BlockseBlock

www.blockseblock.com

Title:

Mini Task 1: Build & Explain a Simple Blockchain

Submitted By:

Name: J Andrea

Internship Role: Blockchain Developer Intern

(Group F)

Date of Submission:

8th June, 2025

THEORETICAL PART

1. Blockchain Basics

- **Define Blockchain:**

A **blockchain** is like a digital notebook that keeps records of transactions in a secure and organized way. Instead of one person controlling it, many computers (called nodes) work together to maintain and update it. Each record (called a block) contains data like time, transaction info, and a unique code called a hash. These blocks are connected in a chain, and once something is added, it cannot be easily changed or deleted. That's what makes it trustworthy.

Blockchain is **decentralized**, which means no single person or company controls it. It's also **transparent** and **tamper-resistant**, making it useful for situations where trust and security are important.

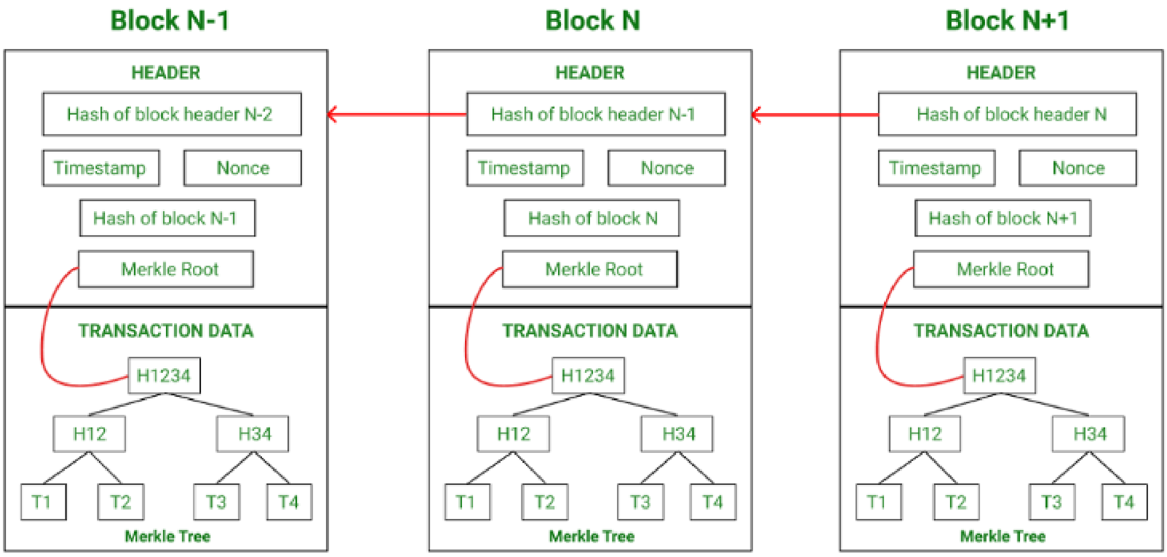
- **Two Real-Life Use Cases:**

i. Supply Chain Management
In supply chains, products go through many steps before reaching the customer — like being made, shipped, and stored. With blockchain, each step of a product's journey can be recorded in a secure, transparent way. For example, if you're buying a smartphone, blockchain can show where its parts came from, who assembled it, and how it was shipped. This helps verify that the product is real and not fake. It also helps companies spot delays or problems more quickly. Since the data can't be easily changed, it builds trust between suppliers, businesses, and customers.

ii. Digital Identity
Managing identity online can be risky — passwords get hacked, and personal data can be stolen. Blockchain offers a safer way. People can store their digital identity (like ID cards, certificates, or medical records) in an encrypted, tamper-proof format. Instead of giving all your details to every website or company, you can use blockchain to prove who you are with just enough information. This protects your privacy and gives you more control over your personal data. It's especially useful in areas like online banking, voting systems, or applying for government services.

2. Block Anatomy:

- **Block Showing: Data, Previous Hash, Timestamp, Nonce, and Merkle Root:**

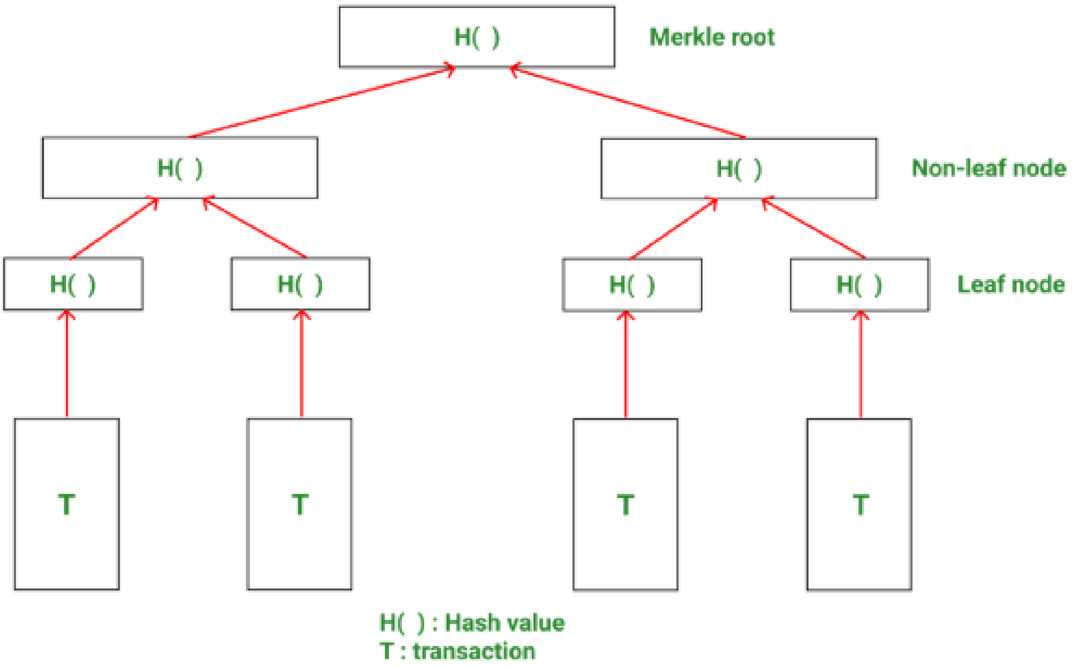


Each block comprises of block header + Merkle tree

Example:

- **Explanation of How the Merkle Root Helps Verify Data Integrity with Example:**

A **Merkle Root** is like a summary of all the transactions in a block. It is created by hashing pairs of transactions together until you get one final hash — the root.



Structure of Merkle tree

1. **Blockchains have thousands of blocks**, and each block can have thousands of transactions. So, storing and checking all of this takes a lot of memory and computing power.
2. **Merkle trees** help solve this by allowing transactions to be verified using a small amount of data, which makes things faster and more secure.
3. A **Merkle tree** works like this:
    - Each transaction is first hashed (turned into a short digital fingerprint).
    - These transaction hashes are grouped in **pairs**.
    - Each pair is hashed again to create a **parent node**.
    - This process repeats until you get one final hash at the top — called the **Merkle Root**.
    - The Merkle Root is stored in the block's header.
4. There are **three types of nodes** in a Merkle tree:
    - **Leaf nodes**: These contain the hash of each transaction.
    - **Non-leaf (intermediate) nodes**: These contain the hash of two child nodes.
    - **Root node (Merkle root)**: The final single hash representing all transactions.
5. **Bitcoin uses the SHA-256** hashing algorithm to calculate all these hashes securely.
6. A Merkle tree is a **binary tree** — it works best with an even number of leaf nodes. If there's an odd number, the last transaction hash is copied to make the number even.

Merkle Root Helps Verify Data Integrity:

Imagine you have a block with many transactions. Normally, to check if all transactions are valid, you'd need to go through each one — which is slow and uses a lot of memory.

But with a **Merkle tree**, all transaction data is turned into **hashes**, and these hashes are grouped and hashed again until a single final hash — the **Merkle root** — is created.

Suppose someone tries to **change just one transaction** (say, T3):

1. The hash of T3 (called **Hash3**) will change.
2. This change causes the **parent hash** (e.g., HashB) that includes Hash3 to also change.
3. This change **propagates upward** in the Merkle tree.
4. Eventually, the **Merkle root** at the top of the tree will also change.

Since the **Merkle root is stored in the block header**, and every full node in the blockchain has a copy of it, the nodes can **quickly compare the Merkle root**:

- If it's different, they **know immediately** that something in the transaction data was tampered with.
- They **don't need to check every single transaction** — just the path from that transaction up to the root.
- **Saves time**: You only check a few hashes, not the whole block.
- **Saves space**: You don't need to store all the data to verify it.
- **Boosts security**: Even the smallest change is caught, because hashes are extremely sensitive to input changes.

3. **Consensus Conceptualization**:

- **What is Proof of Work and why does it require energy?**

  Proof of Work is a consensus algorithm where miners solve complex mathematical puzzles to validate transactions and add new blocks to the blockchain. This process requires significant computational power, which consumes a large amount of electrical energy. The energy-intensive work ensures security by making it costly to attempt malicious attacks like double spending. The first miner to solve the puzzle gets to add the block and receive a reward.

- **What is Proof of Stake and how does it differ?**

  Proof of Stake is a consensus method where validators are chosen to create new blocks based on the amount of cryptocurrency they hold and "stake" or lock up as collateral. Unlike PoW, PoS does not require heavy computational work or massive energy consumption. Instead, validators are incentivized to act honestly since they risk losing their staked coins if they validate fraudulent transactions.

- **What is Delegated Proof of Stake and how are validators selected?**

  Delegated Proof of Stake is a variation of PoS where coin holders vote to elect a limited number of trusted delegates or validators who are responsible for validating transactions and creating blocks. These validators take turns producing blocks in a round-robin fashion. DPoS increases efficiency and transaction speed by reducing the number of nodes involved in consensus while maintaining decentralization through the voting process. Validators can be voted out if they perform poorly or dishonestly.

PRACTICAL PART (CODE-BASED TASKS)

GitHub Repository:

https://github.com/AndreaJohnMartin/Blockchain_Internship