools to assist developers with converting requirements into proof of concept for the DevOps process, for spinning a new network as required.

MetaMask:

- This is a middleware that bridges an application running in the browser with the Ethereum blockchain.

- It is an open source initiative supported and consumed widely by all Ethereum developers.
- Users can perform transactions in a web application through MetaMask.

Ethers.js:

This is a JavaScript-based library with full implementation of the Ethereum wallet as per the specification.

Developers use this open source library to create user wallets, perform transactions, and much more. This library is also well known for its recent support for Ethereum Name Service (ENS).

Nethereum:

- This is an open source library used to build Ethereum-based blockchain solutions in .NET environments.
- Nethereum offers .NET developers an SDK called NuGet, which is integrated into the Visual Studio Integrated Development Environment (IDE) for using web3 functionalities across web and mobile applications.

Developing smart contracts using IDEs and plugins

Traditional software developers are more familiar and comfortable with working in IDEs, and the vibrant developer communities of blockchain have considered this.

The Remix IDE:

- Remix has been the de facto IDE for smart contract development and deployment. This open source IDE is used by developers who are interested in developing, debugging, and deploying solidity smart contracts for Ethereum network.
- Notably, this IDE works well with private networks and offers regular updates.

The EthFiddle IDE

- EthFiddle is an open source initiative by Loom Network to facilitate code experimentation online and provides the ability to share experimental code snippets of solidity smart contracts among developers for easier collaboration.

The YAKINDU plugin for Eclipse:

- Several enterprise developers have yearned for plugins for current IDEs, and this plugin offers just that.
- YAKINDU offers basic syntax highlighting and other common language package features for solidity smart contract development in the Eclipse IDE.

The Solidity plugin for Visual Studio Code:

- This plugin can be installed on Visual Studio Code, one of the most used IDEs.
- It boasts to be one of the leading plugins used for solidity smart contract development.

The Etheratom plugin for Visual Studio Code:

- Etheratom is a plugin available for GitHub's Atom editor, offering IDE features such as syntax highlighting, including a deployment interface to a local Ethereum node. It uses web3.js to interact with a local Ethereum node.

Assignments

Assignments

Toppers:

Investigate a specific industry or enterprise and analyze how a private Blockchain can be implemented to address specific business challenges or enhance operations.

Above Average:

Investigate how blockchain technology can be applied to improve transparency and traceability in supply chain management. Provide real-world examples.

Average:

Write a detailed report on private blockchains, explaining their key features, advantages, and differences from public blockchains.

Below Average:

Research and present an in-depth analysis of different consensus mechanisms (e.g., Proof-of-Work, Proof-of-Stake, Delegated Proof-of-Stake). Discuss their security, scalability, and energy efficiency.

Slow Learers:

Compare and contrast public and private blockchains. Discuss their use cases, advantages, and disadvantages. Provide examples of each.

Part A – Q & A

Unit - I

PART - A Questions

1. Define centralized systems and distributed systems (K1,CO1)

In centralized software systems, the components are located around and connected with one central component. In contrast, the components of distributed systems form a network of connected components without having any central element of coordination or control.

2. List the Advantages of Distributed Systems

- The major advantages of a distributed system over single computers are:
- Higher computing power
- Cost reduction
- Higher reliability
- Ability to grow naturally

3. Define Distributed Peer-to-Peer Systems

Peer-to-peer networks are a special kind of distributed systems. They consist of individual computers (also called nodes), which make their computational resources (e.g., processing power, storage capacity, data or network bandwidth) directly available to all other members of the network without having any central point of coordination. The nodes in the network are equal concerning their rights and roles in the system. Furthermore, all of them are both suppliers and consumers of resources.

4. Write the Purpose of the Blockchain

The blockchain is a tool for achieving integrity in distributed software systems. Hence, it can be seen as a tool to achieve a nonfunctional aspect of the implementation layer.

5. Define Trust and Integrity in Peer-to-Peer Systems

- Integrity is a nonfunctional aspect of a system to be safe, complete, consistent, correct, and free of corruption and errors.
- Trust is also the firm belief of humans in the reliability, truth, or ability of someone or something without evidence, proof, or investigation. Trust is given in advance and will increase or decline based on the results of interactions on an ongoing basis.

6. Define blockchains

The blockchain is a purely distributed peer-to-peer system of ledgers that utilizes a software unit that consist of an algorithm, which negotiates the informational content of ordered and connected blocks of data together with cryptographic and security technologies in order to achieve and maintain its integrity

7. Define block

A **block** is merely a selection of transactions bundled together and organized logically.

8. Define transaction

A **transaction** is a record of an event, for example, the event of transferring cash from a sender's account to a beneficiary's account. A block is made up of transactions, and its size varies depending on the type and design of the blockchain in use.

9. Define genesis block

A **genesis block** is the first block in the blockchain that is hardcoded at the time the blockchain was first started.

10. Define nonce

A **nonce** is a number that is generated and used only once. A nonce is used extensively in many cryptographic operations to provide replay protection, authentication, and encryption.

11. Define Merkle root

Merkle root is a hash of all of the nodes of a Merkle tree. Merkle trees are widely used to validate the large data structures securely and efficiently. In the blockchain world, Merkle trees are commonly used to allow efficient verification of transactions. This means that verifying only the Merkle root is required to verify all transactions present in the Merkle tree instead of verifying all transactions one by one.

12. Define state machine

A blockchain can be viewed as a state transition mechanism whereby a state is modified from its initial form to the next and eventually to a final form as a result of a transaction execution and validation process by nodes.

13. Define nodes

A node in a blockchain network performs various functions depending on the role it takes. A node can propose and validate transactions and perform mining to facilitate consensus and secure the blockchain.

14. What is smart contract

These programs run on top of the blockchain and encapsulate the business logic to be executed when certain conditions are met.

15. Define Consensus

Consensus is a process of agreement between distrusting nodes on a final state of data. In order to achieve consensus different algorithms can be used. It is easy to reach an agreement between two nodes (for example in client-server systems) but when multiple nodes are participating in a distributed system and they need to agree on a single value it becomes very difficult to achieve consensus. This concept of achieving consensus between multiple nodes is known as **distributed consensus.**

16. Write about consensus mechanisms

A consensus mechanism is a set of steps that are taken by all, or most, nodes in order to agree on a proposed state or value.

17. Define Consensus algorithms

They are the specific instructions programmed on computers in a network so that they have a common definition of objects and instructions to agree on changes. Crashes, failures, and Byzantine faults in computers led to a better approach in forming an agreement in a digital network and so consensus algorithms rose to great heights, well before the dawn of the internet.

18. Define Proof of work

It is introduced by the founder of Bitcoin – Satoshi Nakamoto. This type of consensus mechanism relies on proof that enough computational resources have been spent before proposing a value for acceptance by the network. It leverages a combination of cryptography, P2P network communications and a Merkle data structure to offer a distributed immutable and cumulative state of accounts

19. Define Proof of Burn (PoB)

It is a consensus algorithm with an interesting approach to solving transition problems from one version of cryptocurrency to another in the blockchains. Through the PoB algorithm, the old cryptocurrency (or its preceding version) is burnt in order to reduce its supply and gradually increase the supply of the new cryptocurrency (or its succeeding version).

20. Define Proof of Elapsed Time (PoET)

It is a consensus algorithm developed and used by the Hyperledger Sawtooth blockchain framework. The PoET algorithm ensures security and randomness involved in the leadership of validator nodes with special CPU instructions available in most of the advanced processors featuring secure virtual environments.

21. Define Public Blockchain

- These blockchains are also known as permission-less ledgers.
- These blockchains are open to the public and anyone can participate as a node in the decision-making process.
- Public blockchains were designed and developed with a focus on ensuring that any number of interested parties can execute the business logic and access transactional information.

22. Define Private Blockchain

- Private blockchains are implemented to ensure that access to business information is limited and only accessible to a limited set of participating stakeholders.
- Private blockchains as the name implies are private and are open only to a consortium or group of individuals or organizations that has decided to share the ledger among themselves.

23. Define Permissioned blockchains

- Permissioned blockchains are hybrid implementation of both public and private blockchains.
- A permissioned ledger is a blockchain whereby the participants of the network are known and already trusted.

24. Define Bitcoin

- Bitcoin is a completely decentralised, peer-to-peer, permissionless cryptocurrency put forth in 2009 by Satoshi Nakamoto.
- Bitcoin is the first blockchain application.
- It is permissionless , i.e. open to anyone.
- Bitcoin blockchain size is growing exponentially.

25. Define mining

- It is a mechanism to generate hash of the block.
- Mining involves creating a hash of a block of transactions that cannot be easily forged, protecting the integrity of the entire blockchain without the need for a central system.

26. What is meant by Bitcoin wallets?

They keep a secret piece of data called a private key or seed, which is used to sign transactions, providing a mathematical proof that they have come from the owner of the wallet.

27. Define Ethereum

Ethereum is a public blockchain that was designed by Vitalik Buterin in 2013 as an enhancement to the incumbent Bitcoin blockchain, by including transaction-based state management with business logic scripting using a special-purpose programming language and a virtual machine called the **Ethereum Virtual Machine** (**EVM**).

28. List the blockchain platforms

Hyperledger

Ethereum

IBM Bluemix

Multichain

Corda

openchain

29. Mention IDEs to develop smart contracts

Remix

Ethfiddle

30. Define Dapps

Decentralized applications (dApps) are digital applications or programs that exist and run on a blockchain or peer-to-peer (P2P) network of computers instead of a single computer. DApps (also called "dapps") are thus outside the purview and control of a single authority.

Part B – Questions

Part-B Questions

Q. No.	Questions	CO Level	K Level
1	Explain Bolckchain in detail	CO1	K2
2	Explain the structure of block in detail	CO1	K2
3	Explain consensus algorithms in detail	CO1	K2
4	Explain bitcoin in detail	CO1	K3
5	Explain in Ethereum in detail	CO1	K3
6	Explain different bitcoin wallets in detail	CO1	K3
7	Explain hyperledger in detail	CO1	K3
8	Explain different types of blockchain in detail	CO1	K3
9	Explain dApps in detail	CO1	K3
10	Explain privacy in blockchain	CO1	K3

Supportive online Certification courses (NPTEL, Swayam, Coursera, Udemy, etc.,)

R.M.K GROUP OF INSTITUTIONS

Supportive Online Certification Courses

Sl. No.	Courses	Platform
1	Blockchain and its Applications	NPTEL
2	DeFi Decentralized Finance	Coursera
3	Blockchain Basics	Coursera
4	Blockchain	NPTEL

Real time Applications in day-to-day life and to Industry

R.M.K GROUP OF INSTITUTIONS

Real Time Applications

1. Write an algorithm to search for a book (Tile of the Book is given as a input) in the library. Analyze the efficiency of the algorithm.
2. Consider the following algorithm.

   ALGORITHM Mystery(n)

   //Input: A nonnegative integer n

   S ←0

   For i ←1 to n do

   S ←S + i ∗i

   return S

   a. What does this algorithm compute?

   b. What is its basic operation?

   c. How many times is the basic operation executed?

   d. What is the efficiency class of this algorithm?

   e. Suggest an improvement, or a better algorithm altogether, and indicate its efficiency class. If you cannot do it, try to prove that, in fact, it cannot be done.
3. You are in charge of the cake for a child's birthday. You have decided the cake will have one candle for each year of their total age. They will only be able to blow out the tallest of the candles. Count how many candles are tallest. Write an algorithm for the above problem and analyse the efficiency of the algorithm.

Content Beyond Syllabus

R.M.K
GROUP OF
INSTITUTIONS

Unit I Content Beyond Syllabus

Important Data Structures

Graphs:

A graph G=(V,E) is defined by a pair of two sets: A finite set V of items called vertices and a set E of items called edges. If a pair of vertices (u,v) is same as (v,u) in a graph then the graph is called as undirected graph. If pair of vertices (u,v) and (v,u) is not same in a graph then it is called as directed graph or digraph. A graph with every pair of vertices connected by an edge is called complete. A graph with relatively few possible edges missing is called dense; a graph with few edges relative to number of its vertices is called sparse.

Graph representations: Adjacency Matrix and Adjacency List.

Adjacency Matrix: A graph with n vertices is represented by nxn matrix. The entries of the matrix are 0 and 1. The entry A[u,v]=0 indicates there is no edge from u to v and A[u,v]=1 if the edge from u to v is present.

Adjacency List: Collection of linked lists, one for each vertex that contains all the vertices adjacent to the particular vertex.

Weighted graph is a graph with numbers assigned its edges. These numbers are called as weights. If this graph is represented by adjacency matrix A[u,v]=weight of the edge [u,v] and ∞ if there is no edge. Such a matrix is called as weight matrix or cost matrix.

A path from vertex u to vertex v is sequence of adjacent vertices starts with u and ends with v. length of the path is total number of vertices in a vertex sequence defining the path minus 1. A directed path is a sequence of vertices in which every consecutive pair of the vertices is connected by a directed edge. A graph is said to be connected if for every pair of its vertices u and v there is a path from u to v. a graph with no cycles is said to be acyclic.

Trees:

A tree is a connected acyclic graph.
|E|=|V|-1 no of edges is no of vertices -1
Rooted trees: for every two vertices in a tree there exists one simple path from one of these vertices to the other. Select an arbitrary vertex in a free tree and it is root.
Ancestors: for any vertex v in a tree, all the vertices on the simple path from the root to that vertex are called ancestors of v.

Parent & child: if (u,v) is the last edge of the simple path from the root to vertex v, u is parent and v is child.

Siblings: vertices that have same parent

Unit I Content Beyond Syllabus

Important Data Structures

Leaf: vertices with no children

Descendants: all the vertices for which a vertex v is an ancestor are said to be descendants of v

Subtree: vertex v with all its descendants is called the subtree.

Depth: length of the simple path from the root to v

Height: length of the longest simple path from root to a leaf.

Ordered trees: rooted tree in which all the children of each vertex are ordered.

Binary tree: ordered tree in which each vertex has no more than two children. Each child is designated as left and right child.

Binary search tree: A number assigned to each parental vertex is larger than the numbers in its left subtree and smaller than its right subtree.

Sets & Dictionaries:

Set: unordered collection of distinct items called elements of the set.

Bit Vector: i^{th} element is 1 iff i^{th} element of U is included in set S.

Dictionary: ordered collection of words. Operations are: searching, adding, and deleting an item.

Assessment Schedule (Proposed Date & Actual Date)

R.M.K GROUP OF INSTITUTIONS

Assessment Schedule

Assessment Tool	Proposed Date	Actual Date	Course Outcome	Program Outcome (Filled Gap)
Assessment I			CO1, CO2	
Assessment II			CO3, CO4	
Model			CO1, CO2, CO3, CO4, CO5	
End Semester Examination			CO1, CO2, CO3, CO4, CO5	

Prescribed Text Books & Reference

Prescribed Text & Reference Books

TEXT BOOK:

1. Ganesh Prasad Kumble, Anantha Krishnan, "Practical Artificial Intelligence and Blockchain: A guide to converging blockchain and AI to build smart applications for new economies", Packet Publications, 2020.
2. Melanie Swan, "Block Chain: Blueprint for a New Economy", O'Reilly, 2015

REFERENCES:

1. Daniel Drescher, "Block Chain Basics", Apress; 1st edition, 2017

Mini Project Suggestions

R.M.K
GROUP OF
INSTITUTIONS

Mini Project

Amazon Use Case:

1. Drones and robots have come into existence which are automated and are used to deliver Amazon packages to a specific location. Since the source and destination is known, the drone should move in the ordered direction by following the shortest path to keep delivering the package in a minimum amount of time. Write algorithm to direct drone in shortest path to deliver Amazon package in minimum amount of time.

Other Mini Projects

1. Music Player – Songs in music player are linked to previous and next song. You can play songs from either starting or ending of the list.
   Write an algorithm for the above problem and analyse the efficiency of the algorithm .
2. File System: It is highly useful in file system handling where for example the file allocation table contains a sequential list of locations where the files is split up and stored on a disk. Remember that overtime it is hard for an OS to find disk space to cover the entire file so it usually splits these up into chunks across the physical hard drive and stores a sequential list of links together as a linked list.
   Write an algorithm for the above problem and analyse the efficiency of the algorithm.

Thank you

Disclaimer: