Blockchain and Its Application

CS 740 (3-1-0) 4

Dr. Sourav Kanti Addya,

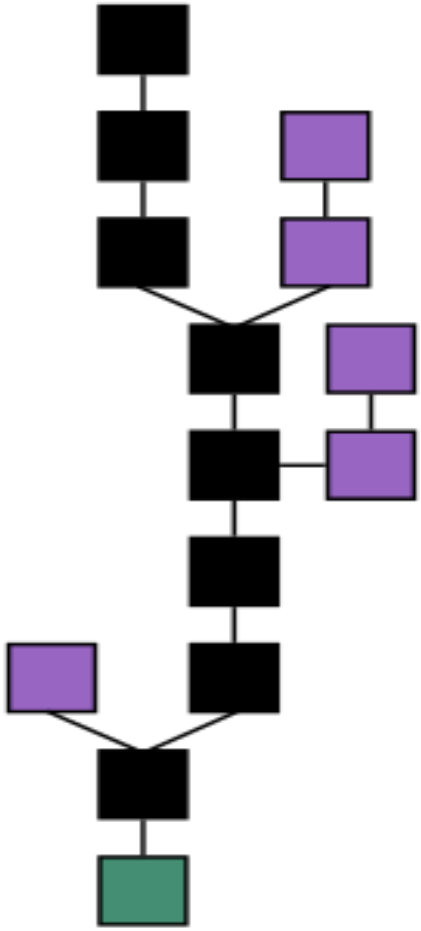
Dept of CSE, NIT Karnataka, Surathkal

- **What is Blockchain?**
- **Hashing Algorithm**
- **Immutable Ledger**
- **Distributed P2p Networks**
- **What is Mining?**
- **Consensus protocol**

Dr. Sourav Kanti Addya

What is Blockchain? (Definition 1)

- Growing list of records, called blocks, that are linked using cryptography



- Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data

Dr. Sourav Kanti Addya

What is Blockchain? (Definition 2)

- A decentralized computation and information sharing platform that enables multiple authoritative domains who do not trust each other, to cooperate, coordinate and collaborate in a rational decision making process.

Use Case

- Ms Word share document
  - Disadvantages: Can’t update at the same time.
- Google doc share document
  - They can edit at the same time
  - Disadvantages: Environment is centralized.
    - Single point failure
    - No Internet/ redundant information updating.

Centralized System

- Centralized
  - Single Coordinator
  - Fast (No need of consensus)
  - Problem: Single point of failure
  - Example: Google Doc

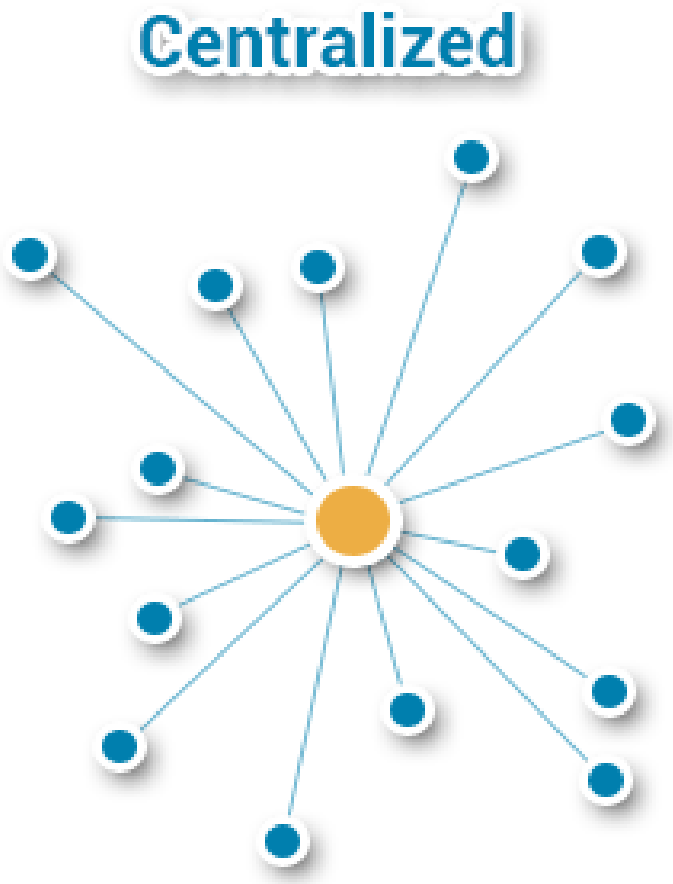


Image Source: https://mutualwelfare.org/centralized-decentralized-and-distributed-from-the-pyramid-to-the-circle/

Decentralized System

- Decentralized
  - Multiple Coordinator
  - Slow (Need common consensus among the coordinators)
  - Advantage: If a coordinator fails, there is another to provide service

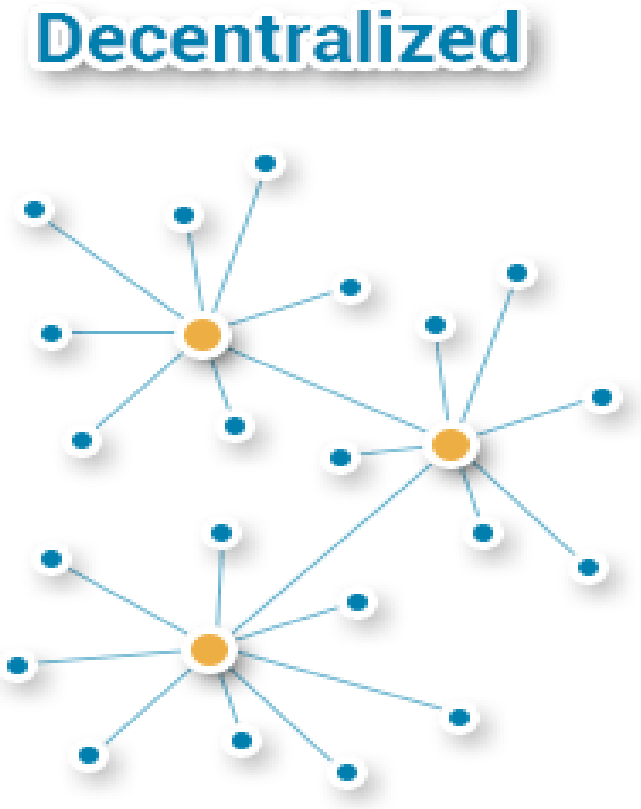


Image Source: https://mutualwelfare.org/centralized-decentralized-and-distributed-from-the-pyramid-to-the-circle/

Distributed System

- Distributed
  - Coordination task is ditributed
  - Advantage: Coordination does not depend on a single point

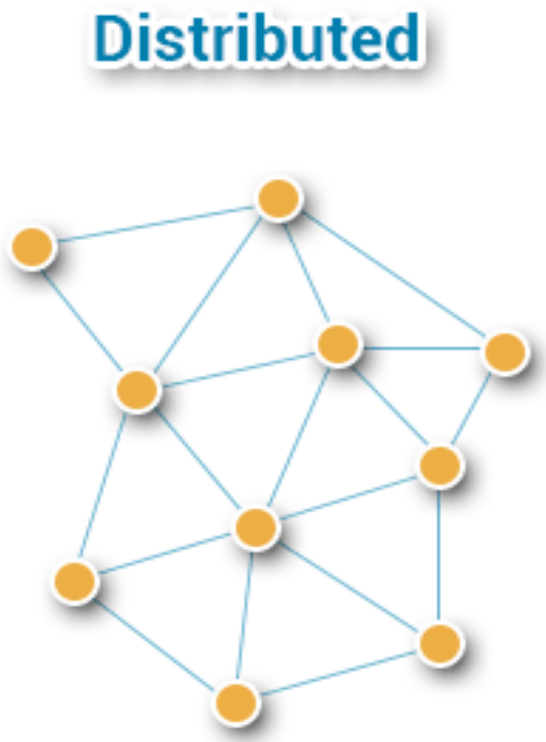


Image Source: https://mutualwelfare.org/centralized-decentralized-and-distributed-from-the-pyramid-to-the-circle/

Dr. Sourav Kanti Addya

Blockchain, a decentralized approach

- Every node will edit and maintain their local copy of the global data sheet
- The system will ensure consistency among the local copies
- Local copies are identical
- Local copies are updated based on global information

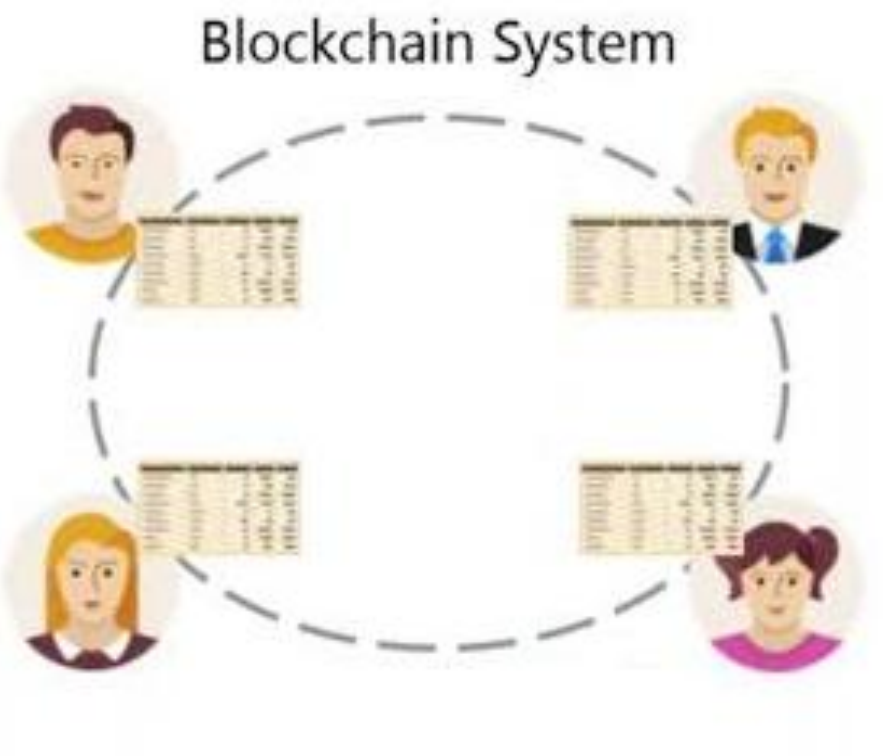


Image Source: https://msdynamicsworld.com/story/video-review-using-blockchain-supply-chain-compliance

Public Ledger

- Maintains the history of all the activities
- Helpful to validate a new activity based on the previous activities in the history

- Example:
  - Course registration
  - Banking transaction

Blockchain and Public Ledger

- Blockchain works like a public ledger
- Different Aspects
  - Protocols for commitment: Ensure that every valid transaction from the client committed and included in the blockchain within a finite time.
  - Consensus: Ensure that the local copies are consistent and updated.
  - Security: The data needs to be tampered proof. Note that the clients may act maliciously or can be compromised.
  - Privacy and Authenticity: The data (or transactions) belong to various clients; privacy and authenticity needs to be ensured.

What is Blockchain? (Formal Definition)

- A Blockchain is "an open, distributed ledger that can record transaction between two parties efficiently and in a verifiable and permanent way" (Iansiti, Lakhani 2017).

- Keywords: **Open** (accessible to all), **Distributed or decentralized** (no single party control), **efficient** (fast and scalable), **verifiable** (everyone can check the validity of information), **permanent** (the information is persistent).

Iansiti, Maco; Lakhani, Karim R. (Jan 2017), "The truth about Blockchain", Harvard Business review, Harvard University.

Blockchain based Cryptocurrency

- Traditional digital currency system works through centralized system
  - Easy to attack by governments and hackers
  - Can suffer from single point of failure problem

- Blockchain based cryptocurrency evolved, which is free of any central authority or point of control that can be attacked or corrupted
- Example: Bitcoin network

Text and Reference books

- Texts:
  - Roger Wattenhofer, The Science of the Blockchain, Inverted Forest Publishing, First Edition, 2016.

- References:
  - Don Tapscott, Alex Tapscott, Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies is Changing the World, Portfolio Penguin, 2018.
  - Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Bitcoin and Cryptocurrency Technologies – A Comprehensive Introduction, Princeton University Press, 2016.
  - Andreas M. Antonopoulos, Mastering Bitcoin: Programming The Open Blockchain, Shroff/O'Reilly, Second edition, 2017.

Cryptographic Hash Function

- Can be applied to any sized message *M*
- Produces fixed-length output *h* (256 bit used for Blockchain)
- Easy to compute $h=H(M)$ for any message *M*

Three security Properties

Collision Free

Hiding

Puzzle Friendly

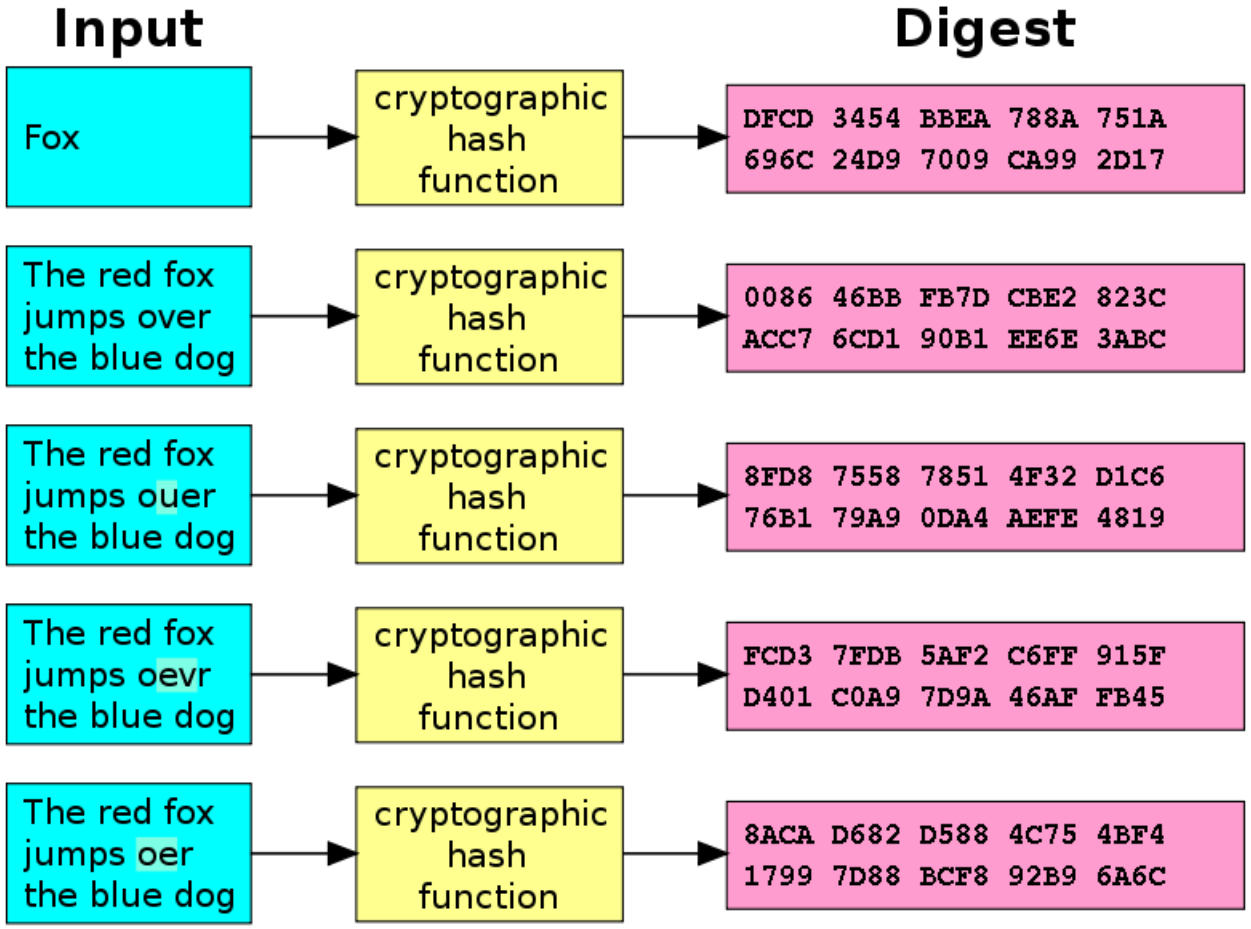
Cryptographic Hash Function

- Ex: **H(x)=x%n** where x, n are integers % is the modular operations. x can be any arbitrary range but H(x) is with in [0,n-1]
- Cryptographically Secured:
  - Oneway: given a **x** we can compute **H(x)** but given an **H(x)** no deterministic algorithm to compute x.
  - For two different value of **x**, i.e. **x1** and **x2** the **H(x1)** and **H(x2)** should be different.

Avalanche Effect

- Example: MD5, SHA256
- **X** is called the message and **H(X)** is called the message digest.

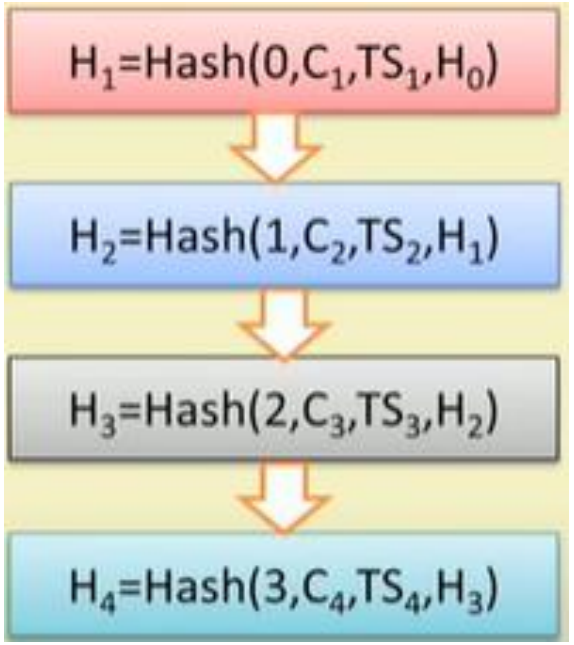
- A small change in the data results in a significant change in the output – called **Avalanche effect**



Dr. Sourav Kanti Addya

Cryptographically Secured Chain of Blocks

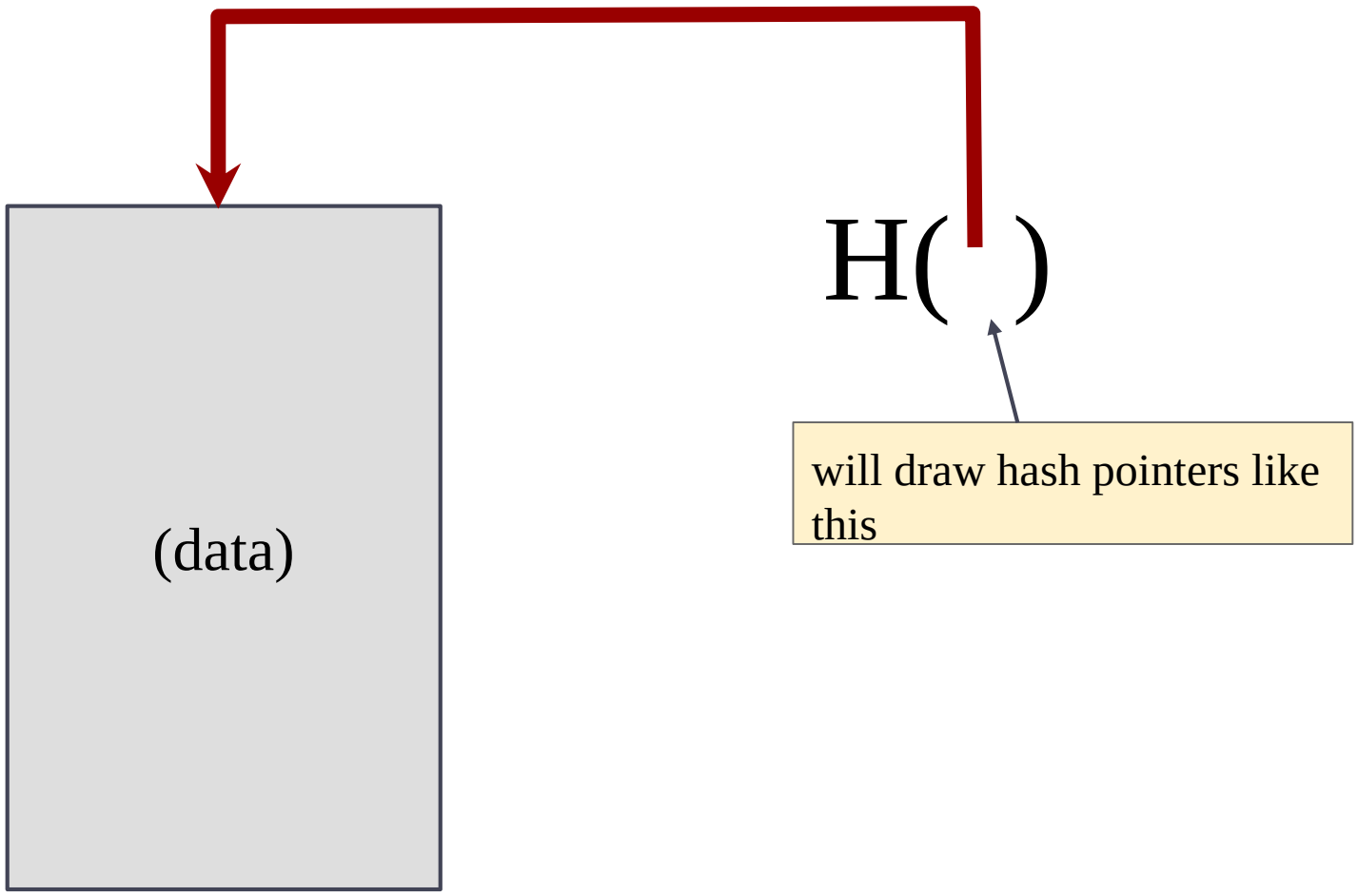
- **Harber** and **Stornetta** first proposed "How to ***time-stamp a digital document***" in 1991
- A new block is created and time stamped whenever a new client edit the document
- A Block consists of Seq-of-access, Client-id, Time-stamp and hash value of the previous block



Hash Pointer

- Point to where some info is stored, and (cryptographic) hash of the info

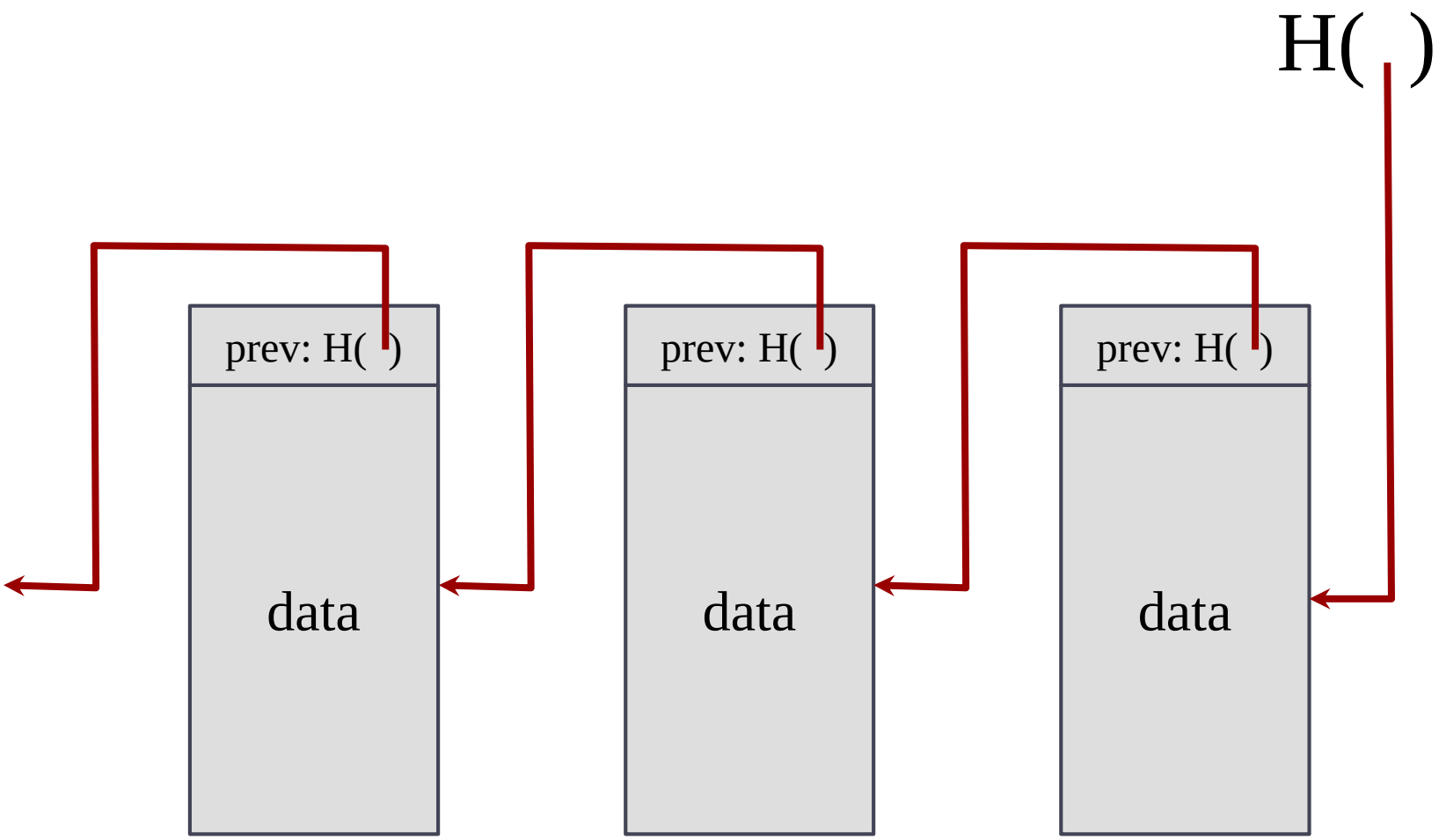
- If we have a hash pointer, we can
  - Ask to get the info back, and
  - Verify that it hasn't changed


H( )
(data)
will draw hash pointers like this

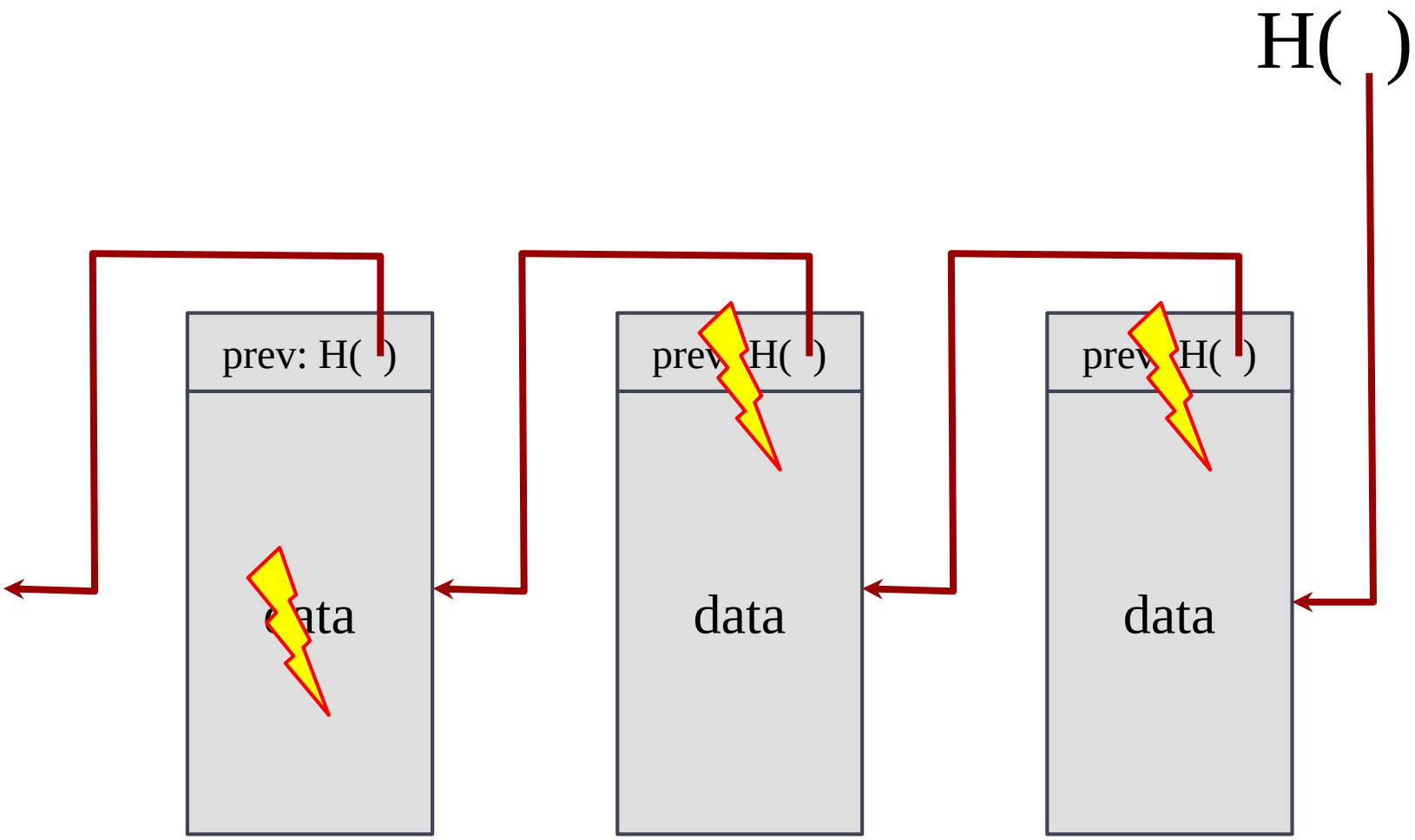
Key Idea:

Build data structures with hash pointers

Linked list with hash pointers

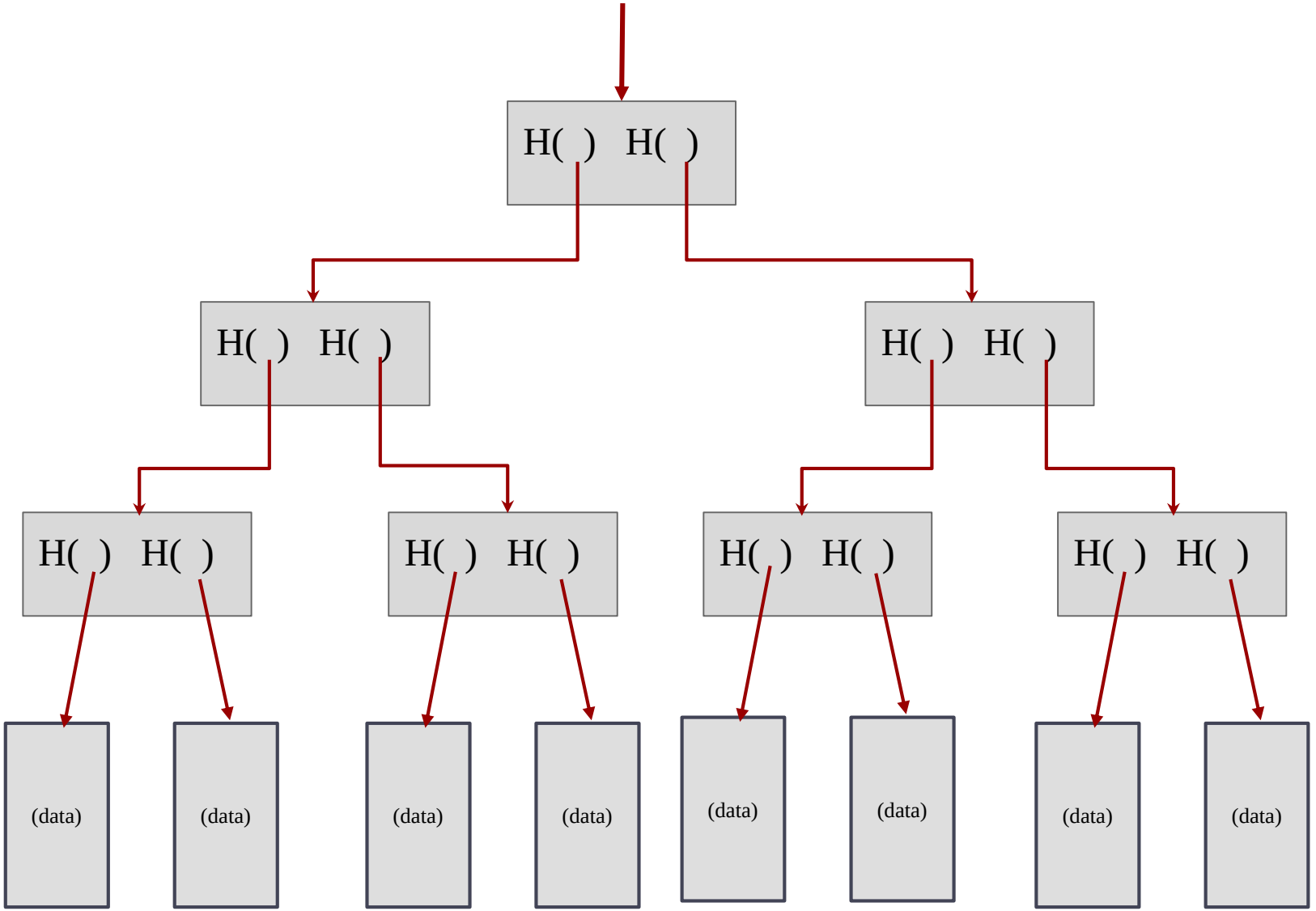


Detecting tampering



use case: tamper-evident log

Binary tree with hash pointers = “Merkle tree” (Ralph Markle, 1979)/ Hash Tree



- Every leaf node is labelled with the hash of a data block
- Every non leaf node is labelled with hash of the labels of its child nodes.

Data Verification in Merkle tree

- Assume we have root hash
- We download some chunk of data from the untrusted network
- We ask the server to provide the proof that this chunk is in the tree
- The server returns the appropriate hashes
- Using this information, you compute the root hash and verify it against the root hash with which you accessed the file.

Example

Peer wants to verify that "Y" chunk exists in the file and is untampered

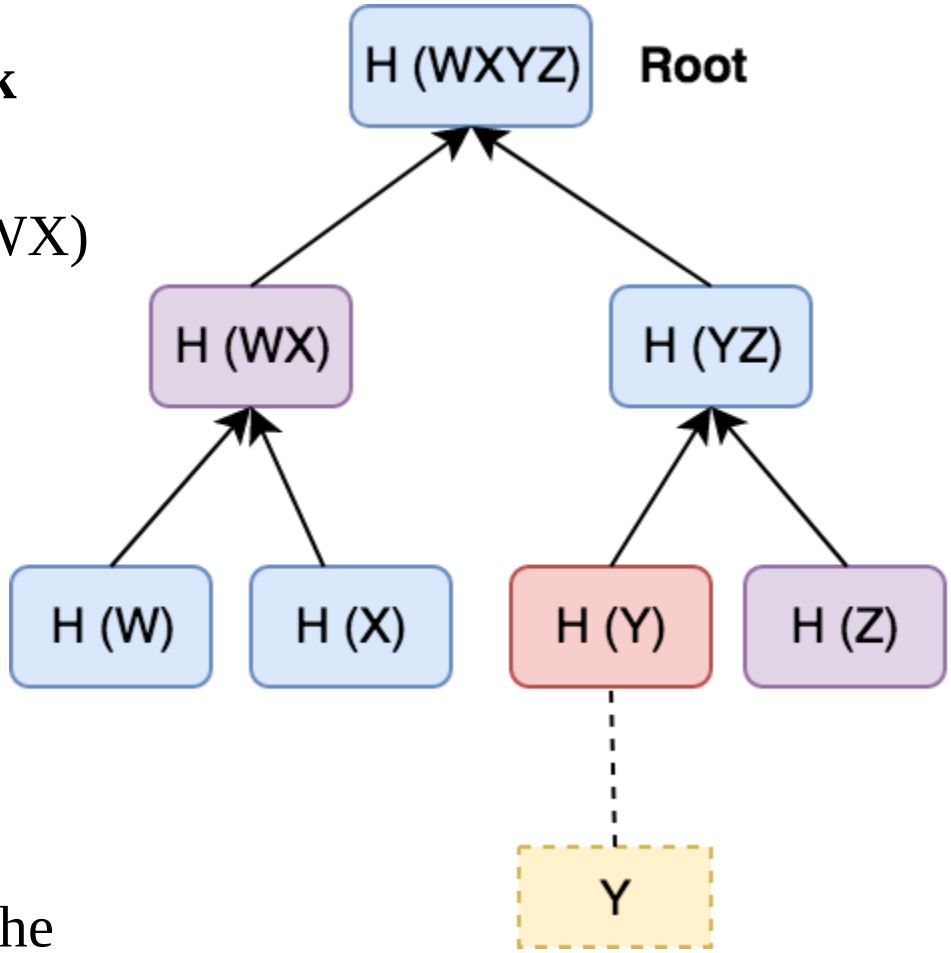
Server returns the info H (Z) and H (WX)

We can compute:

H (YZ) from H (Y), H (Z)

H (WXYZ) from H (YZ), H (WX)

Compare the root hash we have with the computed root hash H (WXYZ)



In case of Failure

We can chuck this peer and ask for the same chunk from another peer who has the file that we're looking for.

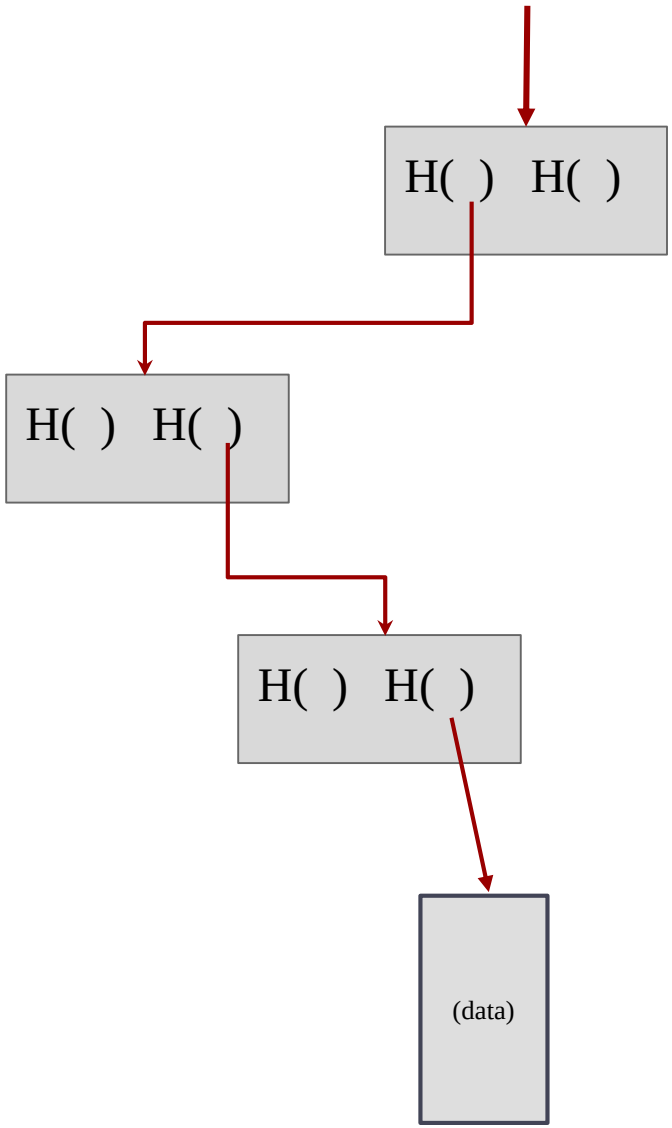
This process is also known as audit proof.

Good Part

- A very little information was required from the trusted server to verify the data. If the number of data chunks are doubled, the additional information required for verification would be just two more hashes, and the verification on the client side would require two more hash computations.

- Since the size of data verification packet is small, it helps us to save bandwidth.

Time Complexity



Only O(log n) items

Use of Merkle trees

Bayer, Harber and Stornetta used Markle Tree in 1992 to improve the efficiency by combining timestamping of several documents into one block

- Tree holds many items but just need to remember the root hash
- Can verify membership in O(log n) time/space
- Variant: sorted Merkle tree
  - can verify non-membership in O(log n)
  - (show items before, after the missing one)

Inventor of Bitcoin

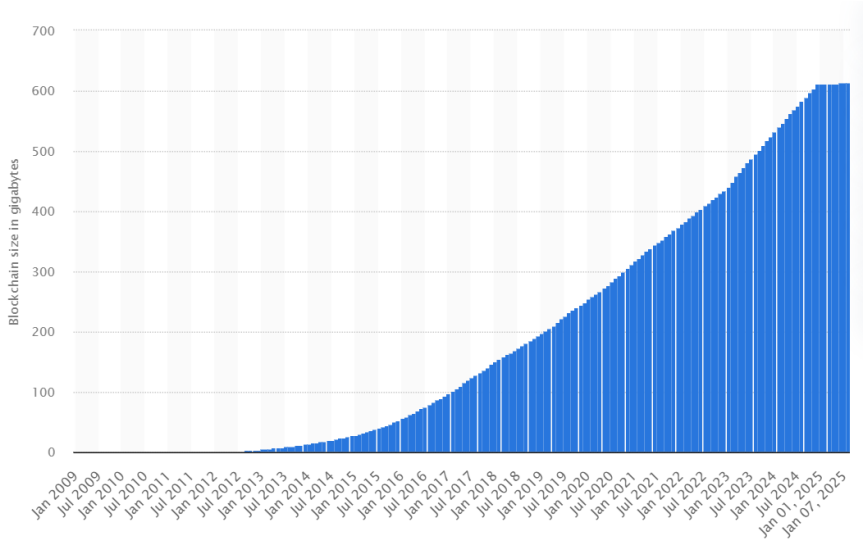
- Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008 (https://bitcoin.org/bitcoin.pdf)
- **Satoshi Nakamoto** is a pseudonym for the person or people who helped develop the first bitcoin software and introduced the concept of cryptocurrency to the world in a 2008 paper

Bitcoin

- Bitcoin is a completely decentralized, peer-to-peer, permissionless cryptocurrency put forth in 2009.
  - Completely decentralized: No central party for ordering or recording anything.
  - Peer-to-peer: Software that runs on machines of all stakeholders to form the system.
  - Permissionless: no identity; no need to signup anywhere to use; no access control – anyone can participate in any role.

Bitcoin

- Bitcoin
- Used as a cross country, untraceable currency which is not under the control of any Govt. and hence free from regulation
- Current BTC price 1 BTC = $99139.33 / INR 8580817.24 (as of Jan 16, 2025 @ 15.35 pm)
- The Bitcoin blockchain size as of Jan 15, 2025 is ~630.03 GB.



Bitcoin Transaction Life Cycle

- Alice opens her Bitcoin Wallet
- Provides the address of Bob and the amount to transfer, and sends
- The wallet constructs the transactions, sign using Alices' private key, and broadcasts it to the network.
- The network nodes validate the transactions based on the existing blockchain, and propagate the transaction to the **miners**.
- The **miners** include the transaction to the next block to be **mined**.
- Miners collect all the transaction fo the say 10 min.
- Miners construct a new block- The mining procedure
- The updated Blockchain propagated in the network.
- Bob also received the updated blockchain.

Types of Blockchains

- Public (Permission-less)
- Private (Permissioned)
- Consortium
- Hybrid

Public Blockchain (Permission-less)

- A non-restrictive, permission-less distributed ledger system
- A node or user which is a part of the public blockchain is authorized to *access current and past records, verify transactions or do proof-of-work for an incoming block, and do mining*
- Public blockchains are mostly secure if the users strictly follow security rules and methods
- However, it is only risky when the participants don't follow the security protocols sincerely.
  - **Example:** Bitcoin, Ethereum, Litecoin

Advantages and Disadvantages

- Trustable: participants do not need to worry about the authenticity of the other
- Secure: Harder it is for hackers to hack the entire network
- Open and Transparent: No one shows a fake transaction or hides an existing one as every node has an updated copy of the database at any given point of time

- Lower TPS: Huge network with a lot of nodes
- Scalability Issues: Increase of nodes will decrease the speed

Private Blockchain (Permissioned)

- Restrictive and operative only in a closed network
- Used within an organization or enterprises where only selected members are participants of a blockchain network
- Similar in use as a public blockchain but have a small and restrictive network
- Valuable for enterprises who want to collaborate and share data, but don’t want their sensitive business data visible on a public blockchain

  - **Examples:** Hyperledger projects (Fabric, Sawtooth), Corda, etc.

Advantages and Disadvantages

- Speed: Limited number of nodes fastens the consensus or verification process
- Scalability: Flexibility to increase or decrease the size of the network

- Needs Trust-building: Need to build trust to transmit confidential information within a network
- Lower Security: If anyone of the nodes gains access to the central management system

Consortium Blockchain

- Collaboration of a group of entities
- Governed by a group rather than a single entity
- Offers some of the best use cases for the benefits of blockchain, bringing together a group of “frenemies”- businesses who work together but also compete against each other

Hybrid Blockchain

- A combination of the private and public blockchain
- Only a selected section of data or records from the blockchain can be allowed to go public keeping the rest as confidential
- A transaction in a private network of a hybrid blockchain is usually verified within that network
- But users can also release it in the public blockchain to get verified.
- Increase the hashing and involve more nodes for verification. This enhances the security and transparency of the blockchain network.
  - **Example** of a hybrid blockchain is Dragonchain.

Public Ledger (A Small Example)

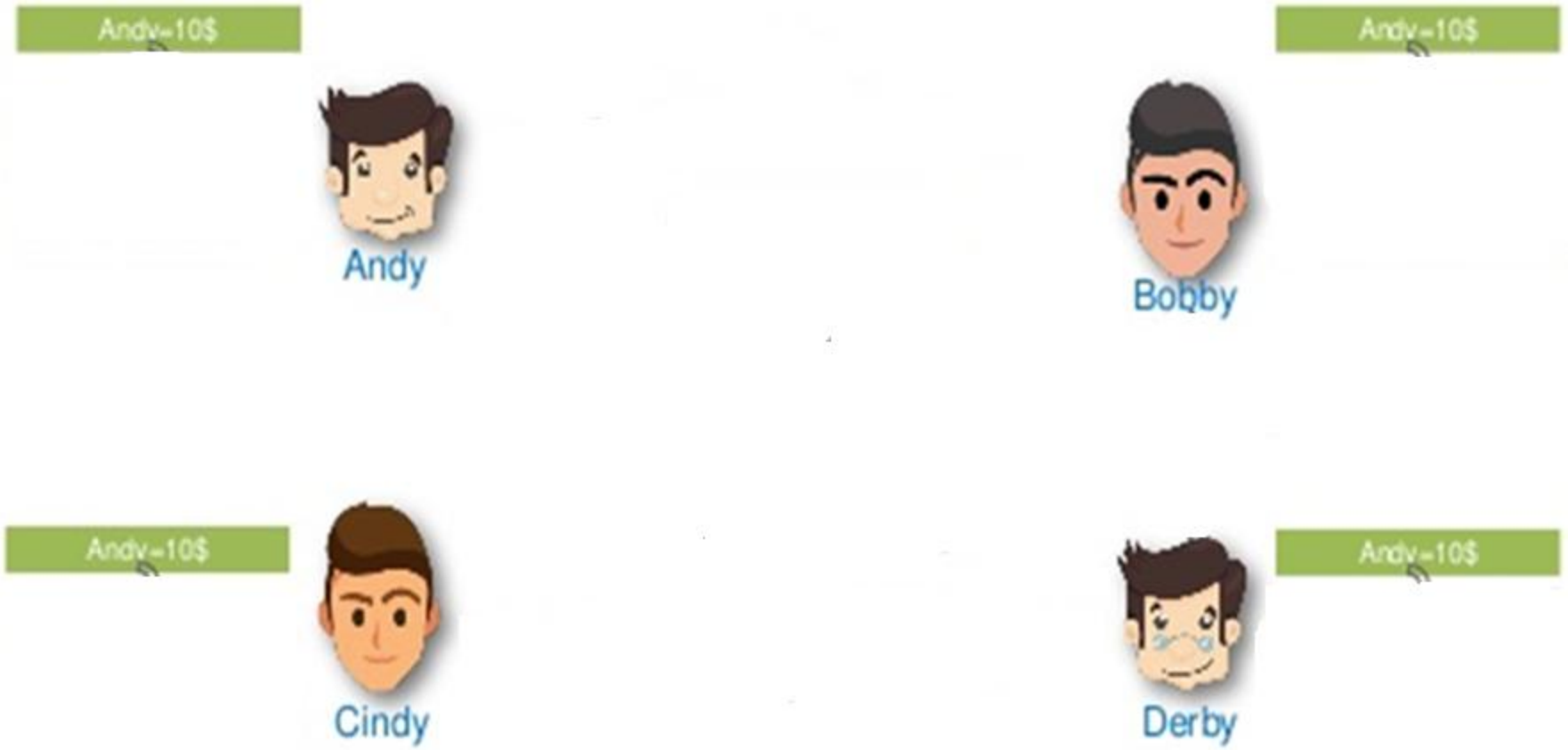


Image Source: https://www.slideshare.net/EdurekaIN/blockchain-tutorial-getting-started-with-blockchain-blockchain-certification-training-edureka

Public Ledger (A Small Example)

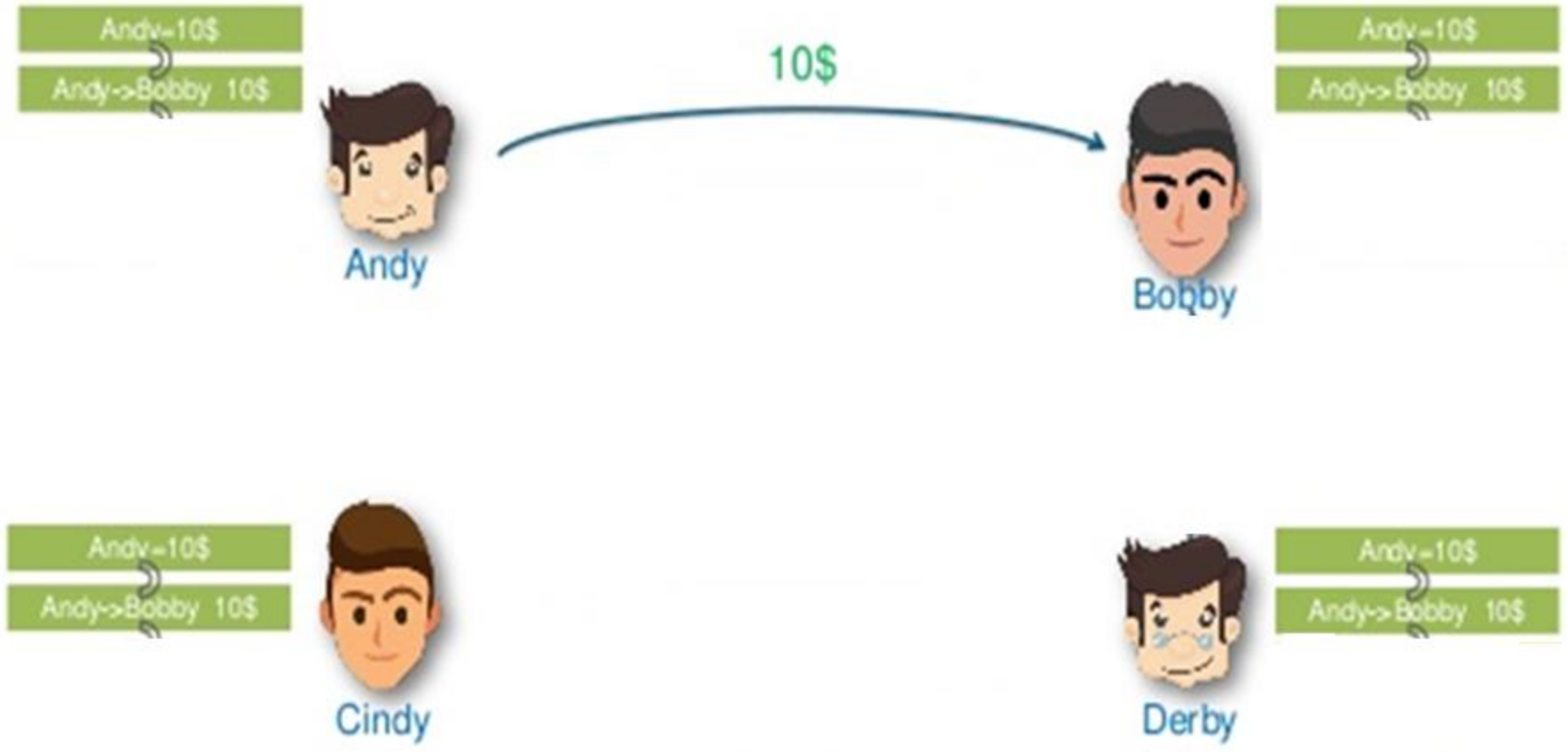


Image Source: https://www.slideshare.net/EdurekaIN/blockchain-tutorial-getting-started-with-blockchain-blockchain-certification-training-edureka

Public Ledger (A Small Example)

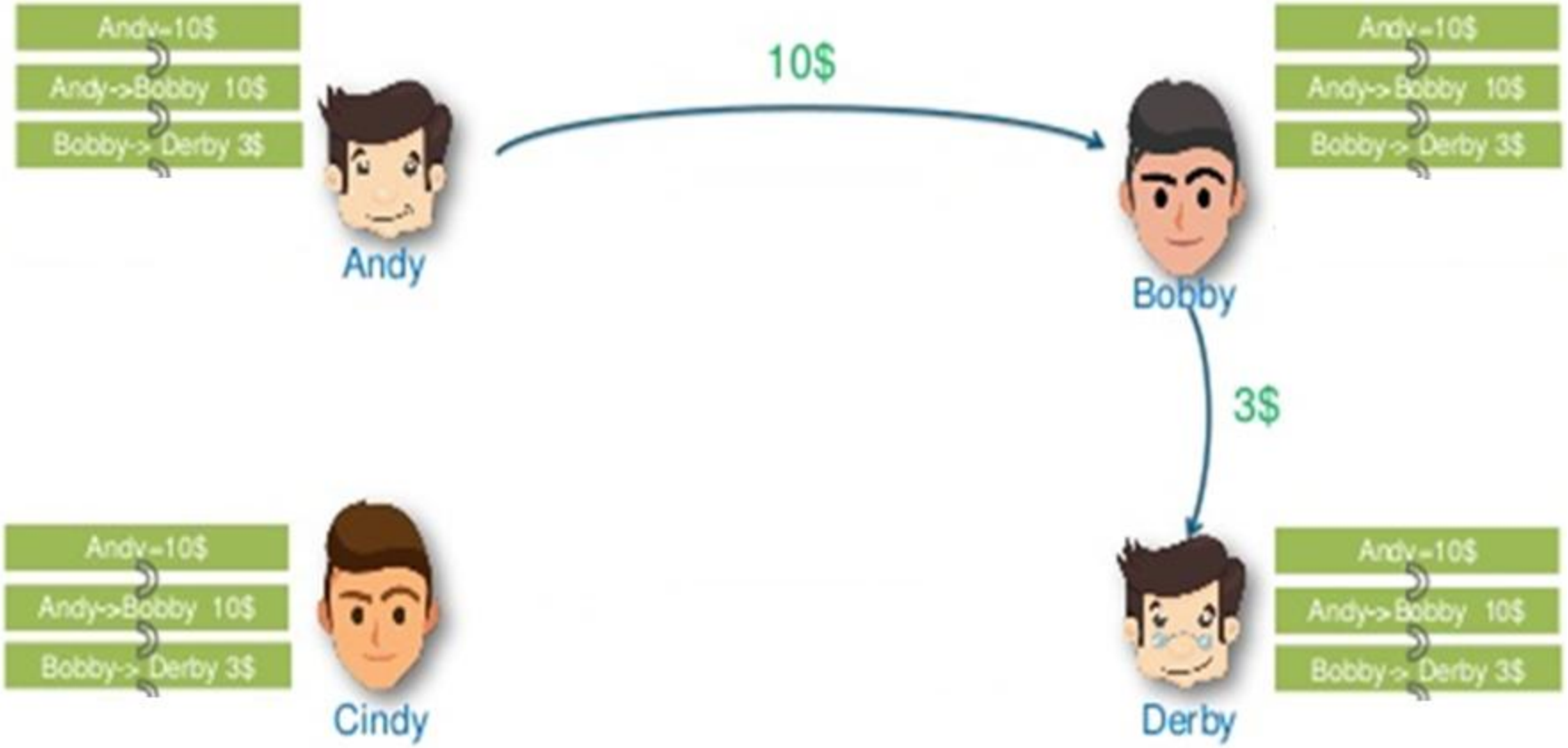


Image Source: https://www.slideshare.net/EdurekaIN/blockchain-tutorial-getting-started-with-blockchain-blockchain-certification-training-edureka

Public Ledger (A Small Example)

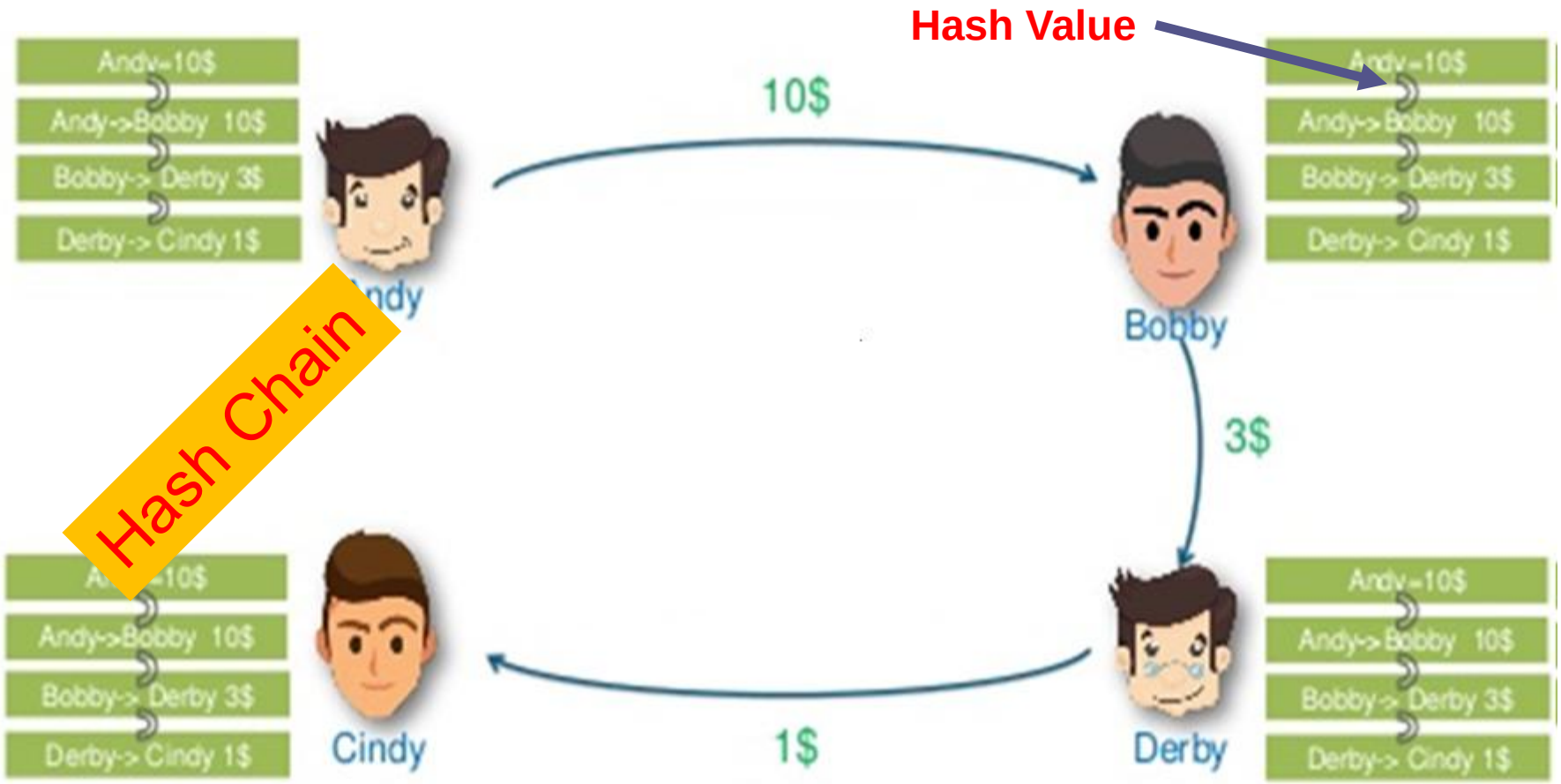


Image Source: https://www.slideshare.net/EdurekaIN/blockchain-tutorial-getting-started-with-blockchain-blockchain-certification-training-edureka

How Blockchain Works?

Dr. Sourav Kanti Addya

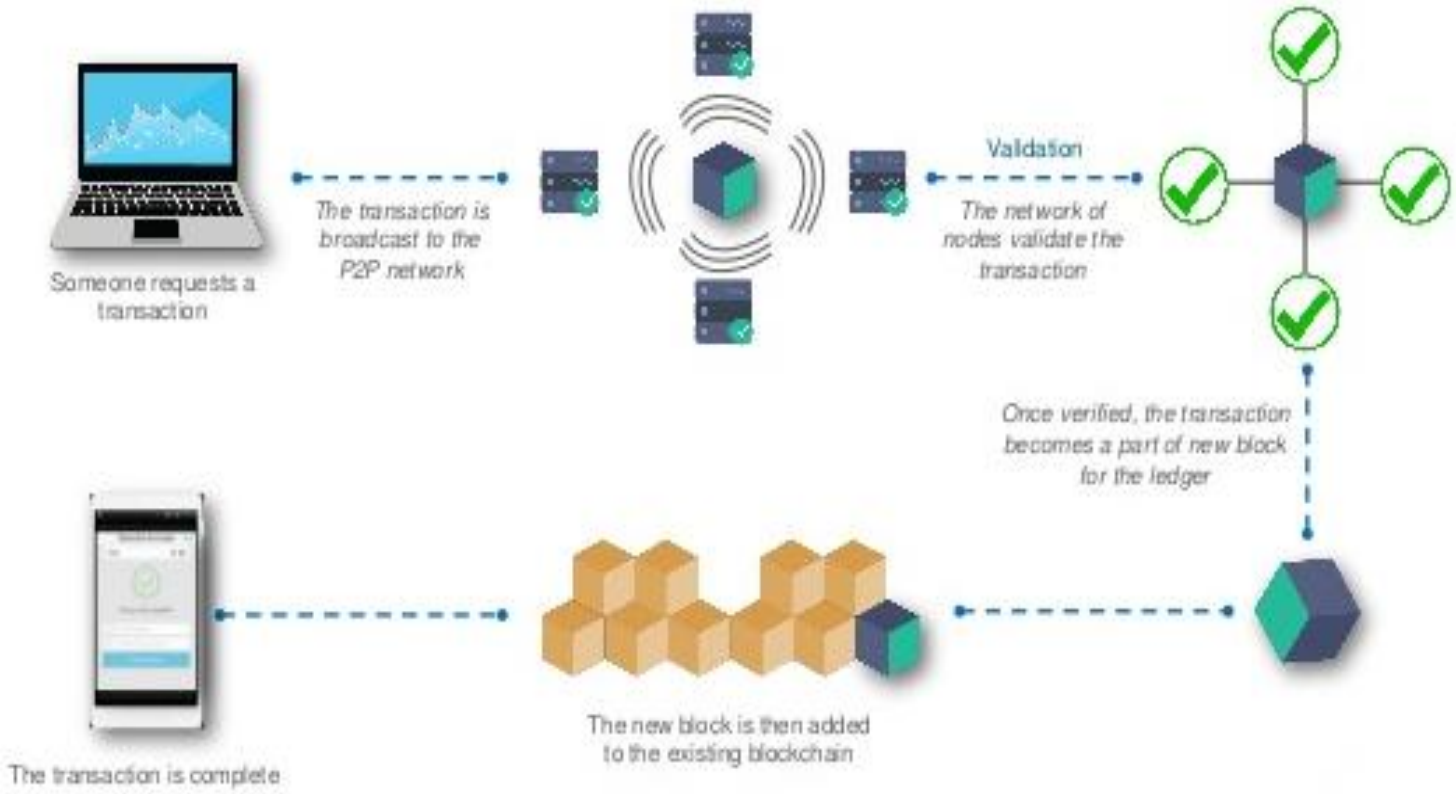


Image Source: https://www.slideshare.net/EdurekaIN/blockchain-tutorial-getting-started-with-blockchain-blockchain-certification-training-edureka

Aspects of decentralization in Blockchain

- How the ledger is maintained?
- How the validity of transaction is decided?
- How the new blocks are created?

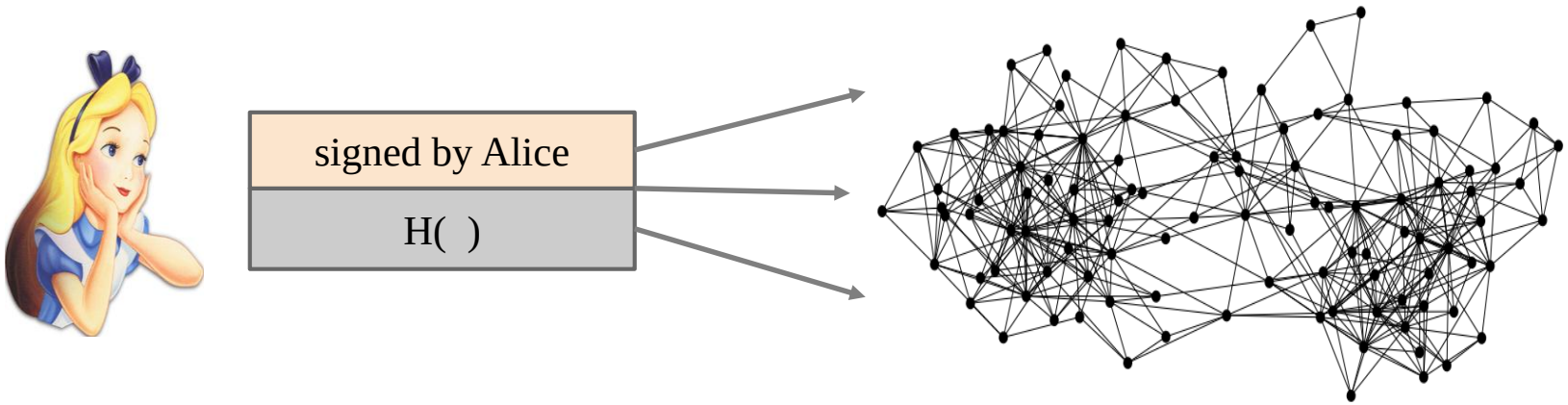
Distributed Consensus

- The protocol terminates and all correct nodes decide on the same value

- This value must have been proposed by some correct node

Blockchain is a peer-to-peer system

- When Alice wants to perform an action
- she broadcasts the action information as a transaction to all nodes in the networks



Smart Contracts in Blockchain

A **Smart Contract** (or cryptocontract) is a computer program that *directly* and *automatically* controls the transfer of digital assets between the parties under certain conditions.

A **smart contract** works in the same way as a traditional contract while also automatically enforcing the contract. Smart contracts are programs that execute exactly as they are set up(coded, programmed) by their creators. Just like a traditional contract is enforceable by law, smart contracts are enforceable by code.

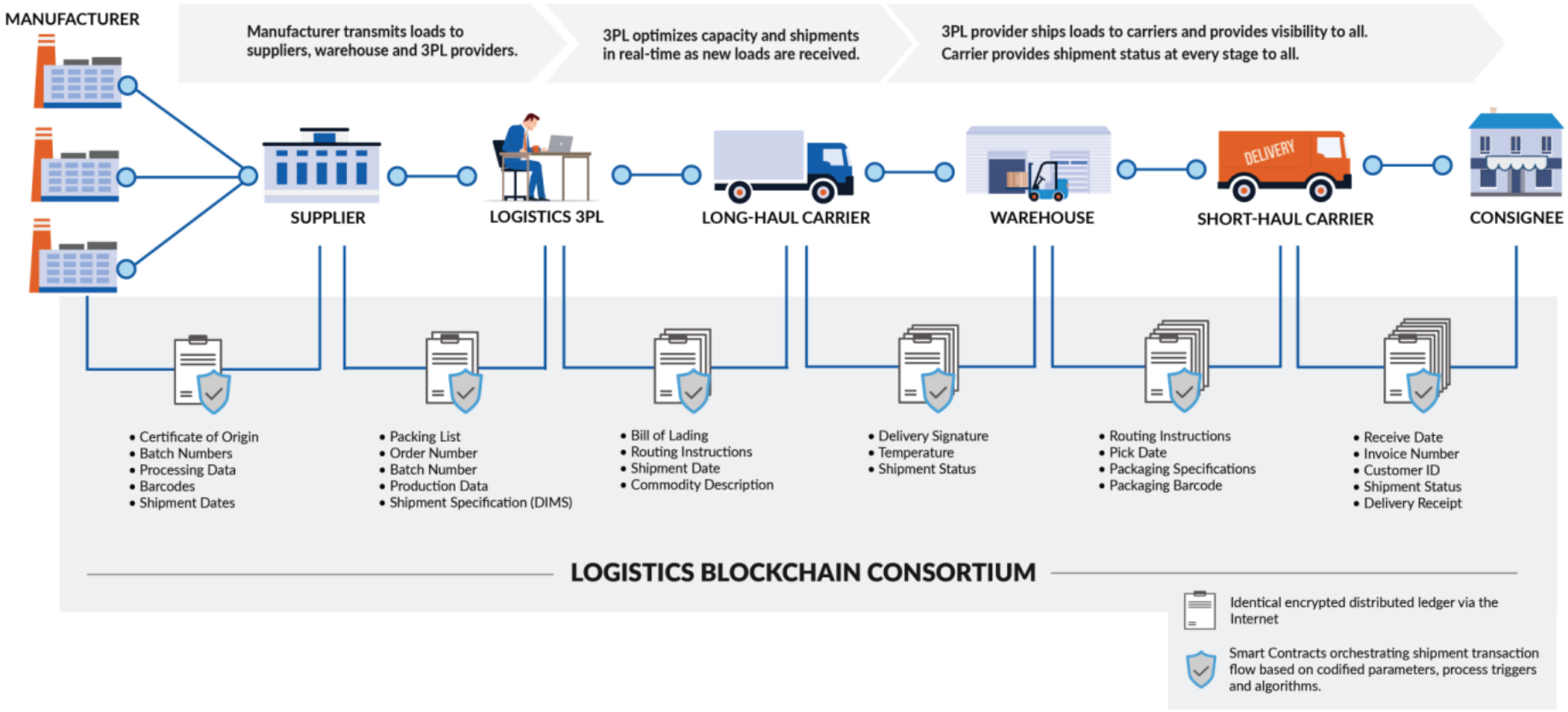
History:

In 1994, **Nick Szabo**, a legal scholar, and a cryptographer recognized the application of a decentralized ledger for smart contracts. He theorized that these contracts could be written in code which can be stored and replicated on the system and supervised by the network of computers that constitute the blockchain. These smart contracts could also help in transferring digital assets between the parties under certain conditions.

- The **bitcoin network** was the first to use some sort of smart contract by using them to transfer value from one person to another.
- The smart contract involved employs basic conditions like checking if the amount of value to transfer is actually available in the sender account.

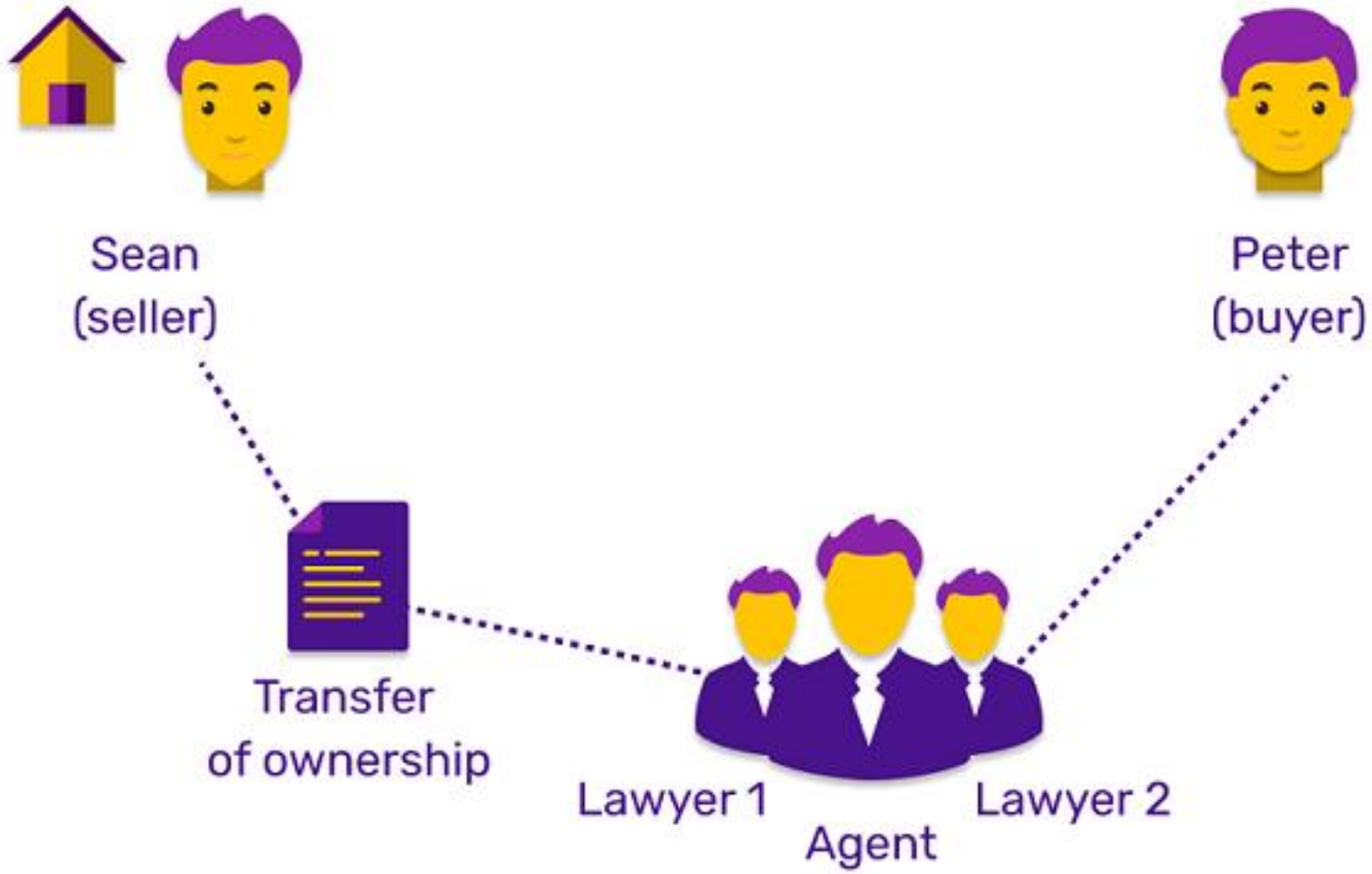
- Later, the **Ethereum platform** emerged which was considered more powerful, precisely because the developers/programmers could make custom contracts in a Turing-complete language.
- It is to be noted that the contracts written in the case of the bitcoin network were written in a Turing-incomplete language, restricting the potential of smart contracts implementation in the bitcoin network.
- There are some common smart contract platforms like Ethereum, Solana, Polkadot, Hyperledger fabric, etc.

International Carriage and Sales of Goods



Source of the picture: https://www.globaltranz.com/blog/blockchain-technology-transform-logistics/

Traditional Contract



In 1994, Nick Szabo (a cryptographer), came up with the idea of being able to record contracts in the form of computer code

Image Source: https://blockspoint.com/guides/ethereum/what-is-a-smart-contract

Crowdfunding

1. You have an interest in a project but unable to execute du to less findings.
2. Submit your proposal to in a crowdfunding platform.
3. The platform ensures that you will get complete money if your project will successful.
4. Multiple supporters supports the project with small findings.

Trust the platform, commission to the middleman.

Smart Contracts in Blockchain

A **Smart Contract** (or cryptocontract) is a computer program that *directly* and *automatically* controls the transfer of digital assets between the parties under certain conditions.

A **smart contract** works in the same way as a traditional contract while also automatically enforcing the contract. Smart contracts are programs that execute exactly as they are set up(coded, programmed) by their creators. Just like a traditional contract is enforceable by law, smart contracts are enforceable by code.

History:

In 1994, **Nick Szabo**, a legal scholar, and a cryptographer recognized the application of a decentralized ledger for smart contracts. He theorized that these contracts could be written in code which can be stored and replicated on the system and supervised by the network of computers that constitute the blockchain. These smart contracts could also help in transferring digital assets between the parties under certain conditions.

- The **bitcoin network** was the first to use some sort of smart contract by using them to transfer value from one person to another.
- The smart contract involved employs basic conditions like checking if the amount of value to transfer is actually available in the sender account.

- Later, the **Ethereum platform** emerged which was considered more powerful, precisely because the developers/programmers could make custom contracts in a Turing-complete language.
- It is to be noted that the contracts written in the case of the bitcoin network were written in a Turing-incomplete language, restricting the potential of smart contracts implementation in the bitcoin network.
- There are some common smart contract platforms like Ethereum, Solana, Polkadot, Hyperledger fabric, etc.

Smart Contract

A program that runs within a blockchain

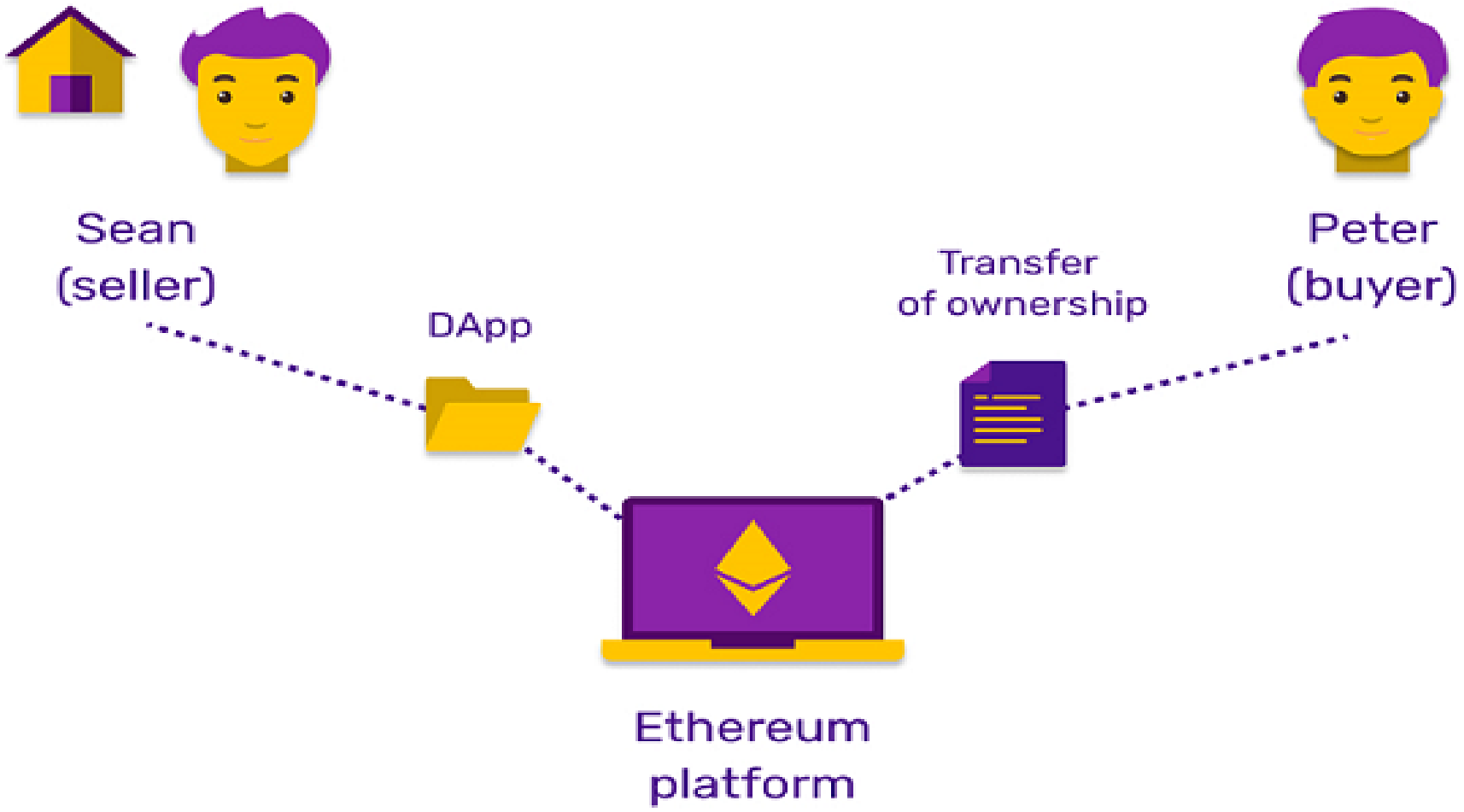
Contains a set of rules that constitute an agreement made between two or more parties

When these rules are met, the digital contract executes the transaction

It’s like a regular application that implements some business rules, only it uses a blockchain as a database

Source : https://steelkiwi.com/blog/making-effective-use-of-smart-contracts/

Smart Contract - Example



WHEN Peter pays Sean 300 Ether, THEN Peter will receive ownership of the house

Image Source: https://blockspoint.com/guides/ethereum/what-is-a-smart-contract

Smart Contract - Characteristics

Self-verifying due to automated possibilities

Self-enforcing when the rules are met at all stages

Tamper-proof, as no one can change what’s been programmed

Source : https://steelkiwi.com/blog/making-effective-use-of-smart-contracts/

Smart Contract - Capabilities

Automate processes done manually

Ensure security

Reduce relation to trusted intermediaries

Support multi-signature accounts to distribute funds as soon as all parties involved confirm the agreement

Source : https://steelkiwi.com/blog/making-effective-use-of-smart-contracts/

Smart Contract - Capabilities

Manage users' agreements

Provide utility to other contracts (similar to how a software library works)

Store information about an app (domain registration information, membership records, etc.)

Source : https://steelkiwi.com/blog/making-effective-use-of-smart-contracts/

How do smart contracts work?

Identify Agreements

Multiple parties identify a cooperative opportunity and desired outcome

Agreement could include business processes, asset swaps, transfer of right and more

Source : https://steelkiwi.com/blog/making-effective-use-of-smart-contracts/

How do smart contracts work?

Set Conditions

Smart contract could be initiated by the parties themselves or by satisfaction of certain conditions like natural disasters, or event via GPS location

Source : https://steelkiwi.com/blog/making-effective-use-of-smart-contracts/

How do smart contracts work?

Code the business logic

A computer program is written in a way that the arrangement will automatically perform when the conditional parameters are met

Source : https://steelkiwi.com/blog/making-effective-use-of-smart-contracts/

How do smart contracts work?

Execution & Processing

In a blockchain iteration, when consensus is reached on authentication and verification, the smart contract is written to a block

The code is executed and memorialized for complince and verified

Source : https://steelkiwi.com/blog/making-effective-use-of-smart-contracts/

How do smart contracts work?

Network Updates

After execution of the smart contract, all computers in the network update their ledgers to reflect the new state

Once the record is verified and posted to the blockchain, it cannot be altered, it is append only

Source : https://steelkiwi.com/blog/making-effective-use-of-smart-contracts/

Integrity of the smart contract

Once the program is added into the blockchain, nobody can modify it

People can examine the source code to understand what exactly the program does

This code cannot be modified by hackers or viruses because smart contracts rely on blockchain and cryptography

Advantages

Direct business relationships: Removes the necessity of a third party

Availability: The network will still work as predetermined even if someone leaves it

Trustworthiness: There’s no way to dispute the conditions of a contract once it’s been established

Speed: Use code and live on the internet. This saves time for many business processes and eliminates the need to process documents manually

Advantages

Security: Smart contracts use the same level of security as a cryptocurrency. As of today, they're the safest way to store data on the web

Keeping records: All contracts are stored in chronological order and can be easily accessed when necessary

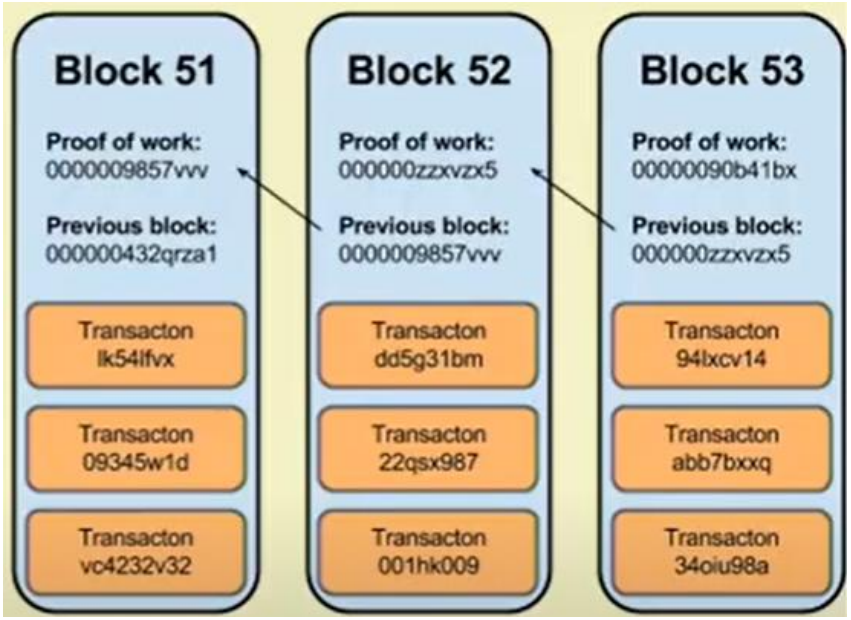
Paper-free: Contracts that are signed on paper can be lost, stolen, or destroyed, whereas smart contracts exist in lines of code in digital space

Smart Contract platform

HYPERLEDGER

Blockchain

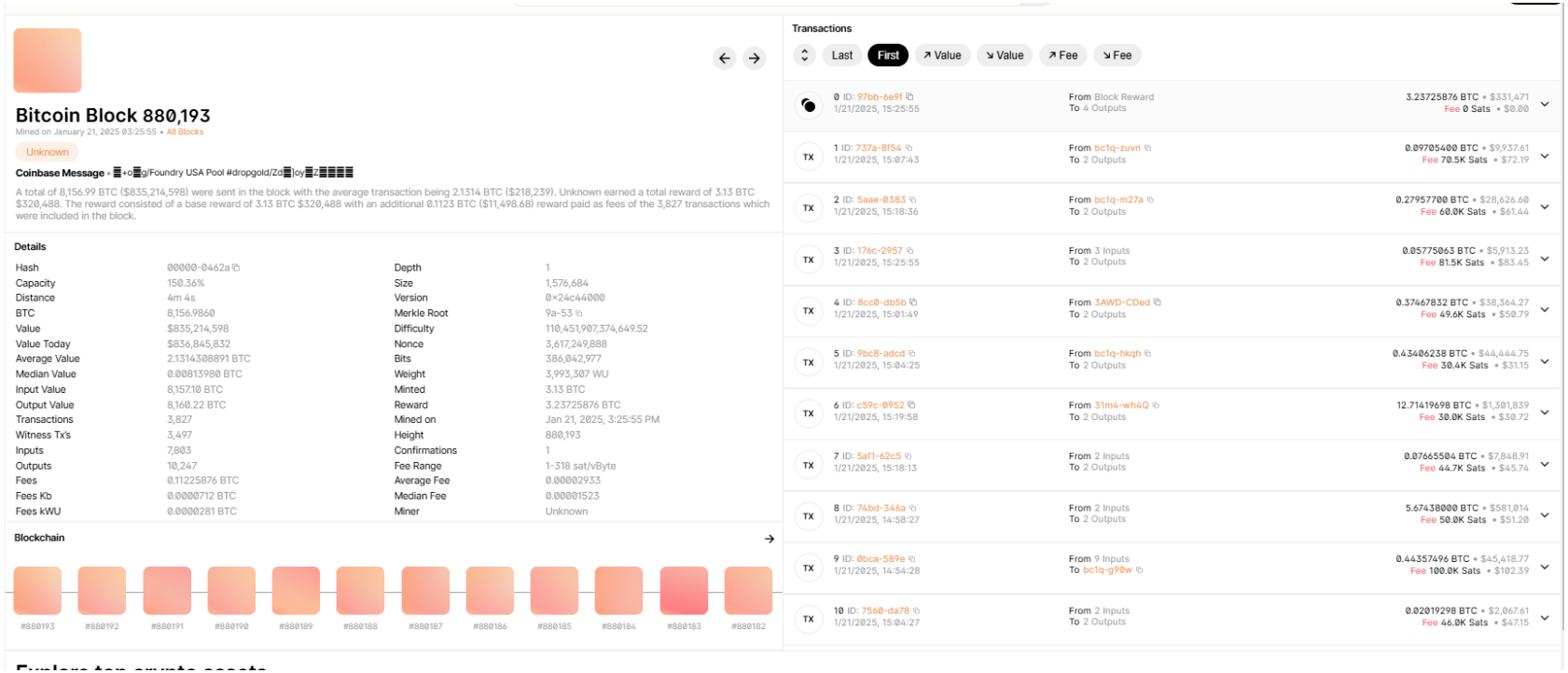
- Digitally signed and encrypted transaction verified by the peers.
- Cryptographic security- Ensures that participants can only view information on the ledger that they are authorized to see



- A block is a container data structure that contains a series of transactions.
- In Bitcoin: A block may contain more than 500 transactions on average, the average size of a block is around 1 MB (an upper bound proposed by Satoshi Nakamoto in 2010.

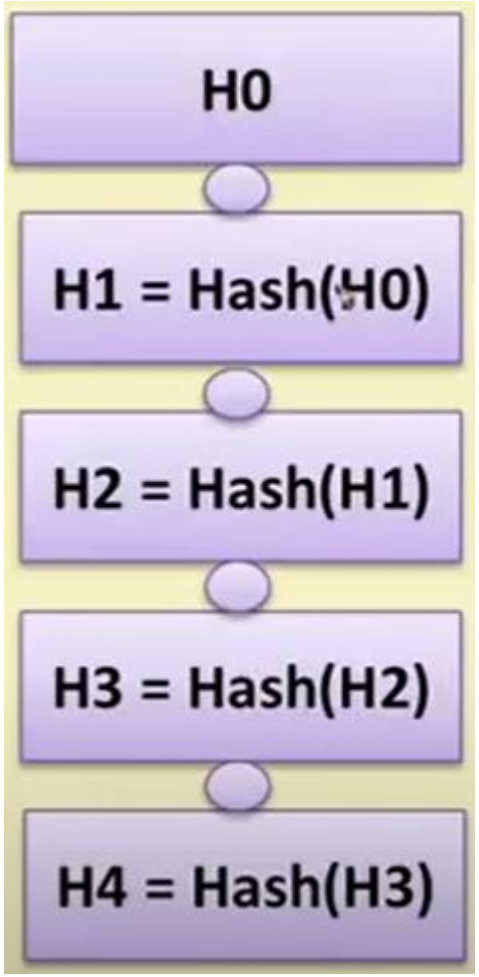
Structure of a Block (Ref. Bitcoin)

- Two components:
- - Block Header,
- - List of transactions



Block Header (Ref. Bitcoin)

- Metadata about a block – (1) Previous block hash, (2) Mining statistics used to construct the block, (3) Merkel tree root.
- Previous block hash: Every block inherits from the previous block – this make the blockchain tamper proof.



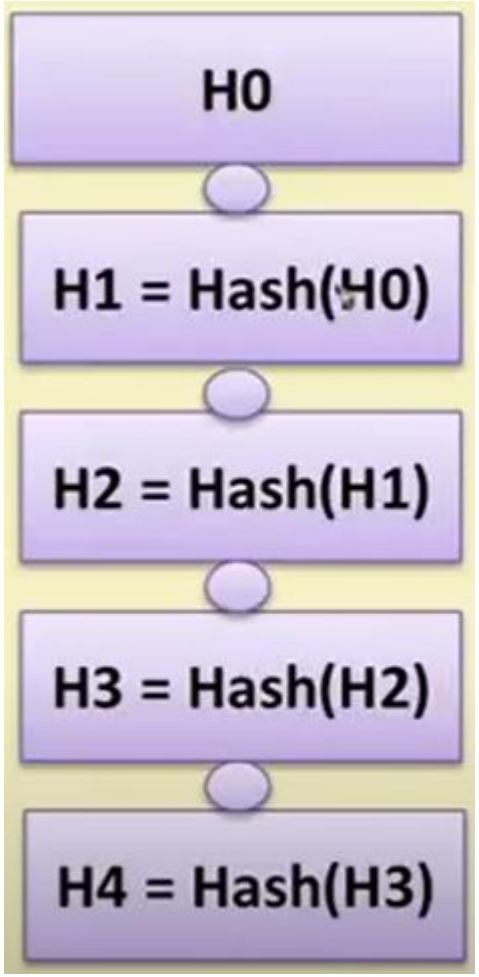
Block Header (Ref. Bitcoin)

Mining- the mechanizing to generate the hash

- The mechanisim needs to be complicated enough, to make the blockchain tamper proof.
- **Bitcoin Mining**: $H_k = Hash(\ H_{k-1}\ ||T||Nonce)$
- Find the nonce such that Hk has the certain predefined **complexity** (number of zeros at the prefix).

The header contains mining statistic – timestamp, nonce and difficulty.

Merkle tree root: already discussed



Transaction in a Block(Ref. Bitcoin)

- Transactions are organized as a Merkle tree. The Merkle Root is used to construct the block hash.
- If you change a transaction, you need to change all the subsequent block hash.
- The **difficulty** of the mining algorithm determines the toughness of tampering with a block in a blockchain.

The Blockchian Replicas

- Every peer in a Blockchain network maintains a local copy of the Blockchain.
- Requirements:
  - All the replicas need to be **updated** with the last mined block
  - All the replicas need to be **consistent** – the copies of the Blockchain at peers need to be **exactly similar**.

The Notion of Distributed Consensus

- Ensure that different nodes in the network see the same data at nearly the same point of time.
- All nodes in the network need to agree or consent on a regular basis, that the data stored by them is the same.
- No single point failure – the data is decentralized
- The system can provide service even in the presence of failures.
- Starting for early 90's a large number of works have been devoted on the development of consensus algorithms over network.
- The basic philosophy requires that the participants in the consensus algorithm knows each other.

The Notion of Distributed Consensus

- Can we achieve consensus **even when the network is artrarily large,** and **no participant in the network really knew all other participants?**
- An open network scenario – the permission-less protocol – you do not record your identity while participating in the consensus system.
- A **challenge-response** based system – the network **would pose a challenge**, and each node in the network **would attempt to solve the challenge**.

Challenge-Response to Permission-less Consensus

- The challenge-response protocol: The nodes in the network tries to solve the challenge posed by the network
  - The nodes or the participants do not need to reveal their identity.
- The node that is abel to solve the challenge first, would get to dictate what the set of data or state elements to be added should be
- This will continue iteratively at different rounds.

Challenge-Response to Permission-less Consensus

- **Design of a good challenge-** ensures that different nodes will win the challenge at different rounds.
- This ensures that no nodes would be abel to control the network
- The Bitcoin **Proof of Work (PoW)** algorithm – **ensures consensus over a permission-less setting based on challenge-response**.
- The challenge-response requires that every node spend large amount of computational power to solve a mathematical challenge in each iteration of consensus.
- **What is the incentive for nodes?** Only one (or sometime a very few of them) will win in each round.

The Economics behind the Blockchain Consensus

- **The Digital money:**
  - Ensures operational efficiency
  - More levels of controlling monetary policy.
- 1998: Wei Dai published ‘b-money’ – an anonymous distributed cash system.
- Cryptocurrency- a currency beyond the control of bank and governments.
- The mining ensures that no nodes has the power to sabotage the network and gain control
  - No one can hold the control of the cryptocurrency
- The computational effort expended by the nodes in achieving consensus would be paid for by cryptocurrency generated and managed by the network.
- Blockchain ensures that the currency is secure and tamper-proof.

Summary

- The technology behind Blockchain
  - **The Data Structure** – Distributed Ledger
  - **Cryptography and Digital Signatures** – Ensures security and tamper-proof architecture.
  - **The consensus** over a Permission-less Enveironment
  - **The Economy f the Revenue Model** – Encourages participants to join the minig procedure.