BTX Whitepaper

Author: jaBro
Date: August 2025 ver 3

Abstract:

BTX is a scalable, privacy-focused, and decentralized Layer-2 protocol for Bitcoin. It enables fast, low-cost, and globally accessible transactions while remaining anchored to Bitcoin's security model and economic foundation. Designed for both human and machine use, BTX achieves instant local finality, offline transaction capability, and strong resistance to censorship or surveillance—without compromising Bitcoin’s core principles.

Executive Summary

BTX (short for Bitcoin Transaction Extension) is a Bitcoin-native Layer-2 solution designed to deliver scalable, decentralized, and privacy-preserving payments for both humans and machines. It introduces a novel transaction DAG (Directed Acyclic Graph) that finalizes payments in approximately five seconds under normal load, while maintaining anchoring to the Bitcoin blockchain for external auditability and long-term integrity.

BTX is built with a dual-layer architecture:

- A minimal, immutable Core, responsible for transaction structure, DAG logic, anchoring behavior, and enforcement of base transaction fees.
- A set of modular, upgradeable Governance Modules that define adjustable parameters such as fee structures, validator incentives, privacy behaviors, and governance rules.

BTX introduces two wallet types:

- Human Wallets, which follow simplified rules (1 MOTO minimum per transaction, percentage-based fees).
- AI Wallets, which support micro-payments (sub-MOTO values), batching, and flat fee logic optimized for machine-to-machine use.

Offline transactions are supported by default, with staged finality via SMS, mesh network, or file-based relays. Transactions are stored locally and finalized once any connection is re-established.

The system operates without wrapped tokens, bridges, or trusted custodians. All transactions are denominated in MOTO, BTX's internal unit, which is algorithmically pegged to Bitcoin (1 BTC = 10,000,000,000 MOTO). Redemption to and from Bitcoin is seamless, decentralized, and enforced by the Core.

To ensure sustainability, BTX includes a low, dynamically adjustable fee system with direct incentives for both validators (BTC full nodes) and Bitcoin miners. A capped development fund, governed by a multi-stage randomized consensus process, funds long-term evolution without introducing centralized control.

BTX offers default-on privacy via address encryption, decoy transaction logic, and transaction shuffling. All metadata linking is broken by design, ensuring temporal unlinkability and resistance to long-term surveillance.

This whitepaper defines the complete BTX model, including:

- Anchoring logic
- Transaction DAG structure
- Wallet types and restrictions
- Fee logic and validator incentives
- Offline transaction mechanics
- Governance architecture
- Minting, redemption, and BTC buffer rules
- Privacy and surveillance resistance
- Technical structure and bootstrap behavior

BTX fills a critical gap in Bitcoin’s evolution. Without undermining Bitcoin’s decentralization, security, and monetary policy, It enables global retail payments and high-frequency automated microtransactions.

Table of Content

Chapter 1 – Introduction

Bitcoin is the most secure and decentralized monetary network ever created. However, its Layer-1 architecture was never designed to support the high-frequency, low-cost transactions needed for everyday payments, machine-to-machine transfers, or real-world commerce. As a result, Bitcoin adoption is often limited to long-term savings, institutional custody, or centralized exchange use.

BTX (Bitcoin Transaction Extension) is a new Layer-2 protocol designed to close this gap. It enables scalable, censorship-resistant, and privacy-preserving transactions that remain fully anchored to Bitcoin, without relying on bridges, custodians, or wrapped tokens.

Unlike most Layer-2 systems that introduce external tokens or control structures, BTX is governed entirely through modules built on Bitcoin full node infrastructure. It is designed to be run by the same decentralized group that protects Bitcoin today, offering natural alignment between Layer-1 and Layer-2 participants.

BTX introduces the following core principles:

Bitcoin-native anchoring:
Every BTX transaction is ultimately secured through anchoring into the Bitcoin blockchain. This ensures external verifiability and long-term integrity without sacrificing speed or scalability.

Directed Acyclic Graph (DAG) transaction model:
Instead of sequential blocks, BTX uses a transaction DAG, where each transaction confirms two others. This model enables asynchronous processing, rapid finality, and scalable throughput.

Dual-wallet design:
BTX supports both Human and AI wallets, each with its own transaction policies. This ensures simplicity for casual users and precision for automated systems.

Offline and delayed finality support:
BTX is designed for global reach, including environments with unreliable connectivity. Transactions can be signed and stored offline, then finalized via mesh networks, SMS, or internet relays.

Minimal, immutable Core with modular governance:
The BTX Core enforces only fundamental rules. All dynamic logic—such as fees, validator incentives, governance models, and anchoring cadence—is managed through external modules.

Privacy by default:
All transactions use encrypted addresses, decoys, and transaction path mixing to ensure metadata protection. BTX obfuscates not just transaction values, but the underlying graph itself.

Validator and miner incentive alignment:
BTX transaction fees are split between BTX validators (Bitcoin full node operators) and Bitcoin miners. This ensures both groups remain economically aligned as the Bitcoin block subsidy decreases over time.

Sustainability and governance:
A capped development fund and multi-stage, randomized validator voting system allow for decentralized upgrades, without relying on token-based voting or centralized control.

BTX is not designed to replace Bitcoin or compete with other Layer-2s. Its purpose is to enable a transactional ecosystem that is fast, secure, and human-friendly—while strengthening the Bitcoin network and preserving its core values.

For technical specifications related to this chapter, see: Appendices A.1–A.3, B.1–B.3, C.1–C.2, and D.

Chapter 2 – Anchoring to Bitcoin

BTX is fully secured by Bitcoin through a recurring anchoring process. Every BTX transaction is embedded within a Directed Acyclic Graph (DAG), which is regularly hashed and recorded into the Bitcoin blockchain using a standard OP_RETURN transaction.

This anchoring mechanism provides the following benefits:

- Immutable history: Each BTX DAG snapshot is permanently recorded in Bitcoin, ensuring auditability.
- Decentralized trust: Validators and users can verify DAG correctness by comparing anchored hashes.
- Lightweight sync for new nodes: Anchored hashes allow nodes to verify historical finality without downloading the entire DAG.

Anchoring Frequency

Anchoring frequency is determined by a rolling 4-day average of total BTX transaction volume (TX/day), measured via governance-approved modules.

Cadence adjustment follows an asymmetric rule:

- To Reduce Frequency:
  BTX reduces anchoring frequency *only* if:
  1. The rolling average exceeds the next threshold by at least 5%, and
  2. This condition is sustained for two consecutive rolling windows (8 days total)
- To Increase Frequency:
  If the rolling average drops below a threshold, anchoring cadence is increased immediately, without delay or margin. This ensures conservative behavior during low activity or potential disruptions.

Volume Thresholds (Based on Rolling 4-Day Average):

Volume Level	Anchoring Frequency
< 1,000 TX/day	Anchor every Bitcoin block
1,000–10,000 TX/day	Anchor every 3rd Bitcoin block
> 10,000 TX/day	Anchor every 12th block

This model prevents flapping near boundary values, preserves validator predictability, and strengthens audit integrity.

Note: During low activity, AI and privacy-enabled wallets may insert dummy transactions to maintain DAG density and cadence stability. This behavior is voluntary but incentivized.

Validator Selection and Enforcement

Each anchoring round selects one validator at random from eligible online participants. If the selected validator fails to anchor within 60 seconds, a new one is selected. Failure to anchor while online results in a 7-day suspension from all validation duties.

Offline validators are excluded from selection but are not penalized.

Anchoring Format

Each Bitcoin anchor transaction includes:

- Hash of the confirmed BTX DAG snapshot
- Timestamp
- Reference to previous anchor
- Minimal metadata (TX count, anchor index)
- Hashed Validator ID (for accountability)

Anchors are chained together to create a verifiable trail of BTX states anchored to Bitcoin. The hashed Validator ID ensures traceability without compromising privacy.

Auditability and Finality

Anchors act as public finality checkpoints. Observers can verify DAG state up to the latest anchor. After 125 DAG confirmations, transactions may be pruned unless required for BTC redemption.

Anchoring ensures that BTX inherits Bitcoin's long-term security guarantees without compromising speed, scalability, or privacy.

For technical specifications related to this chapter, see: Appendices A and B

Chapter 3 – Transaction DAG Model

BTX replaces the traditional block structure used in Bitcoin with a Directed Acyclic Graph (DAG) model. In a DAG, every new transaction must reference and confirm two prior transactions. This structure enables parallel processing, asynchronous confirmation, and much higher scalability than block-based designs.

Core DAG Rules

Each BTX transaction must:

- Confirm two existing transactions in the DAG
- Include a valid digital signature
- Reference the sender's most recent transaction
- Comply with wallet-specific transaction rules

The DAG is validated continuously by all participating nodes. Transactions propagate through the network and are confirmed recursively as new transactions reference them.

Finality and Confirmation Depth

Finality is defined as the point at which a transaction has received 100 subsequent confirmations in the DAG. This typically takes between 5 and 10 seconds under normal conditions.

To ensure a consistent and efficient pruning model, transactions are removed from validator memory after 125 confirmations unless they are part of the BTC redemption chain. Transactions related to BTC minting and redemption may be retained longer for auditability.

The DAG structure ensures that:

- No transaction can be reversed after finality

- Transactions can confirm each other asynchronously
- Finalization does not require a global clock or sequencing

Transaction Linking and Double-Spend Protection

Each transaction references:

- Two parent transactions it confirms
- The most recent transaction by the same sender

This structure prevents double-spending by making every new transaction dependent on the sender's latest DAG state. Any conflicting transaction that references the same funds will fail validation.

Validators reject any transaction that:

- References invalid parents
- Breaks sender sequence rules
- Spends already-spent inputs

This creates a deterministic and tamper-proof execution environment without the need for global synchronization.

DAG Growth and Volume Scaling

As transaction volume increases, the DAG grows denser. This improves confirmation speed and overall throughput. DAGs naturally scale with usage, unlike blockchains which rely on fixed block intervals.

The BTX DAG is self-healing and adaptive:

- Under low volume, dummy transactions are inserted to keep finality moving
- Under high volume, confirmation speed increases as more transactions enter the graph
- The structure can scale to tens of thousands of TPS without centralization

DAG integrity is enforced by the Core, while dummy transaction behavior and anchoring cadence are governed by modules.

BTX achieves a scalable, resilient transaction graph that adapts dynamically to network conditions while maintaining security, transparency, and high-speed finality.

For technical specifications related to this chapter, see: Appendices B and C

Chapter 4 – Wallets and Transaction Rules

BTX supports two distinct wallet types: Human Wallets and AI Wallets. Each has its own rules for transaction formatting, value thresholds, batching behavior, and fee structures. This dual model ensures a user-friendly experience for human participants while enabling efficient microtransactions and automation for machines and software agents.

Human Wallets

Human wallets are designed for simplicity, clarity, and ease of use. They interact with the network using only whole MOTO units and are optimized for predictable retail behavior.

Full-MOTO enforcement:
Human wallets may only send and receive full MOTO values (≥ 1 MOTO).
Any transaction below this threshold is rejected by validators.

No sub-MOTO values permitted:
Sub-MOTO amounts are neither displayable nor accepted.
This eliminates ambiguity and ensures clean address recycling.

No batching logic:
Human wallets submit one transaction at a time and are not eligible for batching discounts.

Fee auto-calculation:
Fees are automatically determined by the wallet using active policy modules. Manual fee entry is not supported.

This configuration ensures the wallet remains intuitive, avoids stuck balances, and supports consistent privacy, fee logic, and recycling behavior.

All Wallets

AI wallets are optimized for automation, batching, and high-frequency transactions. They support fine-grained control and cost-efficiency for software systems and embedded devices.

Can send and receive sub-MOTO values

Batching Support:

- Can batch up to 21 outgoing transactions
- Each batch confirms multiple prior DAG transactions

Fee Model: Flat fee per transaction, adjusted for purchasing power

- Starts at 250 MOTO
- Discounts applied for batching:
    - 3 TXs → 1%
    - 6 TXs → 2%
    - 12 TXs → 4%
    - 18 TXs → 6%
    - 21 TXs → 7%

Offline Transaction Limits

- Max 21 unfinalized transactions
- Max total offline value = 2,000 NAKA (200 million MOTO)

AI wallets are subject to randomized revalidation (see Appendix A) and must periodically recompute their cryptographic logic as part of anti-spam and integrity checks.

Transaction Structure and Core Enforcement:

All transactions must comply with Core rules regardless of wallet type. Each transaction includes:

- Sender and receiver addresses
- Amount in MOTO (even if displayed in NAKA or SAT)
- DAG references (two prior TXs and most recent sender TX)
- Timestamp and cryptographic signature

The BTX Core enforces:

- Valid DAG connections
- No double-spending
- Wallet-specific minimums and limits
- Finalization behavior

Any transaction violating these rules is automatically rejected by validators.

Finalization and DAG Behavior:

- Online transactions are finalized in ~5 seconds via DAG consensus
- Offline transactions are staged locally and finalized upon reconnection (see Chapter 5)
- Each wallet stores its most recent confirmed TX reference, which is used to build the DAG

All wallets operate within the same DAG environment, ensuring consistent rules, transparency, and decentralized enforcement.

For technical specifications related to this chapter, see: Appendices A and G.

Chapter 5 – Offline Transactions & SMS Relay

BTX is engineered to operate in environments with unreliable, censored, or absent internet access. Transactions can be created and stored offline, then finalized once either the sender or receiver regains connectivity. This capability makes BTX viable in regions with poor infrastructure or heavy surveillance.

5.1 — Staged Transaction Logic

Offline ("staged") transactions are fully formed BTX transactions that are cryptographically signed and stored locally in both sender and receiver wallets. Each staged transaction:

- Is timestamped
- References two valid DAG parents
- References the sender's last confirmed transaction
- Contains the sender's signature

These rules ensure staged transactions maintain full DAG structure and anti-double-spend guarantees, even before network submission.

When either party reconnects via internet, SMS, or mesh, the transaction can be broadcast for inclusion in the DAG.

5.2 — Finalization Mechanism

Once broadcast, a staged transaction is processed like any other BTX transaction and finalized within ~5 seconds under normal conditions.

If the originally selected DAG parents are no longer valid (e.g., pruned after 125 confirmations), the submitting wallet automatically:

1. Selects two recent, valid DAG parents

2. Re-references them in the transaction
3. Re-signs if necessary
4. Submits the updated transaction

This keeps the DAG structurally intact even after extended offline periods. Upon finalization, both sender and receiver are updated.

5.3 — Local Fast Finality

If both wallets are online and have received each other's signed transaction, payment can be treated as locally final within ~1 second—before DAG confirmation.
The transaction is still broadcast and finalized globally, but the wallet may display early confirmation to the user for smoother UX.

5.4 — Expiry & Abuse Prevention

To limit abuse (e.g., hoarding dust transactions offline), BTX enforces:

- **Expiry:** Staged transactions expire after 72 hours
    - Wallets warn users at 51 hours
- **Transaction count cap:**
    - Human wallets: max 9 staged transactions
    - AI wallets: max 21 staged transactions
- **Value cap:**
    - Human wallets: max 1,000 NAKA staged
    - AI wallets: max 2,000 NAKA staged

Transactions exceeding these limits are rejected or locally purged, with user notification.

5.5 — SMS Relay Protocol

When internet is unavailable, wallets can send staged transactions via SMS to **relay nodes**:

- Wallet compresses and obfuscates transaction before sending
- Relay nodes accept signed data without alteration
- SMS relay schedule: 7–10 random relay slots per day, each ~15 minutes
- Relay fee: 11,500 MOTO (~$0.10), waived in Year 1

The relay node injects the transaction into the DAG for normal confirmation.

5.6 — Privacy & Compression

SMS transactions:

- Are compressed and obfuscated to reduce metadata leakage
- Reveal only timestamp, DAG parents, and encrypted payload
- Conceal wallet balances and sender identity from relays

Once finalized, SMS-originated transactions are indistinguishable from internet-submitted ones.

5.7 — Resilience & Redundancy

Staged transactions can be finalized via:

1. Internet
2. SMS
3. Mesh networks (if enabled)
4. Local peer relay (file transfer / physical transfer)

This multi-path model ensures BTX remains operational under extreme network failure or censorship.

For technical specifications related to this chapter, see Appendices N and O.

Chapter 6 – Fees, Validator Incentives and Economic Sustainability

BTX is designed to sustain validator operations, incentivize long-term participation, and ensure Bitcoin miners remain supported even after block rewards decline. The system uses a modular, percentage-based fee structure combined with purchasing power (PP) stabilization to achieve predictable, low-cost payments without undermining security.

Validator Pool Structure and Scaling

BTX uses a dynamically scalable validator pool model to support high throughput, resilience, and decentralization. The network maintains two types of validator groups: transaction pools and a finalizer pool, each composed of randomly selected active validators.

Transaction Pools

Each transaction submitted to the DAG is validated by a transaction pool consisting of 21 randomly selected validators. For the transaction to be accepted, at least 16 out of 21 validators must confirm it. This 16/21 threshold ensures strong consensus while allowing for some node downtime or disagreement.

Transaction pools can scale dynamically based on network demand:

- New transaction pools are added if the following two conditions are met:
    - The average transaction volume exceeds 0.6 transactions per second (≈ 50,000/day), and
    - There are at least 51 unused active validators available (21 for the new pool, and a buffer to preserve finalizer rotation).

This scaling ensures that validation throughput increases proportionally with network usage, without overwhelming individual nodes.

Finalizer Pool

After a transaction is accepted by a transaction pool, it proceeds to the finalizer pool—a separate group of 21 validators that sign off on global finality and timestamp anchoring into the DAG. Finalizer pool members are entirely distinct from the transaction pool validators to prevent collusion and ensure architectural separation of responsibilities.

Finalization typically occurs within ~5 seconds under normal conditions. Finalizer confirmation requires ≥17 of 21

signatures before a transaction attains global finality; the Transaction Pool acceptance threshold remains 16 of 21.

The number of finalizer pools remains fixed at launch but may be expanded via a governance module if sustained high volume requires further scaling. Finalizer duties are rotated to ensure fairness and decentralization.

Randomized Validator Selection and Rotation

All validators—whether selected for transaction pools or the finalizer pool—are reselected at random intervals. Instead of a fixed schedule, the reselection occurs unpredictably within a defined window (e.g., every 20–28 hours or every 2,700–3,300 DAG confirmations). This randomness is derived from recent Bitcoin block hashes, ensuring transparency, unpredictability, and resistance to manipulation.

This design guarantees that no validator can predict or influence their assignment, preserving neutrality and minimizing the risk of validator capture.

Base Fee Model

BTX charges a small percentage-based fee on each transaction:

Default fee: 0.25% of transaction value

- Reduced fee: 0.10% after network volume surpasses ~10,000 transactions/day
- Fee adjustment is automatic and governed by module logic
- The reduction is designed to occur as network activity becomes self-sustaining

All BTX transactions are denominated in MOTO, but fee rates apply regardless of value unit (NAKA, SAT, TBTC, etc.).

Fee Floors and Caps

To prevent spam and protect high-value transfers, BTX uses a dual limit model:

- Minimum fee: 5,000 MOTO (≈ $0.06 at BTC = $120,000)
- Maximum fee: 2,100,000 MOTO (≈ $25.20 at BTC = $120,000)

These values are dynamically adjusted using a 5-month rolling BTC/USD average, ensuring stable purchasing power over time. The fee limits are enforced by a governance module, not the Core, allowing for future updates.

Minting and Redemption Fees

Conversions between BTC and BTX (via mint/burn contracts) include a flat redemption fee to discourage high-frequency flipping:

- Fee: 100,000 MOTO per mint or burn transaction (≈ $1.2 at BTC = $120,000)
- Adjusted for purchasing power via the same 5-month BTC/USD index

This fee is small enough to support fair conversion but deters misuse of the minting mechanism for arbitrage or short-term churn.

AI Wallet Flat Fees and Batching

AI wallets operate under a flat-fee model optimized for microtransactions:

- Base fee per TX: 250 MOTO
- Discounts for batching:
    - 3 TXs → 1%
    - 6 TXs → 2%

- 12 TXs → 4%
- 18 TXs → 6%
- 21 TXs → 7%

Batching allows a single AI wallet to confirm up to 21 transactions in one batch, reducing fee burden and improving DAG density.

Flat AI fees are also purchasing power-adjusted, ensuring predictability and usability for machine systems.

Fee Distribution Model

All transaction fees are split among the key actors securing the system:

- 82% to BTX validators (who run Bitcoin full nodes)
- 15% to Bitcoin miners
- 3% to the capped BTX development fund

Validator fees are distributed evenly among the 21 members of the validation pool that processes the transaction. The finalizer pool, which signs off on global finality, does not receive a fee share.

Bitcoin Miner Fee Distribution

The miner share of BTX fees (15%) is distributed directly to Bitcoin miners who have mined recent Bitcoin blocks, based on actual block production.

A dedicated module or external distribution script:

- Reviews a defined lookback period (e.g., last 144 Bitcoin blocks)
- Tallies how many blocks were mined by each miner or mining pool
- Distributes BTX miner fee share in direct proportion to mined blocks

This ensures that all BTC miners are supported even if they do not run BTX validator nodes. The logic for miner fee distribution is not part of the BTX Core, allowing the community to update or replace it via governance if needed.

Dynamic Miner Boost

To protect Bitcoin security long-term, the miner share increases automatically if Bitcoin difficulty drops significantly:

- If Bitcoin network difficulty falls below 85% of its average over the past 6 difficulty adjustment periods (≈ 12 weeks), the miner share rises linearly up to 50% of all BTX transaction fees
- This mechanism is enforced by module and does not affect Core security

The adjustment helps stabilize miner income in case of sustained hashrate decline or post-halving disruptions.

Development Fund and Cap

BTX allocates 3% of all transaction fees to a development fund, which is:

- Hard-capped at $50 million per year (purchasing power)
- Stored in a 21-of-26 multisig wallet (see Chapter 7)
- Surplus funds beyond the cap are redirected to Bitcoin miners
- Cap enforcement is handled via module, not the Core

This ensures a steady, predictable funding model for long-term upgrades without risking fund bloat or centralization.

BTX's fee system is designed to be simple, fair, and adaptive—serving both everyday users and long-term network health.

For technical specifications related to this chapter, see: Appendices B, J and L.

Chapter 7 – Governance, Dev Fund and Upgrade Logic

BTX governance is designed to be decentralized, technically rigorous, and resistant to capture. The system separates immutable Core logic from upgradeable modular components, enabling adaptability while preserving trust and long-term alignment with Bitcoin.

Immutable Core, Modular Upgrades

The BTX Core enforces fundamental transaction and anchoring rules. All other protocol elements—including fee structures, validator incentives, minting logic, and anchoring cadence—are governed via modular components.

- The Core is fixed and cannot be upgraded through normal governance
- Modules are upgradable via a decentralized two-stage process
- Validators adopt or reject modules individually, forming consensus by majority participation

This design ensures protocol evolution can occur without compromising decentralization or neutrality.

Module Upgrade Process

Module upgrades follow a transparent, decentralized two-step procedure:

Core Dev Advisory Signal (non-binding):
The Dev Fund multisig group (26 members) issues an advisory signal for each proposed module. This signal includes both:

- A technical assessment (validity, security, stability)
- A philosophical alignment check (Bitcoin ethos, neutrality)

The signal uses three public categories:

- Green – Technically valid and consistent with Bitcoin principles
- Amber – Unclear implications or in need of revision
- Red – Risks Core integrity or violates key security assumptions

The signal requires a minimum 21-of-26 consensus to be issued. It is advisory only and does not block upgrades.

Validator Approval (binding):

Validators vote on the proposed module upgrade.
A module is accepted when 80% of active validators approve it via signed signaling or verifiable majority adoption. Validators who do not adopt the module may become incompatible with the dominant DAG and lose transaction finalization eligibility until resynced.

This model ensures that Bitcoin-aligned developers provide expert review, while final control remains in the hands of decentralized BTX node operators.

Dev Fund Purpose and Keyholder Eligibility

BTX allocates 3% of all transaction fees to a development fund, dedicated to supporting:

- BTX Core maintenance and upgrades
- Bitcoin Core development
- (Optionally) approved BTX governance modules

Funds may be used for audits, infrastructure, community support, or critical fixes—subject to consensus among multisig keyholders.

To ensure alignment with Bitcoin's security model, Dev Fund keyholders must be active, verifiable contributors to the Bitcoin Core project**.** Eligibility is confirmed by validator signaling and enforced through transparent community review.

Dev Fund Quorum Selection and Rotation

Keyholders are selected using a randomized validator-driven quorum process:

- All eligible Bitcoin Core developers are placed into randomized groups of 5
- Each group selects a representative via 4-of-5 consensus
- New randomized groups are formed from selected reps
  (Group size may be adjusted from the second round onward to ensure a sufficient pool)
- The process repeats until 26 unique individuals are selected
- The Dev Fund is controlled by a 21-of-26 multisig quorum

Keyholders are re-selected every two years through this process. Full logic and timing are defined in Appendix A.

This structure ensures decentralization, resistance to collusion, and regular refreshment of multisig members.

Development Fund Cap and Anchored Transparency

The development fund is hard-capped at $50 million per year, adjusted for purchasing power. Once the cap is reached, all additional funds are redirected to Bitcoin miners to support long-term Bitcoin security.

Each Dev Fund spend is anchored to Bitcoin via a hashed commitment. This hash includes the purpose, amount, BTC-equivalent value, and recipient ID, but is obfuscated by default. Anchoring ensures irreversible timestamping and accountability without compromising operational privacy.

Full spend details may only be decrypted if:

- At least 21 validators initiate a disclosure vote, and
- A simple majority (51%) of active validators approve disclosure

This guarantees Dev Fund oversight while protecting recipients from surveillance or external coercion.

Optional BTC Anchoring Buffer and Auto-Fill

To ensure validators can reliably anchor to Bitcoin during low-fee periods, maintaining a BTC anchoring buffer (e.g., 0.01 BTC) is recommended.

While not currently enforced by the Core, anchoring requires BTC fees. Validators without a sufficient BTC balance may miss anchoring duties and be suspended from validation for 7 days. As a result, maintaining a buffer is a standard best practice.

Validator software includes an optional auto-fill tool, allowing users to convert earned MOTO into BTC automatically when their anchoring buffer drops below a user-defined threshold. This preserves validator autonomy while ensuring continuous anchoring capacity.

This behavior may be enforced by module logic in the future if required by network conditions.

BTX Core Upgrade Procedure

The BTX Core is designed to be immutable. However, a failsafe upgrade mechanism exists to support critical updates, such as urgent security patches or structural changes (e.g., quantum resistance if it cannot be handled via module).

A Core update requires:

- 21-of-26 Dev Fund multisig signal
- 95% validator approval, tracked over a 4-week window
- 30-day activation delay to allow for public review and challenge

Core updates must also maintain compatibility with Bitcoin Core and uphold BTX's decentralization guarantees. Full logic is detailed in Appendix A.

Future-Proof Governance

BTX governance is designed to evolve without compromising its principles. The modular structure allows rapid upgrades, while the Core remains immutable and minimal. Bitcoin-aligned developers provide technical guidance, while decentralized validators retain ultimate control.

No centralized treasury, foundation, or corporate actor holds governance authority. BTX is governed by those who run it—just like Bitcoin.

For technical specifications related to this chapter, see: Appendices A and J

Chapter 8 – Minting, Redemption, and BTC Anchoring Buffer

BTX operates as a Bitcoin Layer-2 where each unit of MOTO is backed 1:1 by BTC held in federated, auditable custody (see Appendix I). The design is upgradeable to a trustless peg when technically viable, without changing the user experience.

Minting BTX (MOTO)

Users mint MOTO by sending BTC to a rotated, single-use federation address. Key rules:

- Conversion rate: 1 BTC = 10,000,000,000 MOTO
- Confirmations: Minting occurs after the BTC deposit has 2 on-chain confirmations
- Receiver: The BTX receiving address is provided in OP_RETURN or associated metadata; if absent, the user's connected BTX wallet is used
- Recording: The mint is issued on BTX with a record linking to the BTC txid and peg version

Minting is executed against federated multisig reserves with public proof-of-reserves and periodic external audits (Appendix I). A flat mint/burn fee applies, stabilized by the PP-Oracle (Appendix L).

Redeeming BTC

To redeem, the user burns MOTO on BTX and supplies an encrypted BTC destination address:

- Granularity: Redemption amounts correspond to whole satoshis (e.g., 100 MOTO)
- Processing: Upon burn finalization and module checks, the federation releases BTC from the reserve pool to the user's address; the BTC transaction then confirms on-chain
- Neutrality: Any MOTO holder may redeem, regardless of who minted it

One-Way Independence

Minting and redemption are decoupled:

- The minter need not be the redeemer

- Any MOTO can be redeemed once burned 1:1
- BTC is released only when the corresponding MOTO is burned

This preserves censorship resistance and removes identity linkage.

RCAI (Redemption Contract Aggregator & Interface)

Redemptions route through RCAI, which:

- Selects the optimal redemption route across available pools by liquidity, cost, and responsiveness
- Provides automatic fallback if a route fails, retrying every 30 seconds until completion
- Exposes an upgradeable interface so new vault types or contract backends can be added without Core changes

Anchoring to Bitcoin

BTX periodically anchors a DAG snapshot hash to Bitcoin via OP_RETURN. The anchor includes:

- Finalized transaction set digest
- Minimal metadata (e.g., TX count, anchor index, prior-anchor ref)
- Hashed Validator ID for accountability

Anchoring cadence scales with volume (Appendix M):

- < 1,000 TX/day: every Bitcoin block
- 1,000–10,000 TX/day: every 3rd block
- > 10,000 TX/day: every 12th block

Hysteresis rules prevent flapping and preserve predictability. Anchors act as public finality checkpoints.

BTC Anchoring Buffer

A single validator is randomly selected each round to publish the anchor. If the selected validator fails to anchor within the duty window, a new one is selected; missed duty while online triggers 7-day suspension (Appendix M).

To ensure reliability:

- Anchoring buffer: Validators maintain a small BTC balance to cover anchor fees
- Auto-fill (optional): Validator software can auto-convert earned MOTO → BTC when the buffer dips below a user-defined threshold
- Anchor Gas Credit (AGC, module): A small stipend may accrue from fees to cover expected BTC fee for the next anchor (Appendix M)

Security and Auditability

- Federated custody: Jurisdictionally distributed multisig; threshold signing for releases
- Open auditing: Public custody addresses, on-chain proofs of reserves, and periodic external audit reports
- Anchored state: Regular peg state digests and anchor trail enable independent verification

Upgrade Path to Trustless Peg

When viable, the federation backend is replaced by a trustless, cryptographic peg:

- Mint/burn interfaces remain unchanged
- Reserves migrate under the new mechanism without user impact
- Auditing shifts from custody proofs to protocol proofs (Appendix I)

For technical specifications related to this chapter, see: Appendices I, M (and L for PP-stabilized fees).

Chapter 9 – Fee System Architecture and Modular Distribution

BTX uses a simple, percentage-based fee system to ensure sustainability, security, and affordability. This system is designed to scale with transaction value, remain stable in real purchasing power, and incentivize all key participants without inflating protocol complexity.

All fee logic is handled outside the Core. The Core only enforces the presence of a transaction fee and the 3% allocation to the development fund. All specific values, limits, and distributions are enforced through upgradeable governance modules.

Transaction Fee Structure

Each transaction pays a fee based on its value:

- Default: 0.25% of transaction value
- Reduced: 0.10% once network volume exceeds 10,000 transactions/day
- The switch is automatic and handled by module logic

This proportional model ensures low fees for microtransactions while scaling appropriately for higher-value payments.

Wallets automatically calculate and apply the correct fee based on the active network policy defined by modules. There is no advantage to overpaying, and manual fee input is not required.
If a user submits a transaction with a fee below the required amount (due to outdated logic or conditions), the validator will reject it. The wallet is expected to detect this and automatically retry using the correct fee. Validators never modify or adjust underpaid transactions and enforce policy strictly based on module consensus.

Fee Floors and Caps

To protect the network from spam and prevent excessive costs for high-value transfers, BTX uses adjustable minimum and maximum fee limits:

- Minimum fee: 5,000 MOTO (≈ $0.06 at BTC = $120,000)
- Maximum fee**:** 2,100,000 MOTO (≈ $25.2 at BTC = $87,000)

These boundaries are dynamically adjusted based on a 5-month rolling BTC/USD average, enforced by module logic. This ensures long-term consistency in purchasing power without requiring Core changes.

AI Wallet Batching Discounts

AI wallets can bundle up to 21 transactions into a single DAG entry to reduce network load and fee burden. Discounts are structured as follows:

- Base per-TX fee: 250 MOTO

- Batching discounts:
    - 3 TXs → 1%
    - 6 TXs → 2%
    - 12 TXs → 4%
    - 18 TXs → 6%
    - 21 TXs → 7%

This incentivizes batching, improves DAG density, and supports efficient AI-to-AI microtransactions. Discount rules are defined in the AI wallet module and may evolve through governance.

Minting and Redemption Fees

Conversions between BTC and BTX are subject to a flat fee:

- Mint/Burn Fee: 100,000 MOTO (≈ $1.2 at BTC = $87,000)

This prevents high-frequency flipping between BTC and MOTO while remaining low enough for general user activity. The flat fee is adjusted for purchasing power and enforced by the same modular system that governs transactional fees.

Fee Distribution Structure

Transaction fees are split across three key roles:

- 82% to active BTX validators
- 15% to Bitcoin miners
- 3% to the capped development fund

Validator fees are evenly distributed across the 21-node processing pool that validates and confirms the transaction. The finalizer pool (which handles global timestamp finality) does not receive a fee share.

Fee distribution is enforced entirely by a modular component. This allows adjustments to splits or reward timing through decentralized governance.

Dynamic Miner Fee Adjustment

To maintain Bitcoin security, BTX dynamically increases miner rewards if the Bitcoin network shows signs of weakening.

- If Bitcoin difficulty drops below 85% of its average over the previous 6 difficulty epochs (~12 weeks), the miner share rises linearly up to a maximum of 50%
- This safeguard ensures miners remain economically supported even as block rewards decline

The adjustment logic is handled by an external module that tracks Bitcoin's published difficulty and activates fee shifts when the trigger threshold is reached.

Development Fund Cap and Overflow Logic

The 3% fee allocated to the BTX development fund is subject to a hard cap:

- Maximum: $50 million per year (adjusted for purchasing power)

Once this cap is reached:

- All excess is automatically redirected to Bitcoin miners
- This is enforced by the dev fund cap module, not the Core

This ensures development funding remains predictable and limited, while also supporting Bitcoin's long-term hash

power if fund needs are met.

Modular Enforcement Overview

Fee Element	Enforced By
Fee % (0.25 / 0.10)	Fee policy module
Min / max values	Purchasing power module
AI batching rules	AI wallet module
Mint/Burn fee	Redemption module
Fee split (82/15/3)	Fee-split module
Miner boost (difficulty)	Difficulty-tracking module
Dev fund cap	Cap management module

The modular system ensures that all values can evolve as needed, without ever changing the Core or compromising validator consensus. These modules are upgradeable via governance, with advisory input from the Core Dev multisig.

For technical specifications related to this chapter, see: Appendices B and L.

Chapter 10 – Privacy Architecture and Threat Resistance

BTX is built to protect user privacy by default—while preserving auditability and alignment with Bitcoin. Unlike systems that treat privacy as an afterthought, BTX embeds obfuscation, unlinkability, and surveillance resistance directly into its transaction structure and address model.

All validators are required to support and process these protections. Wallets offer a simplified interface that allows users to choose between a default “Basic Privacy” setting and an optional “Privacy Mode” for enhanced protection.

Default Privacy Mechanisms (Enforced by Network)

The following privacy protections are always active at the protocol level and apply to all transactions. Protections differ slightly depending on whether the transaction is internal to BTX or crosses between BTC and BTX.

MOTO-MOTO Transactions (BTX Internal Transfers)

These occur entirely within the BTX Layer-2 system and benefit from the following protections:

- One-time encrypted addresses:
  Each recipient address is encrypted, used once, and invisible to observers or validators. This breaks linkability between sender and receiver.
- Address recycling:
  Once a MOTO address is fully emptied and inactive for a defined timeout, it is recycled into the available pool. Reuse is randomized and unlinkable to prior owners.
- Role blending:
  Active, inactive, and dummy addresses all share the same format and namespace. This prevents external observers from determining whether an address belongs to a real wallet or a placeholder.

These features ensure that MOTO-MOTO transactions cannot be clustered, mapped, or correlated using standard

blockchain analysis techniques.

Mint and Burn Transactions (BTC Entry/Exit)

These handle conversion between BTC and MOTO and include the following protections:

- Minting (BTC → MOTO):
  Each mint transaction requires BTC to be sent to a rotated, single-use Bitcoin address.
    - There is no address reuse
    - The connection between the BTC transaction and the resulting MOTO issuance is not publicly visible
- Burning (MOTO → BTC):
  The user submits a burn transaction that includes a fully encrypted Bitcoin destination address, known only to the validator processing redemption.
    - No external observer can link the redeemed BTC to a specific MOTO wallet
    - Even a compromised validator cannot trace the full flow unless both sides are under its control

These protections ensure that minting and redemption flows do not expose user identity or traceability at the Bitcoin Layer-1 level.

Wallet Privacy Modes

To keep the user experience simple while offering advanced protection when needed, BTX wallets support two privacy modes:

- Basic Privacy (default):
  Enabled by default for all users, this mode ensures strong address-level privacy and unlinkability without requiring configuration. It includes one-time encrypted addresses, address recycling, role blending, and obfuscation of mint/redeem flows.
- Privacy Mode (optional):
  This enhanced setting applies additional measures for users operating in high-risk or censorship-prone environments. It activates additional wallet-side obfuscation and transmission logic while retaining the protections of Basic Privacy.

Feature Comparison by Mode

Feature	Basic Privacy (default)	Privacy Mode
One-time encrypted addresses	Enabled	Enabled
Decoy (dummy) address mixing	Disabled	Enabled
Address recycling	Enabled	Enhanced
Role blending	Enabled	Enhanced
Obfuscated mint addresses	Enabled	Enabled
Encrypted burn/redeem address	Enabled	Enhanced
Dummy transaction insertion	Disabled	Enabled
Timestamp masking and compression	Disabled	Enabled
Privacy scoring and visibility (wallet)	Hidden	Visible

Feature	Basic Privacy (default)	Privacy Mode
AI wallet delay and pattern obfuscation	Disabled	Enabled

Threat Resistance by Mode

Basic Privacy defends against:

- On-chain address linking
- DAG-level transaction clustering
- KYC-exchange linkage to MOTO addresses
- Passive blockchain surveillance
- Mint/redeem trail mapping

It is suitable for:

- General daily use
- Merchants and remittance users
- Most standard users in moderate environments

Privacy Mode adds resistance to:

- Real-time traffic inspection
- Network-level metadata analysis
- Behavioral pattern extraction
- Timing attacks
- Surveillance in hostile jurisdictions

Summary

BTX achieves robust default privacy through mandatory validator support for encrypted addresses, obfuscated mint/burn operations, and address unlinkability. All users benefit from strong baseline protections without any configuration.

For users with higher security needs, BTX wallets offer a single “Privacy Mode” toggle that activates additional protections like dummy transaction injection, timestamp masking, and AI-based TX delay logic. These features are applied at the wallet level and do not require network-level changes or third-party support.

BTX ensures that privacy is not a patch—but a foundational property of the system.

For technical specifications and future enhancements, see: Appendices K and O.

Appendix A – Core Logic Summary and Modular Interfaces

This appendix summarizes the critical logic enforced directly by the BTX Core and clarifies how modular components interface with that logic.

The BTX Core is designed to be minimal, immutable, and resistant to upgrade churn. It enforces the absolute minimum logic necessary for network integrity and Bitcoin anchoring. All extended behavior is handled through upgradeable modules governed by validator consensus and Core Dev advisory signals.

1. Core-Enforced Logic

The following behaviors are always enforced at the protocol level by BTX Core:

Transaction Structure:

- All transactions must reference two parent transactions (for DAG integrity)
- Each transaction must include a valid signature, timestamp, and a transaction fee
- Transactions are finalized by a rotating 21-validator pool
- Transactions must be timestamped and appear in chronological order
- Double spends are invalid and rejected

Anchoring:

- A snapshot of the DAG is anchored to Bitcoin via OP_RETURN hashes
- Anchoring frequency is determined by network volume (via module)
- A single validator is randomly selected to anchor each round
- If the validator does not anchor within 1 minute, a new one is selected
- Validators that fail anchoring duty are suspended from validation for 7 days

Redemption (Burning):

- Redemption TXs must match 1:1 with BTC (1 BTC = 10,000,000,000 MOTO)
- Redemption amounts must convert to whole satoshis
- Redeemed BTC is released only after 2 Bitcoin confirmations
- Burn TXs contain an encrypted BTC address visible only to redemption module

Minting:

- MOTO is issued only after BTC has been received in a valid, one-time BTC mint address
- Each mint address is single-use and is tied to a corresponding issuance record
- Validators confirm the on-chain receipt before authorizing MOTO issuance
- MOTO is credited to the receiving wallet specified in the mint request (auto or manual) only after 2 Bitcoin confirmations

Human Wallets:

- May only send or receive full MOTO values (≥ 1 MOTO)
- Sub-MOTO values are finalized normally but are not displayed in wallet UI until the balance ≥ 1 MOTO

I Wallets:

- May send and receive sub-MOTO values
- Must comply with batching limits (maximum 21 TX per batch)
- Transactions from AI wallets are subject to validator-enforced micro-fee validation rules
- Batching logic itself is handled via module, but fee enforcement is enforced by Core

Address Recycling:

- Any MOTO address that has been emptied and remains inactive beyond a defined timeout is marked as recyclable
- Recycled addresses are returned to the available pool and cannot be linked to prior ownership
- Core enforces that recycled addresses cannot be re-associated with their history

Dummy Transactions vs. Dummy Addresses:

- Dummy Transactions: Valid TXs with no economic content; included to improve DAG density or add noise. Treated identically to real TXs and finalized as normal
- Dummy Addresses: Decoy recipient addresses included in Privacy Mode to obscure real recipients; indistinguishable from valid recipients but are discarded by the receiving wallet module
- Validators must support both, but only process real recipients and properly formed TXs

These Core-level rules are fixed and cannot be bypassed, disabled, or altered by module logic.

2. Modular Interfaces (Governance-Controlled Behavior)

The following functions are handled outside the Core, by consensus-approved modules. These modules extend validator behavior without altering base-level consensus.

Fee module:

- Defines the current transaction fee rate (0.25% or 0.10%)
- Applies min/max values adjusted for purchasing power
- Sets the flat fee for mint and redemption TXs

Fee split module:

- Splits fees between validators, Bitcoin miners, and the development fund
- Dynamically adjusts miner share based on Bitcoin difficulty
- Applies modular scaling logic to adapt to changing conditions

Validator pool module:

- Defines how validators are selected and rotated
- Handles pruning windows and finality rules
- Ensures consensus around transaction state across the pool

Redemption module:

- Verifies burn TXs and encrypted BTC destinations
- Executes RCAI fallback logic if a contract becomes unresponsive

- Attempts new redemption paths every 30 seconds until success

AI wallet module:

- Applies batching structure and discount logic
- Defines transaction aggregation behavior and optional timing obfuscation
- Manages micro-fee tiers and ensures compliance with fee floor enforcement

Privacy module:

- Handles dummy address generation and decoy mixing
- Enables timestamp masking, TX bundling, and compression
- Activates obfuscation logic for users who enable Privacy Mode
- Works in conjunction with wallet-side settings; not mandatory for human wallets

Development fund module:

- Enforces 3% allocation cap, adjusted for purchasing power
- Redirects overflow to miners if the cap is reached
- Anchors dev fund spending to Bitcoin via OP_RETURN hashes
- Allows validator-triggered voting (≥ 51%) to reveal spending hashes in case of suspected misuse

Anchoring Schedule Module:

- Anchoring cadence is determined by a rolling 4-day average of BTX transaction volume (TX/day).
- Thresholds and rules are defined as follows:

Volume Level	Anchoring Frequency
< 1,000 TX/day	Anchor every Bitcoin block
1,000–10,000 TX/day	Anchor every 3rd Bitcoin block
> 10,000 TX/day	Anchor every 12th block

Transition Rules:

- Downward Transition (Reduce Frequency):
    - Rolling average must exceed the next threshold by at least 5%
    - Condition must persist for two consecutive rolling windows (8 days total)
    - Prevents frequency shifts due to temporary spikes
- Upward Transition (Increase Frequency):
    - If the rolling average drops below the current cadence's threshold, anchoring frequency increases immediately
    - Prioritizes security and auditability during network slowdowns or outages

This logic is governed by module and may be adjusted via validator-approved governance updates. The Core enforces anchor submission deadlines and validator penalties but does not determine cadence.

All modules are versioned, signed, and governed by validator consensus. Proposals are evaluated by the Core Dev group, which may provide non-binding signals to guide voting decisions.

3. Core Dev Advisory Signal

The Core Dev group is selected via a randomized 4-of-5 → 4-of-5 → 4-of-5 election process, ensuring decentralization and resistance to capture. Their role includes:

- Reviewing module code and upgrade proposals
- Issuing audit signals:
    - Green: secure and recommended
    - Yellow: functional but questionable
    - Red: risky, insecure, or poorly designed

The advisory signal has no enforcement power but serves as a quality filter for validator governance.

4. Summary

BTX's Core defines strict, minimal behavior required for network operation, DAG finality, Bitcoin anchoring, and transaction integrity. All other system logic—governance, privacy, fees, batching, redemption, and more—is handled by upgradeable modules.

This architecture ensures:

- A stable, unchanging Core with minimal attack surface
- Rapid adaptability via modular upgrade paths
- Clean separation between consensus, governance, and wallet-level logic

BTX's modular architecture ensures long-term resilience and adaptability without compromising Core-level trust or security.

Appendix B – Fee Structures and Conversion Units

This appendix provides an overview of BTX's fee logic, conversion ratios, and wallet-level rules. All fee-related policies are modular and can be updated without changes to the BTX Core.

1. Base Transaction Fee Logic

BTX uses a simple percentage-based fee model that applies to all MOTO-denominated transactions.

- Default transaction fee: 0.25%
- Reduced transaction fee: 0.10% (triggered once volume ≥ 10,000 transactions/day)
- Min fee floor: 5,000 MOTO (≈ $0.06 at BTC = $120,000)
- Max fee cap: 2,100,000 MOTO (≈ $25.2 at BTC = $120,000)

The floor and cap values are adjusted based on a 5-month rolling BTC/USD average to maintain real-world purchasing power. Fee switching and adjustments are handled via a modular policy component.

2. Minting and Redemption Fees

To discourage high-frequency flipping between BTC and BTX while maintaining affordability, mint and burn operations apply a flat fee:

- Minting and redemption fee: 100,000 MOTO (≈ $1.2 at BTC = $12,000)
- This fee is dynamically adjusted for purchasing power via the same module logic as regular TX fees

The minting module ensures MOTO is only issued after 2 Bitcoin confirmations. Redemption also requires 2 confirmations before BTC is released.

3. AI Wallet Micro-Fee and Batching Rules

AI wallets may batch up to 21 transactions into a single DAG entry. Each individual TX in the batch pays a fixed fee of 250 MOTO, reduced by volume-based discounts:

Batch Size	Discount	Effective Per-TX Fee
3 TX	1%	247.5 MOTO
6 TX	2%	245.0 MOTO
12 TX	4%	240.0 MOTO
18 TX	6%	235.0 MOTO
21 TX	7%	232.5 MOTO

These rules are enforced by validator logic and defined in the AI wallet module. Human wallets are not eligible for batching or sub-MOTO transactions.

4. Unit Definitions and Conversions

A. Bitcoin (Layer-1 Units)

- 1 BTC = 100,000,000 SAT
  (Standard Bitcoin base unit)

B. BTX (Layer-2 Units)

- 1 BTC = 10,000,000,000 MOTO
  (MOTO = BTX base unit)
- 1 SAT = 100 MOTO
  (Conversion between BTC Layer-1 and BTX Layer-2)
- 1 NAK = 1000 SAT = 100,000 MOTO
  (Human-readable pricing unit, similar to fiat base units)

C. Denomination Purpose

- MOTO = fundamental unit, used in all settlements
- SAT = intermediate denomination (used by wallets and pricing logic)
- NAK = top-layer unit for retail display and adoption (≈ USD value anchor)

All conversions are fixed by design and cannot be modified. Wallets may use different display preferences but all settlement occurs in full MOTO units.

5. Human Wallet Rules

- Send and receive limit: Human wallets may only send and receive full MOTO amounts (≥ 1 MOTO)
- Sub-MOTO transactions: Not permitted under any circumstance
- Rationale: This ensures clean address recycling, consistent UX, and eliminates unusable fractional balances

All validation is handled by network validators. Human wallet enforcement is strict and applies to all DAG-level transactions.

6. Summary

All BTX fee rules and unit conversions are modular and enforced by validators via signed module versions. Wallets calculate fees automatically—manual entry is not required or supported.

This design ensures:

- Predictable and affordable fees for users
- Long-term purchasing power stability
- Seamless compatibility between Bitcoin and BTX units
- Fully automated wallet behavior with strong UX guarantees

Appendix C – Validator Rules and AI Wallet Logic

This appendix outlines the responsibilities, constraints, and processing logic enforced by BTX validators, including the unique handling of AI wallets and batching logic.

All validator rules are enforced in coordination with active modules and are essential to maintaining consensus and DAG integrity.

1. Validator Responsibilities

All BTX validators must enforce:

- Transaction structure and signature validity
- Timestamp sequencing and DAG linkage
- Compliance with minimum fee and batching rules
- Address validation and recycling enforcement
- Anchoring duties based on randomized selection
- Handling of dummy TXs and staged offline transactions

Validators rotate through a 21-node confirmation pool, which verifies all submitted transactions. A separate 21-node finalizer pool resolves timestamps and network-wide finality windows.

2. Anchoring Duties

- One validator is randomly selected to anchor the current DAG state to Bitcoin using OP_RETURN
- If the selected validator fails to anchor within 1 minute, another is selected
- A validator who fails their anchoring duty is suspended from all validation roles for 7 days

Validators must maintain sufficient BTC balance to cover anchoring costs. This is encouraged via a BTC anchoring buffer, typically filled by auto-converting MOTO fees to BTC if needed.

3. Offline Transaction Handling

- Staged (offline) transactions must reference two valid DAG parents
- The transaction's timestamp must be strictly later than both referenced parents
- Validators do not enforce global timestamp order, only local consistency
- If the referenced parents have been pruned, the first connected wallet or validator resubmitting the transaction will attach new valid parents
- Once broadcast, the staged TX is finalized like any other

4. Dummy Transactions vs Dummy Addresses

- Dummy Transactions:
  Structurally valid transactions with no economic value, used to increase DAG density or obfuscate activity patterns. Finalized and stored like normal TXs.

- Dummy Addresses:
  Included in Privacy Mode to mask the real recipient. Only one address per TX is real; the rest are encrypted decoys. Validators finalize all recipient fields, but only valid wallets will accept the correct one.

5. Pruning and Finality

- Transactions reach finality after 100 DAG confirmations
- Transactions become eligible for pruning after 125 confirmations
- Burn transactions may be pruned after the corresponding BTC has been released and the TX included in an anchored snapshot

This keeps validator storage efficient while preserving auditability and finality windows.

6. AI Wallet Transaction Rules

- Validators must confirm the AI wallet has:
  - Completed its initial setup (via computational puzzle)
  - Not exceeded its randomized revalidation interval
- Sub-MOTO TXs are accepted only if properly formatted and batched
- Maximum batch size is 21 transactions
- Minimum fee per TX is 250 MOTO, with discounts applied based on batch size
- Transactions exceeding limits or violating batching rules are rejected

7. Human Wallet Rules

Updated Rule:

- Human wallets may only send and receive full MOTO values (≥ 1 MOTO)
- Any attempt to receive sub-MOTO amounts into a human wallet is invalid and rejected by validators
- Sub-MOTO values are reserved for AI wallet infrastructure only
- Ensures address retirement and recycling function correctly

8. Fee Validation Behavior

- Validators enforce minimum and maximum fee bounds based on active fee module
- If a transaction fee is below the required minimum, the TX is rejected
- Wallets are expected to auto-calculate and reattempt using the correct fee
- Fee floor and cap values are adjusted for purchasing power using a rolling BTC/USD average

9. Dummy TX Compression and Bundling

- Validators must accept timestamp-fuzzed, compressed, and bundled TXs created by privacy-enabled wallets
- Compression logic is handled by the wallet; validators only check structure and validity

- Dummy TXs are finalized like normal transactions and stored for pruning after standard confirmation depth

10. Summary

BTX validators enforce a strict set of rules that support:

- Transaction finality and DAG consistency
- Anchoring discipline and economic incentives
- High-throughput batching for AI agents
- Robust offline and obfuscated transaction handling
- Clean separation between AI and human wallet behavior

These rules ensure scalability, privacy, and long-term validator sustainability—while simplifying the user experience and preserving modular flexibility.

Appendix D – Privacy Modules and Obfuscation Layers

This appendix provides an overview of BTX's privacy-enhancing logic and the modular infrastructure that supports it. All privacy features are validated by the network and supported by all validators, but wallets may expose different privacy levels based on user needs.

1. Network-Level Privacy (Always Enforced)

All BTX transactions benefit from the following default protections:

- **One-time encrypted recipient addresses**
  All MOTO-MOTO TXs use encrypted destination addresses derived by the receiving wallet and used only once.
- **Address recycling and unlinkability**
  Once emptied and inactive, addresses are recycled and reassigned to unrelated users, preventing long-term ownership tracking.
- **Role blending**
  Dummy, real, and recycled addresses are indistinguishable on-chain, neutralizing pattern analysis.
- **Mint and burn obfuscation**
  - Minting uses rotated, single-use BTC addresses
  - Redemption TXs contain fully encrypted BTC payout addresses
    This prevents tracking of Layer-1 to Layer-2 flow.

These protections are protocol-level and cannot be disabled or bypassed by validators or third parties.

2. Privacy Modes (Wallet-Level Control)

BTX wallets allow users to select their preferred level of privacy:

- **Basic Privacy (default):**
  - Applies one-time encrypted addresses, recycling, and role blending
  - Obfuscates mint/redeem behavior by default
  - Does not include decoy or dummy TX logic
- **Privacy Mode (optional):**
  - Adds dummy address mixing
  - Enables dummy TX injection
  - Applies timestamp masking and relay compression
  - Activates randomized AI wallet delays
  - Exposes privacy score metrics in the wallet UI

Wallets never expose individual feature toggles—only a mode selector. All privacy features are validated and supported by the BTX validator network.

3. Dummy Address Mixing Logic

In Privacy Mode, each transaction includes a mix of encrypted addresses:

- Only one is valid; the rest are decoys
- All are indistinguishable and encrypted in identical format
- Decoys are recycled after short timeouts
- Wallets accept only matching recipient keys and ignore the rest

Validators finalize all recipient fields without knowing which is valid.

4. Dummy Transaction Handling

Dummy TXs are valid, signed transactions that contain:

- No net economic transfer
- Obfuscated timestamps
- Standard structure to mimic real transactions

They are used to:

- Increase DAG density (especially in low-volume periods)
- Disrupt traffic analysis and timing-based inference
- Mask wallet activity in Privacy Mode

Dummy TXs are inserted probabilistically and are finalized normally by validators.

5. Timestamp Obfuscation and Compression

Wallets in Privacy Mode may:

- Apply small randomized timestamp shifts
- Compress multiple TXs into bundles
- Send TXs in staggered or delayed sequences

Validators accept these as long as timestamp order is valid relative to parent TXs.

6. AI Wallet Obfuscation Logic

In Privacy Mode, AI wallets may:

- Delay transmission of TX batches
- Randomize batch composition and size
- Interleave dummy and real transactions
- Split logical payments across time intervals

These behaviors are designed to obscure algorithmic payment patterns, preventing clustering or usage profiling.

7. Relay Compression and Payload Masking

Privacy Mode wallets may transmit:

- Compressed or obfuscated payloads
- Slightly randomized metadata fields
- Dummy TXs intermixed with valid data

All payloads conform to standard formats. Validators are not required to distinguish between real and dummy packets—they finalize all valid TXs.

8. Future Privacy Extensions

BTX's privacy modules are upgradeable and versioned. Future improvements may include:

- zk-SNARK–based batching for AI agents
- Decentralized TX routing overlays (e.g., via mesh, SMS, or delay relays)
- Encrypted TX metadata fields (beyond recipient)

All upgrades require validator approval and Core Dev advisory review.

9. Summary

BTX privacy is designed as a layered, always-on system:

- Validators support all privacy logic by default
- Users may choose between Basic and Enhanced (Privacy Mode) settings
- Dummy TXs, dummy addresses, and timestamp obfuscation operate without user configuration
- Modular upgrades enable evolving protections against future surveillance models

This model ensures that all BTX users benefit from strong baseline privacy, while those who need maximum resistance can enable additional protections with a single toggle.

10. Incentives and Responsibility for DAG Health:

BTX relies on wallets—especially AI wallets—to maintain DAG density during low-volume periods. These wallets assess network conditions using broadcast metadata and local heuristics (e.g., anchoring frequency, recent volume). When DAG thinning is detected, wallets may insert dummy transactions to preserve finality speed and fee efficiency. These dummy TXs are voluntary and pay standard fees. This behavior is governed entirely by the privacy and AI wallet modules, allowing future enhancements such as fee pooling, density-based rebates, or obfuscation incentives—without affecting the BTX Core.BTX privacy is designed as a layered, always-on system:

Appendix E – Bootstrap Mode and Low Volume Handling

BTX is designed to remain functional and auditable even during periods of low transaction volume or early-stage adoption. This appendix defines the behavior of the network during bootstrap mode and outlines safeguards to ensure integrity, finality, and Bitcoin anchoring under sparse conditions.

1. Bootstrap Mode Definition

Bootstrap mode is triggered when BTX daily transaction volume falls below **1,000 transactions per day**, averaged over a rolling 4-day window. This threshold ensures that DAG density, validator incentives, and anchoring frequency remain appropriate for early network conditions.

Key bootstrap behaviors include:

- Anchoring every Bitcoin block
- Encouraged insertion of dummy transactions by AI and privacy wallets
- Optional validator support incentives for anchoring reliability
- Relaxed pruning thresholds to ensure finality visibility

These conditions are lifted automatically once volume exceeds 1,000 TX/day and remains above this threshold for two full rolling windows (8 days total).

2. Anchoring Frequency: Sliding Volume-Based Cadence

BTX adjusts its anchoring cadence dynamically based on sustained network activity:

Volume Level (4-day avg)	Anchoring Frequency
< 1,000 TX/day	Every Bitcoin block
1,000 – 10,000 TX/day	Every 3rd Bitcoin block
> 10,000 TX/day	Every 12th Bitcoin block

Transition Rules

- To reduce anchoring frequency (e.g., block → every 3rd):
    - Volume must exceed the next threshold by at least 5%
    - Condition must persist for two full windows (8 days)
- To increase frequency (e.g., every 3rd → block):
    - Transition occurs immediately if volume falls below the threshold

This asymmetric model prevents oscillation while prioritizing auditability during quiet periods.

3. Finality During Low Volume

Low transaction volume naturally slows DAG growth. To preserve fast confirmation times, BTX includes the following enhancements:

Dummy transaction injection:
AI wallets and privacy-mode wallets may insert decoy transactions with no economic value to maintain

DAG density and confirmation pace.

Encouraged validator participation:
Validators are incentivized to maintain active anchoring and relay operations even during sparse traffic. This may be enforced by future governance modules, but is voluntary at launch.

Fallback anchoring mechanism:
Anchors are chained, even during low activity, allowing for reconstruction and audit of all DAG states through time. Anchors always include:

- Snapshot hash
- Timestamp
- Prior anchor reference
- Hashed validator ID

4. Transaction Finality Estimates by Volume

Network Load	Approximate Finality Time
5 tx/s	~35 seconds
10 tx/s	~10–12 seconds
20+ tx/s	~5 seconds

These figures assume a healthy DAG with at least minimal dummy injection during low periods. BTX finality scales smoothly as volume increases.

5. DAG Stability and Dummy TX Behavior

Dummy transactions are structurally valid but carry no economic value. Their inclusion:

- Maintains DAG velocity
- Reduces confirmation latency
- Preserves anchoring cadence predictability

Dummy TXs:

- Are inserted probabilistically by participating wallets
- Are signed and finalized like normal transactions
- Are pruned after 125 confirmations unless linked to mint/burn flows

Wallets assess network conditions using heuristics (e.g., recent TX rate, anchor cadence) and determine if dummy TX insertion is helpful. This behavior is governed by wallet-side and modular logic, not enforced by Core.

6. Emergency Anchoring Recovery

In the unlikely event that anchoring is disrupted (e.g., due to validator coordination failure), BTX includes a failsafe:

Anchoring validator reassignment: If the selected validator fails to anchor within 60 seconds, a new validator is randomly selected.

Suspension policy: Validators that fail anchoring duty are suspended for 7 days from validation roles.

Buffer auto-fill tool: Validators may enable automatic conversion of earned MOTO into BTC to maintain anchoring capacity. Anchoring requires a small BTC balance, typically 0.01–0.02 BTC.

These protections ensure that BTX remains anchored to Bitcoin even during prolonged inactivity or technical downtime.

7. Summary

BTX's bootstrap logic ensures that:

- Finality remains fast even in low-volume scenarios
- Anchors continue to be recorded into Bitcoin at high cadence
- DAG density is preserved through dummy TX incentives
- Validators remain engaged and accountable

This model allows BTX to operate securely from Day 1, without reliance on bridges, wrapped tokens, or centralized infrastructure. As volume increases, the system transitions seamlessly into high-throughput mode, preserving the same Core logic.

Appendix F - Possible Future Scaling Options at Visa-Level Throughput

The BTX protocol has been deliberately designed with a minimal and immutable Core, surrounded by a set of modular components that allow for adaptation, expansion, and optimization as network conditions evolve. While the current implementation comfortably supports early-stage and mid-scale adoption, it is anticipated that, should BTX achieve global relevance, throughput requirements may eventually approach or exceed Visa-level performance benchmarks (i.e., 5,000 to 10,000+ transactions per second). This appendix outlines potential future scaling strategies that could be implemented at that point, leveraging the modular architecture of BTX without altering its foundational principles.

1. Overview: Scaling Philosophy

- BTX Core remains unchanged regardless of future scaling needs.
- All scaling strategies are modular, meaning they can be activated, replaced, or ignored based on real-world needs and validator community consensus.
- No proposed upgrade is mandatory or on a fixed roa**dmap**; these represent possible future enhancements.

2. Scaling Milestones and Thresholds

Throughput Range	Recommended Action
0 – 2,000 tx/s	No change needed. Current validator and finalizer structure is sufficient.
2,000 – 5,000 tx/s	Begin preparing modular enhancements (finalizer split, DAG routing logic).
5,000 – 10,000 tx/s	Introduce parallel DAGs and split finalizer responsibilities where needed.
>10,000 tx/s	Activate full modular scaling architecture across all layers.

3. Validator and Finalizer Role Evolution

A. Validator Pools

- Validators currently handle all transaction types across the entire DAG.
- This design scales horizontally and requires no role changes up to high throughput.
- Optional future specialization (e.g., AI-DAG validators) may be introduced based on hardware capability.

B. Finalizer Pools

- Present design uses a single 17-of-21 finalizer pool.
- At ~2,000–5,000 tx/s, a split role architecture can be introduced:
    - Type A (Checkpoint Finalizers): A stable, slow-rotating pool selected via randomized assignment with basic performance filters (e.g., uptime, availability). Type A finalizers may be incentivized through a share of the validator fee pool during their active term. No validator may serve in more than one role simultaneously.
    - Type B (Snapshot Builders): Scalable layer that builds Merkle roots and handles Bitcoin anchoring.

- Timestamp consistency is preserved via DAG topology; absolute timing is secondary to reference ordering.

4. Parallel DAG Introduction

- At sustained throughput >5,000 tx/s, BTX may implement parallel DAGs:
  - Each DAG handles a segment of the transaction flow.
  - Mint/burn logic remains central and universal.
  - Wallets and validators route transactions dynamically to the DAG with least congestion.
- All DAGs share the same finalizer and anchoring logic, preserving integrity across the system.

5. Dedicated DAG for AI and Machine-to-Machine Transactions

- If AI/M2M payments become dominant (millions of tx/s), a dedicated DAG (DAG-M) may be deployed.
- Validators may opt in to process this DAG due to potentially higher resource requirements.
- BTX's AI Wallet Module can be expanded to handle:
  - Sub-MOTO batching
  - Aggregation logic
  - Re-validation and compression
- DAG-M would still anchor through the global finalizer pool and share the unified MOTO supply.

6. Anchoring and Governance in Multi-DAG Mode

- All DAGs produce snapshot roots.
- Type A finalizer pool signs off on global checkpoints.
- Anchoring module combines all DAG snapshots into a single Merkle root anchored to Bitcoin.
- Governance over module activation remains with validator consensus.

7. Summary Table: Throughput vs. Scaling Actions

System Load	Action(s) Triggered
Normal (0–2,000 tx/s)	No change; single DAG, unified finalizer pool
Moderate (2,000–5,000 tx/s)	Prepare modular split for finalizers, activate DAG routing logic
High (5,000–10,000 tx/s)	Deploy 1–2 additional DAGs; begin role specialization if needed
Very High (>10,000 tx/s)	Full modular scaling: multiple DAGs, finalizer role split, DAG-specific validators

---

This appendix presents possible future enhancements based on current architectural understanding. No changes are required at present, and implementation of any strategy outlined herein is subject to validator consensus and future developments in network load and technological capability.

Appendix G – BTX- Neutral Block Template Builder Module

Overview
To support long-term decentralization, transaction neutrality, and anchoring reliability, BTX introduces a decentralized, community-driven block template builder module. This module acts as an open-source, neutral mining pool—offering Bitcoin miners the option to plug in their hashrate and receive templates that maximize fee revenue while ensuring transaction fairness.

While the builder prioritizes fee-maximizing logic, it offers selective incentives for system-critical transactions such as BTX anchors and potentially mint/burn transactions. Its purpose is not to skew market pricing, but to maintain ecosystem health and Bitcoin neutrality.

Purpose and Rationale
As Bitcoin mining infrastructure centralizes around large pools, block template construction has become a critical point of control. These pools can bias transaction selection based on regulatory, political, or commercial priorities. This introduces risks of censorship, unfair prioritization, and diminished decentralization.

The BTX-neutral template builder aims to:

- Maximize miner revenue using pure fee-first logic
- Preserve neutrality in transaction selection
- Provide a credible, open alternative to centralized block template builders
- Include discounted, high-value system-level transactions (like BTX anchors)

Fee Logic and Priority Policy

General Rule: All transactions are prioritized based on highest fee-per-byte.

Anchoring Exception: BTX anchor transactions may be included at a discounted fee level, as they secure BTX's DAG and support downstream fee generation.

Mint/Burn Discount (Optional): Governance may approve minor discounts for mint/burn transactions, reflecting their role in maintaining the BTC/MOTO peg and indirectly generating miner revenue.

Spam Filtering (Optional): The pool may implement filters (e.g., to exclude JPEG inscriptions, ordinal spam), subject to governance by BTX validators and the Bitcoin Core-aligned community.

Governance and Oversight The module's policy, fee discount thresholds, and filtering rules are governed jointly by:

BTX Validators (Bitcoin full nodes running BTX logic)

Bitcoin Core-aligned Developers, either through a multisig council or random selection

Responsibilities include:

- Updating filter and discount policies
- Managing inclusion rules for system-level TXs
- Ensuring fee neutrality and transparency

Miner Incentive Alignment This builder gives Bitcoin miners—especially smaller or solo miners—a clean, profitable alternative to major mining pools:

- Pre-built, fee-optimized templates
- Easy plug-in mining configuration
- Competitive rewards without censorship politics

As BTX grows and channels more transaction fees through Bitcoin (via anchoring and mint/burns), miners aligned with this builder stand to gain increasing revenue—making the system self-reinforcing.

Conclusion
This BTX-neutral block template builder module offers a Bitcoin-aligned, economically rational, and censorship-resistant alternative for block production. By enforcing free market prioritization and offering strategic discounts for ecosystem-critical TXs, it counters the centralization of blockspace control while ensuring miners are rewarded fairly. Anchoring neutrality, ecosystem incentives, and decentralization are all strengthened under a single, open-source framework.

Appendix H – Glossary of Terms

This glossary defines key technical and conceptual terms used throughout the BTX whitepaper to aid understanding for all readers.

Anchoring
The process of committing a cryptographic hash of the BTX transaction DAG into the Bitcoin blockchain. Provides immutability and full auditability.

AI Wallet
A type of BTX wallet used for machine or automated transactions. Supports sub-MOTO amounts, microtransaction batching, and flat fees. Includes randomized re-validation to ensure fair use.

Batching (AI)
The grouping of multiple small transactions into one signed batch by AI wallets. Reduces overall fees and network load.

Bitcoin Difficulty
A measure of how hard it is to find a new Bitcoin block. Used in BTX to determine miner fee share adjustments.

Bootstrap Mode
A temporary operational mode used when BTX network volume is low. Includes dummy transactions and simplified validator logic to ensure fast finality during early adoption.

BTX Core
The minimal, immutable foundation of BTX. Includes transaction rules, anchoring logic, fee enforcement, and privacy guarantees. All other features are handled by modules.

Burn (MOTO)
The act of destroying MOTO tokens to initiate redemption for BTC. Must include encrypted recipient BTC address.

DAG (Directed Acyclic Graph)
The structure used by BTX instead of blocks. Each transaction references two others, creating a graph of transactions that finalizes organically.

Decoy Addresses
Fake addresses included in every transaction to obscure the real recipient. Indistinguishable from real addresses due to encryption.

Dummy Transactions
Fake transactions inserted automatically by BTX Core when volume is low to maintain confirmation times and DAG health.

Encrypted Address
An address that is cryptographically obfuscated. Used in every BTX transaction to preserve privacy and prevent traceability.

Finalization Pool
A top-level validator pool that finalizes transactions after initial approval by a regular pool. Uses 17-of-21 consensus and consists of validators not currently validating regular TXs.

Human Wallet
A wallet intended for everyday users. Supports full-MOTO units only, value-based fees, and simplified UX.

Local Instant Finality
A handshake between sender and receiver wallets to confirm a transaction locally when both are online — useful in low-volume conditions.

Mint (MOTO)
The creation of BTX tokens when BTC is locked in an on-chain contract. Issued at a fixed 1 BTC = 10,000,000,000 MOTO.

MOTO
The smallest BTX unit (Layer-2). 1 SAT = 100 MOTO. Used as the base transaction and accounting unit.

Module
A dynamically upgradable component of BTX that handles logic outside the Core (e.g., fee split, validator elections, governance). Anchored and version-controlled via BTC.

NAKA
A user-friendly unit in BTX, equal to 100,000 MOTO. Intended to feel similar to fiat pricing. ≈ $1.2 when BTC = $120,000.

RCAI (Redemption Contract Aggregator & Interface)
Type: Module (upgradeable)
Purpose: Orchestrates MOTO → BTC redemptions by selecting and driving the best available redemption route (“pool”), with automatic fallback and retries. Not part of the immutable Core.

SMS Relay
The use of SMS messages to deliver staged transactions to the BTX network when internet is unavailable.

Staged Finality
The process where an offline transaction is signed and stored until one party reconnects to finalize it on the BTX network.

TBTC

An internal BTX denomination representing 1 full BTC worth of MOTO. Used for interface clarity.

Validator Pool
A group of 21 randomly selected validators that confirm BTX transactions using 17-of-21 consensus. Pools rotate continuously.

Appendix I – Peg Mechanism Design

1. Purpose

The peg mechanism enables BTX to be issued, redeemed, and maintained at a fixed exchange rate with Bitcoin (BTC). It ensures that every BTX unit in circulation is backed by an equivalent amount of BTC held in secure custody. This mechanism underpins BTX's credibility, stability, and user trust.

2. Peg Model

2.1 Interim Model – Federated Auditable Custody

Until a fully trustless peg is technically feasible on Bitcoin Layer 1, BTX will operate using a federated custody model with the following properties:

Federated Control: A multisignature arrangement across geographically and jurisdictionally distributed custodians.

Open Auditing: All custody addresses are on-chain and publicly verifiable, with proof-of-reserves procedures at defined intervals.

Transparent Mint/Burn: Every mint and burn transaction is recorded both on-chain and in the BTX DAG ledger, with cryptographic links between the two.

Withdrawal Priority: In the event of federation compromise, user-initiated redemption transactions are prioritized in processing.

2.2 Upgrade Path – Trustless Peg

The peg architecture is designed to allow an easy upgrade to a fully trustless, cryptographic mechanism when/if such a solution becomes feasible on Bitcoin. Requirements for this upgrade path include:

- Modular mint/burn interface that can switch custody backends without protocol-wide disruption.
- Versioned peg module specifications to ensure backward compatibility during transition.
- Migration procedure for federated reserves into the trustless contract without loss of user funds.

3. Minting Process

1. User sends BTC to a designated BTX custody address (federated multisig).
2. Federation verifies receipt and triggers mint transaction in BTX DAG ledger.
3. Mint record contains: BTC transaction ID, mint amount, peg version, and custody address proof.
4. User receives BTX in their wallet.

4. Burning Process

1. User initiates burn transaction in BTX DAG ledger, specifying redemption BTC address.
2. Federation verifies BTX burn and releases equivalent BTC from custody.
3. Burn record contains: BTX transaction ID, burn amount, peg version, and BTC release proof.

5. Security Measures

Geographic & Jurisdictional Spread: Custodians are located in multiple legal jurisdictions to minimize seizure risk.

Threshold Signatures: Mint/burn operations require a quorum of federation members.

Regular Audits: Periodic cryptographic proofs and external audit reports published.

Emergency Lockdown: Federation can temporarily halt mint/burn if critical breach detected, pending validator review.

6. User Experience

The peg process is transparent but invisible for end-users:

- Mint, burn, and spend transactions feel as simple as current credit card or mobile payments.
- Wallets handle peg mechanics in the background without requiring technical knowledge from the user.
- Optional advanced views allow power-users to verify peg transactions and custody reserves.

RCAI — Redemption Contract Aggregator & Interface

Type: Module (upgradeable)
Purpose: Orchestrates **MOTO → BTC** redemptions by selecting and driving the best available redemption route (“pool”), with automatic fallback and retries. **Not** part of the immutable Core.

Responsibilities

- Route selection: Choose a redemption backend based on advertised liquidity, cost, responsiveness and policy weights.
- Execution & retries: Drive the burn→release workflow; if a route fails, retry every 30s on the next best route until success.
- Neutrality: Any holder may redeem; no linkage to who minted.
- Auditability: Produce verifiable logs (event IDs, route chosen, attempt history), suitable for public audits.
- Upgradeability: Admit/retire pools and adjust policy without Core changes.

Inputs

- Burn record: Finalized BTX burn (amount in MOTO, burn txid, peg version).
- Encrypted BTC address: Destination address sealed for the executing custodian/pool.
- Policy parameters: Weighting for cost/latency/liquidity, retry interval, max attempts.
- Pool telemetry: Signed metrics from available pools (capacity, fee quotes, recent success rate, SLAs).

Outputs

- BTC txid on successful release (with confirmation status).
- Status feed for the redemption (pending → routing → releasing → confirmed / failed).

- Event log (append-only): route decisions, failures, retries, final outcome.

Behavior (summary)

1. Verify burn record and peg version.
2. Rank pools with deterministic scoring; tie-break using recent BTC blockhash.
3. Attempt release on top route; if it fails or times out, retry every 30s with the next route.
4. On success, return BTC txid; on exhaustion, surface route_exhausted.

Security & Privacy

- No plaintext addresses at rest; destination is envelope-encrypted for the executing pool.
- Least knowledge: RCAI handles amounts/telemetry and encrypted destination only.
- Deterministic policy: Prevents favoritism; signed pool metrics reduce manipulation.
- Audit trail: Hashes of RCAI event logs may be periodically anchored.

Invariants

- 1:1 redemption only after burn is finalized.
- Amounts align to whole satoshis.
- RCAI must never alter amounts or addresses; it only routes.
- On success, exactly one BTC transfer is produced per redemption.

Interfaces (suggested)

- `rcai_quote(amount_sats, urgency?) → [ {pool_id, est_fee, est_latency, capacity} ]`
- `rcai_submit(burn_txid, enc_btc_addr) → redeem_id`
- `rcai_status(redeem_id) → {state, last_attempt_at, active_pool, attempts, btc_txid?}`
- `rcai_cancel(redeem_id) → ok|locked`
- `rcai_params_get() → {retry_interval=30s, max_attempts, weights}`

Failure modes (canonical)

- `liquidity_insufficient` — no pool can satisfy the amount.
- `fee_out_of_bounds` — quoted fee violates policy/PP caps.
- `backend_unreachable` — pool failed SLA window.
- `route_exhausted` — all pools tried; operator intervention suggested.

Versioning & Governance

- Module-scoped version (e.g., rcai/v1.3).
- New pools or policy changes are module upgrades; Core stays untouched.

Notes

- Supports single-route by default; multi-route/partial fills may be enabled by policy if needed.
- Works with federated custody today; drop-in compatible with a trustless peg when available (same interfaces).

7. Governance & Oversight

- **Validator Oversight:** Peg operations are logged and subject to validator review.
- **Public Anchoring:** Periodic peg state digests are anchored to Bitcoin for immutable historical reference.
- **Upgrade Approval:** Transition from federated to trustless peg requires validator supermajority and public notice.

Appendix J – Validator Admission & Sybil Resistance

Purpose
Ensure validator sets remain open yet Sybil-resistant, with measurable cost to admission and ongoing liveness checks.

Invariants

- No identity/KYC.
- Costs scale sublinearly with network size.
- Admission signals are transparently anchorable to Bitcoin.

Admission Flow

1. **Eligibility Proof Bundle (EPB)** (wallet-signed JSON):

- PubKey, Envelope format/version
- Proof-of-Resources (POR): recent uptime attestations, bandwidth probe transcript, disk IOPS sample (self-served challenge/response signed by candidates + N existing validators).
- Stake-of-Work (SoW): verifiable compute puzzle (e.g., Argon2id work receipt ~10 CPU-minutes) with difficulty target published weekly by module.
- Optional: BTC "skin-in-the-game" micro-UTXO set (e.g., 12× 5k-sat UTXOs) to demonstrate Bitcoin liveness and fee payment capability.

2. **Randomized Review Pool**

- 21 validators sampled using recent BTC blockhash VRF.
- 16/21 must endorse within 48h.
- Endorse = signature over EPB digest + POR transcript hash.

3. **Probation Window**

- 14 days with heightened monitoring: missed-duties limit at 50% of normal suspension threshold; extra spot-latency checks.

Ongoing Anti-Sybil / Churn Controls

Jittered reselection windows (already in paper) plus: per-org cap heuristic (HLL/IPSets; soft limit enforced via scoring, not hard bans).

Misbehavior Score (MS): missed anchoring, invalid vote, late relay. MS≥T suspends 7 days.

Decay: MS halves every 14 days liveness-positive.

Interfaces

- validator_apply(epb) → pending_id
- endorse_validator(pending_id, sig)
- query_ms(pubkey) → score, decay ETA

Upgrade Path

- Module-switchable SoW to PoS-style bond only if BTC-custodied, audit-anchored, and slashed by objective faults.

Appendix K – Privacy Primitives Roadmap (Decoys / Rings / ZK)

Goal
Keep default privacy strong and add progressive options without Core changes.

Baseline

- One-time encrypted addresses, recycling, role blending, dummy TXs (already spec'd).

Phase 1 – Decoy Inputs/Outputs

- Add k-decoy outputs per TX (k adjustable; default 3–5).
- Wallet picks decoys from freshness-weighted pool.
- Validator rule: decoy fields syntactically valid; only receiver can spend correct one.

Phase 2 – Ring Signatures (Spend Unlinkability)

- Ed25519-compatible CLSAG-style rings (ring size r=5–11).
- Module exposes params: min r, mix age window.
- Auditors get aggregate linkability metrics (no per-user).

Phase 3 – ZK Balance Proofs (Optional)

- Range proofs (Bulletproofs+) for amount-hiding batches (AI wallets first).
- Constraint: proofs verified by validators must fit per-TX CPU budget (module-tuned).

Failure Modes & Fallbacks

- If proof verification exceeds budget on ≥5% of validators, revert to decoy-only path for that TX (auto retry).

Interfaces

- privacy_params_get() → {k, r, max_proof_ms}
- submit_tx(bundle) → ok | too_expensive | malformed

Appendix L – Purchasing-Power Oracle (PP-Oracle)

Purpose
Stabilize fee floors/caps and dev-fund cap to purchasing power without trusting a single data feed.

Design

- **Aggregator quorum**: 13 oracle providers (exchanges, indexers, miners), 9-of-13 threshold.
- **Report window**: every 20 minutes; each report = median(BTCUSD) over last 20m with MAD (robustness).
- **On-chain anchoring**: hourly digest of signed reports anchored to Bitcoin (hash set + merkle root).
- **Outlier guards**: per-report max delta ±2% vs rolling 24h median; beyond → dropped.
- **Grace mode**: if <9 reports, reuse last valid median up to 24h, then freeze params and surface DEGRADED to wallets.

Validator Use

- Compute PP index = EMA(α=0.1) of accepted medians (≈ 5-month effect).
- Fees/floors/caps/dev-fund cap derive from PP index deterministically.

Interfaces

- pp_read() → {pp_index, last_anchor_height, health}
- pp_submit(signed_report) (oracles only)

Security

- Collusion needs ≥9 providers; anchoring gives audit trail; clients can independently verify signatures.

Appendix M – Anchoring Liveness & Economics

Objective
Guarantee anchors land reliably while making DoS uneconomic.

Mechanics

- **Duty window**: selected validator has 60s to publish anchor; if mempool fee > target, can split anchor into compact variant (shorter metadata) to fit.
- **Adaptive stipend**: small **Anchor Gas Credit (AGC)** paid from global fee pool accumulates per validator and covers expected BTC fee for next anchor.
- **Deficit auto-fill**: optional wallet rule converts MOTO→BTC when AGC<target.
- **Failure penalties**: miss → 7-day suspension (already spec'd) and forfeit AGC to next anchorer.
- **Back-pressure limiter**: cap anchors per 10 minutes even under low volume to avoid fee spikes; module varies cadence within the allowed schedule already defined.

Interfaces

- anchor_task(assign_id, snapshot_hash)
- anchor_result(assign_id, txid, fee_sat)
- agc_balance(pubkey) → sats

Economics

- AGC target = median fee for vsize(anchor) over last 72h.
- If BTC fees spike, cadence rises only if security threshold demands (Appendix A schedule).

Appendix N – Offline Re-parenting Intent-Hash (ORIH)

Problem
Staged TXs may reference parents later pruned; need safe re-attachment without malleability risks.

Solution

- Each staged TX carries an **Intent-Hash (IH)** = H(domain_sep || sender_pk || recv_commit || amount || nonce || earliest_parent_anchor_id).
- Wallets may **re-parent** by swapping the two DAG parents **only**; the spend body stays invariant to IH.
- Validators check:
    1. IH matches TX body,
    2. New parents' timestamps < TX timestamp,
    3. Parents are finalized or active.
- Result: identical payment semantics with fresh parents; no replay to different receiver/amount possible.

Interfaces

- stage_tx(build_spec) → {tx_payload, IH}
- reparent_tx(tx_payload, new_parents) → ok | bad_parent | ih_mismatch

Parameters

- reparent_max_hops: default 4 (after which wallet must rebuild).
- stage_expiry: 72h (already spec'd).

Appendix O – SMS Relay Incentives & Ops

Goals
Make SMS relays abundant, honest-forwarding, and economically neutral.

Relay Model

- **Stateless forwarders**: receive compressed TX bundles, forward to at least 3 validator ingress nodes.
- **Time windows**: 7–10 randomized relay slots/day, ~15 min each (already spec'd), plus "surge mode" if internet partition detected (oracles/validators signal PARTITION).

Fees & Incentives

- **Base relay fee**: 11,500 MOTO per message (module-adjusted by PP-oracle).
- **Proof-of-Forward (PoF)** credit: if a relayed TX finalizes within 10 minutes, the relay can claim a PoF rebate (e.g., 30–50% of fee) by presenting the relayed bundle hash + validator receipt signature.
- **Abuse limiters**: per-SIM/per-window cap; fee escalates if relay drops below 70% PoF success over last 24h.

Security & Privacy

- Messages contain ephemeral relay tokens (one-time MACs) to prevent reuse; no sender identity.
- Relays learn only bundle size and coarse timestamp; payload encrypted.
- Validators accept multiple duplicates; first-seen wins, others ignored.

Interfaces

- relay_offer(capacity, window_id) → accepted/denied
- relay_submit(bundle, mac) → ack_id
- relay_claim_pof(ack_id, anchor_ref) → rebate or reject

Ops Guidance

- Recommend multi-modem gateways, diverse carriers.
- Publish reference docker with autoscaling window timers, PoF claim agent, and metrics export.
