**Blockchain-Based-Cryptocurrency-Platform**

**RAT Coin VS Bitcoin**

Bottom of Form

| **Aspect** | **RAT COIN** | **Bitcoin (Real Cryptocurrency)** |
| --- | --- | --- |
| **Consensus Mechanism** | **Proof-of-Capacity (PoC)**: Simulated by checking hashes in plot files. | **Proof-of-Work (PoW)**: Requires solving computationally expensive mathematical puzzles. |
| **Mining Difficulty** | Simulated by checking if a hash ends with "00". | Dynamically adjusted to maintain a 10-minute block creation time. |
| **Mining Reward** | Fixed at **50 coins per block**. | Bitcoin reward halves approximately every 4 years (halving). |
| **Blockchain Validation** | Checks hash consistency and previous\_hash matches. | In addition to hash verification, Bitcoin ensures decentralization and trust using public-private cryptography and a distributed consensus mechanism. |
| **Block Data** | Stores **transactions**, index, nonce, timestamp, and previous\_hash. | Similar structure but includes **Merkle Trees** to efficiently validate transactions. |
| **Transactions** | Simple: Sender, recipient, and amount. | Includes additional features: signatures, multi-input/output transactions, and transaction fees. |
| **Network Type** | **Centralized**: Single server (Flask application). | **Decentralized**: Peer-to-peer network with thousands of nodes globally. |
| **Blockchain Storage** | Stored entirely in memory within a Python list (self.chain). | Distributed ledger stored on all Bitcoin nodes (full and lightweight nodes). |
| **Node Communication** | No peer-to-peer communication; single-node implementation. | Full peer-to-peer protocol for broadcasting and verifying transactions/blocks. |
| **Mining Process** | Simulated by reading from plot files (proof\_of\_capacity). | Real mining involves solving a cryptographic puzzle using hash power. |
| **Currency Supply** | No defined total supply (only rewards miners). | Bitcoin’s total supply is capped at **21 million coins**. |
| **Security** | Basic hash verification with no cryptographic signatures. | Uses **public-key cryptography** for secure and verifiable transactions. |
| **Smart Contracts** | No smart contract capabilities. | Limited scripting capability via Bitcoin Script (e.g., multi-signature wallets). |
| **Transaction Fees** | None. | Transaction fees are paid to miners for prioritization in the network. |
| **Genesis Block** | Created with static attributes ("Genesis Block"). | Bitcoin’s Genesis Block contains a hardcoded reference to a 2009 newspaper headline. |
| **APIs/Endpoints** | Simple Flask API for adding transactions, mining, and viewing the blockchain. | Bitcoin operates via wallets and nodes using a protocol defined by Bitcoin Core. |
| **Currency Name** | Your custom name: **RAT Cryptocurrency**. | Bitcoin (BTC). |
| **Code Optimization** | Suitable for learning and personal projects but not efficient for production. | Bitcoin’s C++ codebase is highly optimized for performance and scalability. |
| **Scalability** | Limited to a single-node environment; performance degrades as chain grows. | Scalable via Lightning Network and SegWit for faster transaction processing. |
| **Anonymity** | No anonymization; all transaction details (sender, recipient, amount) are visible. | Pseudonymous: Transactions are tied to public keys, not real identities. |
| **Cryptography** | Basic SHA-256 hashing only. | Combines **SHA-256** with **RIPEMD-160** and elliptic-curve cryptography (ECDSA). |
| **Distributed Ledger** | Not distributed; resides on one machine. | Fully distributed across all nodes in the network. |
| **Real-World Usage** | Educational or experimental project. | Used globally as a medium of exchange, store of value, and investment asset. |

**Key Differences**

1. **Decentralization**:
   * Your implementation is centralized and runs on a single machine.
   * Bitcoin is fully decentralized and relies on thousands of independent nodes.
2. **Consensus Mechanism**:
   * You use a basic PoC mining mechanism simulated with file operations.
   * Bitcoin uses PoW, which is computationally intensive and ensures security.
3. **Security**:
   * Your project lacks advanced cryptography (e.g., digital signatures, public/private key pairs).
   * Bitcoin ensures transaction security using elliptic-curve cryptography (ECDSA).
4. **Scalability**:
   * Your implementation is single-threaded and unsuitable for scaling to large datasets.
   * Bitcoin has global scalability solutions like the Lightning Network.
5. **Purpose**:
   * Your code serves as an educational tool for learning blockchain concepts.
   * Bitcoin is a real-world, widely adopted cryptocurrency.