# BT2C: A Pure Cryptocurrency Implementation with Reputation-Based Proof of Stake

BT2C Core Development Team

March 19, 2025

**Abstract.** This paper introduces BT2C (bit2coin), a cryptocurrency designed to function as a pure medium of exchange and store of value without the overhead of smart contracts or decentralized applications. We present a novel reputation-based Proof of Stake (rPoS) consensus mechanism that addresses the energy consumption concerns of Proof of Work while maintaining security properties through cryptographic verification and economic incentives. BT2C implements a deterministic issuance schedule with a fixed maximum supply, combined with a flexible staking model that optimizes for network security and validator participation. This paper outlines the technical architecture, cryptographic foundations, consensus rules, and economic model of the BT2C network.

## 1. Introduction

Cryptocurrencies have evolved significantly since the introduction of Bitcoin in 2009. While many projects have expanded into complex platforms supporting smart contracts and decentralized applications, BT2C returns to the original vision of a peer-to-peer electronic cash system with improvements in energy efficiency, transaction finality, and participation accessibility.

The core innovation of BT2C lies in its reputation-based Proof of Stake consensus mechanism, which selects validators based on a combination of stake amount and historical performance metrics. This approach provides three key advantages:

- Energy efficiency compared to Proof of Work systems
- Resistance to centralization through accessible validator requirements
- Enhanced security through reputation-based incentives

## 2. System Architecture

### 2.1 Network Topology

BT2C implements a peer-to-peer network with specialized validator nodes responsible for block production. The network utilizes a gossip protocol for message propagation with the following components:

- Seed nodes: Entry points for new validators joining the network

- Validator nodes: Full nodes with staked BT2C that participate in consensus
- API nodes: Provide REST and WebSocket interfaces for client applications
- Explorer nodes: Index and serve blockchain data for user interfaces

Network communication occurs over TCP/IP with mandatory TLS encryption. Each node maintains a connection pool with a configurable number of peers, with priority given to connections with validators having higher reputation scores.

### 2.2 Data Structures

#### 2.2.1 Blocks

Each block in the BT2C blockchain contains a header with metadata and a body with transactions. Block hashes are computed using SHA3-256 over the concatenation of critical block data including height, timestamp, merkle root, validator address, reward amount, and previous block hash.

#### 2.2.2 Transactions

Transactions in BT2C follow a simple structure with sender, recipient, amount, fee, nonce, timestamp, signature, and hash. Transaction hashes are computed using SHA3-256, and signatures are generated using the sender's private key.

### 2.3 Cryptographic Foundations

- Key generation: 2048-bit RSA key pairs
- Address derivation: base58 encoded hash of public key
- Transaction signing: RSA-PSS with SHA-256
- Block and transaction hashing: SHA3-256
- Seed phrases: BIP39 with 256-bit entropy (24 words)
- HD wallet derivation: BIP44 path m/44'/999'/0'/0/n

## 3. Consensus Mechanism

### 3.1 Reputation-Based Proof of Stake (rPoS)

BT2C introduces a reputation-based Proof of Stake consensus mechanism that extends traditional PoS by incorporating historical performance metrics into validator selection. The probability of a validator being selected is determined by both their stake amount and reputation score.

The reputation score is calculated based on block validation accuracy, uptime, transaction processing efficiency, and historical participation quality. This formula ensures that even validators with poor reputation maintain a minimum selection probability, while high-performing validators can achieve up to 200% of their stake-based probability.

### 3.2 Block Production

Block production in BT2C follows a time-based schedule with a target block time of 60 seconds. The selected validator has 30 seconds to produce and broadcast a valid block, after which a new validator is selected if necessary.

### 3.3 Block Validation

- Verify the block structure and hash
- Verify the selected validator's eligibility
- Verify each transaction's signature and validity
- Verify the Merkle root matches the transactions
- Verify the block reward calculation

A block is considered finalized when it has been built upon by 6 subsequent blocks, providing probabilistic finality similar to Bitcoin but with significantly shorter confirmation times due to the 60-second block interval.

### 3.4 Validator States

Validators in BT2C can exist in four distinct states: Active, Inactive, Jailed, or Tombstoned. Transitions between states follow specific rules based on validator behavior and protocol compliance.

## 4. Economic Model

### 4.1 Supply Schedule

- Maximum supply: 21,000,000 BT2C
- Initial block reward: 21.0 BT2C
- Halving interval: 4 years (approximately 2,102,400 blocks)
- Minimum reward: 0.00000001 BT2C

### 4.2 Fee Market

Transaction fees in BT2C are dynamic, based on network congestion. The minimum fee increases with network utilization, ensuring fees remain low during normal operation but increase during periods of high demand to prioritize transactions efficiently.

### 4.3 Staking Economics

- Minimum stake: 1.0 BT2C
- No maximum stake: Validators can stake any amount above the minimum
- No fixed staking period: Validators can unstake at any time
- Unstaking queue: Withdrawals enter a FIFO queue processed over time
- Queue processing rate: Limited to 1% of total stake per day

### 4.4 Validator Incentives

- Developer Node Reward: 1000 BT2C (one-time reward for first mainnet validator)
- Early Validator Reward: 1.0 BT2C (for validators joining during distribution period)
- Distribution Period: 14 days from mainnet launch
- Auto-staking: All distribution period rewards are automatically staked

## 5. Security Considerations

### 5.1 Sybil Resistance

The minimum stake requirement of 1.0 BT2C provides a basic economic barrier against Sybil attacks. Additionally, the reputation system requires consistent participation over time to achieve maximum influence, making it costly to establish multiple high-reputation validators.

### 5.2 Nothing-at-Stake Problem

- Slashing conditions: Validators who sign conflicting blocks lose a portion of their stake
- Reputation penalties: Double-signing results in immediate reputation reduction to minimum
- Tombstoning: Severe violations result in permanent exclusion from the validator set

### 5.3 Long-Range Attacks

- Weak subjectivity checkpoints: Published every 10,000 blocks
- Time-bound validator set changes: Validator set changes take effect only after a delay

- Social consensus backstop: Community-recognized canonical chain in case of deep reorganizations

### 5.4 Transaction Replay Protection

Each transaction includes a unique nonce derived from the sender's account state, preventing transaction replay attacks. The nonce is incremented with each transaction, and the network rejects transactions with previously used nonces.

## 6. Implementation Details

### 6.1 Core Components

- Consensus engine: Implements the rPoS algorithm and block validation rules
- Transaction pool: Manages pending transactions and fee prioritization
- State machine: Tracks account balances, stakes, and validator metadata
- P2P network: Handles peer discovery and message propagation
- API server: Provides external interfaces for clients and services

### 6.2 Data Persistence

- Blockchain data: Stored in a custom append-only file format optimized for sequential access
- State data: Maintained in a PostgreSQL database for efficient querying and indexing
- Mempool: Held in memory with Redis backup for persistence across restarts

### 6.3 Performance Optimizations

- Parallel transaction verification: Multiple CPU cores validate transaction signatures concurrently
- Incremental state updates: State changes are applied incrementally rather than recomputing
- Bloom filters: Used to quickly check transaction existence without full lookups
- Connection pooling: Database connections are pooled for efficient resource utilization

### 6.4 Network Parameters

- Target block time: 60 seconds
- Maximum block size: 1MB
- Transaction throughput: Up to 100 tx/s

- Rate limiting: 100 requests/minute
- Network ports: 26656 (P2P), 26660 (metrics)

### 6.5 Infrastructure Requirements

- Hardware: 4 CPU cores, 8GB RAM, 100GB SSD
- Software: Docker & Docker Compose
- Database: PostgreSQL
- Caching: Redis
- Monitoring: Prometheus & Grafana

## 7. Distribution Period Mechanics

To bootstrap the network securely, BT2C implemented a 14-day distribution period with special incentives:

- Early validator reward: 1.0 BT2C for any validator joining during this period
- Developer node reward: 1000 BT2C for the first validator (network founder)
- Automatic staking: All distribution rewards are automatically staked

These mechanics were designed to ensure sufficient initial stake distribution while maintaining security during the critical launch phase.

## 8. Conclusion

BT2C represents a focused approach to cryptocurrency design, combining Bitcoin's economic principles with modern consensus mechanisms. By implementing reputation-based Proof of Stake, BT2C achieves energy efficiency without sacrificing security or decentralization.

The deliberate exclusion of smart contracts and complex programmability allows BT2C to optimize for its core use case: a secure, efficient medium of exchange and store of value. This specialization enables performance optimizations and security hardening that would be challenging in more general-purpose blockchain platforms.

As the network continues to mature beyond its initial distribution period, the reputation system will increasingly reward consistent, high-quality participation, creating a virtuous cycle of improved security and performance.