SPECIFICATIONS DOCUMENT

BTX: A Peer-to-Peer Computational Settlement System

BTX Developers
team@btx.dev

Abstract. Computational finance is the execution and settlement of financial obligations among participants that may not share an operator, a jurisdiction, or a prior trust relationship. In that environment, settlement rules must be machine-verifiable, spend authority must be delegable without discretionary approval at execution time, and the settlement surface must remain stable across participant classes. BTX is a peer-to-peer settlement system designed for that environment. The protocol combines matrix-multiplication proof-of-work, post-quantum spend policies, shielded settlement with selective disclosure, privacy-preserving relay, and a bridge-first layered architecture. The base layer is intentionally narrow: it orders settlement instructions, enforces spend conditions, maintains transparent and shielded settlement state, and publishes a public work benchmark through its difficulty process. Higher-layer systems such as bank ledgers, exchanges, bridge operators, and agent coordination services settle against that base without sharing a global runtime. This paper specifies the principal settlement model and transaction format, shielded settlement model, work construction, relay and admission interfaces, incentive surface, layered-settlement design, and inherited longest-chain security assumptions. Context and motivating use cases are described separately in the companion BTX Context Document [1].

1. Introduction

Administrator-dependent financial systems fail predictably at the margin. Access conditions vary by participant, institution, and jurisdiction, and those variations accumulate into architecture. BTX removes that discretionary layer at the point of final settlement. Whether submitted value reaches final settlement depends on protocol-valid spend conditions and accumulated work, not on administrative preference.

Existing digital-asset systems broadly follow one of two models. The first retains a narrow base-layer settlement model and leaves privacy, key evolution, and application design to adjacent systems. The second exposes a broad shared runtime and asks applications to share a global fee market and execution environment. BTX adopts a third model: the base layer remains narrow, but privacy, bounded authority, post-quantum spend control, and layered settlement interfaces are treated as first-class protocol concerns.

The protocol components used by BTX predate BTX individually. Matrix-multiplication useful work, post-quantum signatures, confidential-transaction techniques, stem-and-fluff relay, covenant-style constraints, and oracle-signature conditions each exist as prior research results, standards, or implementation families [2–6]. BTX assembles them into one public settlement layer with a specific operating profile:

- final settlement remains on a longest-work chain;
- spend authority is expressed through constrained post-quantum output policies;
- confidential settlement is native rather than an overlay;
- the relay layer reduces passive origin inference;
- higher-layer operators settle above the chain rather than inside one shared runtime.

This architecture is aimed at participants crossing trust, accounting, or jurisdictional boundaries: institutions, exchanges, bridge operators, market-makers, software services, and autonomous agents. The protocol makes one commitment that applies to all of them: the rules of settlement are identical for every participant. Any participant satisfying the spend conditions can settle. Any participant not satisfying them cannot.

2. Settlement Objects and Spend Policies

A BTX transaction is the encoded form of a settlement instruction under a constrained, machine-readable spend policy. Public BTX networks standardise on witness-version-2 Pay-to-Merkle-Root (P2MR) outputs. A P2MR output commits to a compact Merkle tree of authorised spend paths. The default profile contains an ML-DSA primary leaf for routine use and an SLH-DSA backup leaf with independent cryptographic assumptions. Optional leaves encode template-constrained or oracle-assisted conditions. The grammar is intentionally narrow.

The narrowness of the grammar is a security property. At the point where value moves, both human and machine implementations benefit from a small set of explicit spend forms rather than an open-ended execution environment. Output meaning is committed at creation time. Counterparties, operators, and validating nodes can evaluate the same object without importing application-specific semantics.

The current default leaf roles are:

Leaf	Role
ML-DSA-44	Primary spend path; routine and agent use
SLH-DSA-128s	Recovery path; independent assumptions
CTV leaf	Template-constrained vault or bridge path
CSFS leaf	Oracle-verified conditional close
Threshold	Multi-party coordinated custody

Both signature algorithms are post-quantum standards: ML-DSA per FIPS 204 and SLH-DSA per FIPS 205 [5, 6]. BTX adopts them from genesis rather than treating quantum migration as a later upgrade problem. Operators subject to

quantum-readiness requirements therefore build on a system whose spend surface is already aligned with the current standards trajectory.

To receive funds, a participant generates a post-quantum key pair and derives an address. To spend, the participant satisfies one of the committed leaf conditions. Validation does not consult an operator, registry, or institutional approval surface. Either the path is valid under consensus or it is not.

The spend policy is therefore the unit of economic authority. It is delegated in advance, bounded in advance, and verified at spend time by consensus. This matters for autonomous participants in particular. A principal can commit a financial scope before deployment; counterparties can verify that scope at settlement time; no one needs to rely on out-of-band approval in the middle of execution.

3. Shielded Settlement and Selective Disclosure

BTX includes a shielded settlement pool active from genesis. The implementation is built on the SMILE v2 single-round lattice confidential transaction design [1]. The protocol supports three spend surfaces:

- transparent-to-shielded deposit;
- shielded-to-shielded transfer;
- shielded-to-transparent unshielding.

Transparent-to-shielded entry is a proofless deposit path. No zero-knowledge proof is required at entry. This keeps deposit costs low and makes the path suitable for high-throughput inflows, while preserving the observability of the depositing transaction at the pool boundary. Inside the pool, sender, receiver, and amount are concealed from general observers. The network enforces value conservation and prevents double-spends through note commitments, one-time nullifiers, and confidential proofs.

BTX also includes a shared-ring `BATCH_SMILE` path for bridge-scale inflows. Throughput across these paths is bounded primarily by proof-serialisation size rather than verifier cost. That design leaves room for later compression and capacity improvements without requiring a consensus redesign of the shielded model itself.

The shielded state model is designed so that recovery depends on chain data rather than permanent wallet retention of full note history. An account registry commits shielded account-leaf payloads into consensus state. Witnesses are compact: leaf index, commitment, and sibling path. Full nodes reconstruct authenticated public account state from consensus-visible data. The aim is to make shielded state recoverable from the chain alone, subject to possession of the relevant viewing or spending material.

Selective disclosure is a first-class property of the model. A participant may disclose to a particular auditor, regulator, or counterparty the specific view material required for that relationship without disclosing the full transaction history to the network at large. This is not a claim of absolute invisibility. Endpoint compromise, physical coercion, and privileged observation remain outside the protocol's control. The narrower claim is that routine settlement need not leak amounts, counterparties, and strategy to unrelated observers by default.

Both transparent and shielded settlement remain first-class paths of a single chain. Block rewards may be handled through shielded flows as an ordinary operating mode rather than as an exceptional privacy overlay.

4. Timestamp Server and Chain of Work

Transactions are timestamped by inclusion in blocks. Blocks form a chain in which rewriting history requires redoing all subsequent work. The authoritative chain is the one with the greatest accumulated valid work, not the one with the greatest block count [2].

A BTX block header commits to the previous block hash, the transaction Merkle root, timestamp, target, nonce, and the fields binding the matrix-multiplication work transcript to the candidate block. Target spacing is 90 seconds. Difficulty adjusts per block using ASERT from genesis rather than coarse retarget epochs.

The operational consequence is standard longest-chain settlement. A transaction's place in history depends on a public sequence of work, not on an institution's internal record. The arbiter is the chain itself.

5. Proof-of-Useful-Work

BTX implements proof-of-work as matrix multiplication over the finite field $\mathbb{F}_{2^{31}-1}$ [3]. The work function is chosen to satisfy two requirements simultaneously:

1. a valid proof must remain cheap to reject and tractable to verify by the network; and
2. producing a valid proof must require present computation tied to current chain state.

The second property is what BTX refers to as *computational liveness*: a valid proof of work is simultaneously evidence that the prover is actively computing now, against a current challenge, rather than replaying a credential or expending value from a pre-funded balance.

Let $p = 2^{31} - 1$. Base matrices $A, B \in \mathbb{F}_p^{n \times n}$ are derived deterministically from chain state. Rank-r perturbation matrices $E, F \in \mathbb{F}_p^{n \times n}$, derived from the candidate header and nonce, prevent selection of favourable instances and precomputation. The miner computes:

$$C' = (A + E)(B + F).$$

A candidate block is valid only if

$$\mathrm{H}(A+E \,\|\, B+F \,\|\, C') < \mathit{target},$$

where H is the block hash function and $\|$ denotes serialised concatenation. Because E and F depend on the candidate header and nonce, the full transcript cannot be prepared independently of present chain state.

Full product verification uses Freivalds' algorithm. The veri-

fier chooses a random vector $\mathbf{r}$ and checks

$$(A+E)((B+F)\mathbf{r}) = C'\mathbf{r}.$$

This reduces full verification cost from cubic to quadratic time, with soundness error approximately $1/p$ per check; repeated independent checks drive the error to negligible levels.

The validation pipeline is split deliberately:

- header-level rejection is cheap;
- full transcript verification is more expensive but bounded;
- expensive verification is rate-limited as part of the node-security model.

Current network parameters are:

Parameter	Mainnet	Testnet	Regtest
Matrix dimension n	512	256	64
Noise rank r	8	4	4
Transcript block b	16	8	8
Validation window	1000	500	10

The work function has an additional economic consequence. Hardware securing BTX remains useful outside mining in AI and numerical-compute markets. The protocol does not claim that all mining hardware is interchangeable with all productive workloads; the narrower point is that the dominant computation class overlaps materially with productive compute markets, so mining capital need not be economically stranded when mining conditions change.

This same work function is reusable above the chain. A service can require a participant to present a fresh work proof tied to current network conditions. Unlike a pure fee gate, such a proof cannot be prepaid in bulk and amortised across requests without performing the actual computation. Difficulty can therefore be calibrated to the resource being requested.

6. The Difficulty Commons: Network Utility Without Token Ownership

BTX publishes, through its live chain state, a continuously updated measure of the work required to produce a valid proof under current network conditions. The chain exposes current target difficulty, recent timing, and derived work metrics to any participant operating a node.

This public benchmark is what BTX refers to as the *difficulty commons*: a tamper-evident, permissionless measure of the current cost of computation as expressed through the network's work market. It is not a price oracle, and it is not a complete market-clearing cost model for all compute markets. It is a public benchmark derived from open competition in proof production.

That benchmark is useful even to participants who never hold or transfer BTX:

- an API service can calibrate work-ticket difficulty against the live chain rather than a static threshold;
- a relay operator can tie anti-spam requirements to current network work conditions;
- two autonomous parties can use the same public benchmark when evaluating one another's liveness tickets;
- researchers and auditors can read the full historical record of the network's work difficulty directly from the chain.

The protocol therefore produces utility for non-holders as well as holders. Any participant can read the benchmark without account creation, operator permission, or token ownership.

7. Network, Relay, and Admission Control

At the base layer, BTX follows standard longest-chain network behaviour:

1. new transactions are broadcast to the network;
2. nodes collect transactions into candidate blocks;
3. miners search for valid work on those candidates;
4. a node finding valid work broadcasts the new block;
5. peers validate the block's transactions, spend conditions, shielded proofs, and work transcript;
6. accepted blocks are extended by later blocks.

Nodes always treat the chain with the greatest accumulated valid work as authoritative. Competing branches resolve when subsequent blocks extend one branch and not the other.

BTX uses Dandelion++-style stem-and-fluff transaction propagation [4]. This is a relay-layer privacy measure, not a consensus rule. In the current public-network deployment profile, Dandelion++ activates at block 250,000 after the network has established more stable peer relationships. The purpose is modest and specific: to reduce the ease with which passive observers can correlate a transaction with its originating node. It does not claim perfect origin anonymity.

Above the base layer, BTX work proofs can be reused as admission tickets. A bridge, relay, or service can require a proof of current work before accepting a request. This differs from fee-gated or balance-gated admission in one key respect: a valid ticket proves not only that the requester holds an identity key, but also that the requester has expended fresh computation now, against current network conditions. That raises the marginal cost of nuisance traffic in direct proportion to request volume.

This admission model is an application-layer use of the chain's work primitive. It is not required for ordinary transaction validation, but it is a designed interface enabled by the proof-of-useful-work construction.

8. Incentive and Distribution

BTX has a fixed maximum supply of 21 million units, an initial subsidy of 20 BTX, and halvings every 525,000 blocks. Transaction fees supplement the subsidy. These monetary rules are consensus rules enforced by full nodes; changing them requires a public fork adopted by participants [2].

All ordinary issuance occurs through valid work. Units are not created by administrative allocation at the protocol layer. A participant with a connected node and valid work competes on the same issuance surface as every other participant.

The initial 50,000 blocks are produced at compressed cadence to establish chain depth and seed the network into active use.

Units produced during this phase are held in a genesis multisig designated for network formation, including bridge seeding, relay infrastructure, and early market depth. The chain records both production and subsequent disbursement activity. The intended distinction from a conventional protocol-level pre-allocation is that these units are produced by open work and then managed through explicit on-chain holdings rather than being created outside the issuance process. Because this bootstrap mechanism is economically material, its operating policy should be documented separately and precisely for operators evaluating governance risk.

BTX is intended to function as settlement currency. Its supply is fixed, its issuance is work-based, and its work surface is tied to productive computation rather than to a purely synthetic hash race. That does not make BTX price-stable in the managed-currency sense. The narrower claim is that the unit is scarce by rule, issuable only by work, and economically anchored to computation markets.

9. Layered Settlement and Bridges

BTX is a settlement base layer, not a monolithic application environment. The base layer timestamps and orders commitments, enforces spend rules, maintains transparent and shielded settlement state, and exposes interfaces on which higher-layer systems can rely. Matching engines, local ledgers, bank liabilities, exchange internals, and high-frequency logic remain above the chain.

Layer-2 bridges are therefore expected rather than exceptional. A bridge operator can maintain a specialised local environment with its own throughput, privacy, risk controls, and user surface, then settle net obligations against BTX when balances cross trust or accounting boundaries. Different operators need not share one runtime, one codebase, or one governance surface in order to interoperate at final settlement.

Current consensus-visible bridge commitments include `BridgeBatchStatement` (version 5) and `BridgeBatchCommitment` (version 3), carrying authenticated roots such as:

- `action_root`
- `data_availability_root`
- `recovery_or_exit_root`
- `policy_commitment`
- `extension_digest`

These fields are committed from genesis. Their presence is a current protocol fact. Not all higher-level semantics against those fields are active today. In particular, future soft forks may further define the semantics of recovery and individual exit claims against `recovery_or_exit_root` without changing the fact that the authenticated commitment surface already exists.

Transaction data is bounded from genesis by fixed envelope limits. The design objective is to keep bridge settlement data explicit and authenticated without converting the base layer into a general-purpose execution environment.

Mass exit capacity

Layered systems fail most severely when entry is easy but exit becomes slow, discretionary, or operator-gated under stress. BTX is designed to avoid turning exit into a purely operator-dependent event. The current architecture supports three distinct egress modes:

- cooperative batch settlement;
- refund issuance without requiring the bridge to remain solvent or continuously available;
- timeout-triggered refund paths that allow users to claim against the base chain after a protocol-defined interval.

The current v2 egress proof is compact and fast to verify. Exit capacity is therefore constrained primarily by blockspace and batching policy rather than by an inherently long proof-verification bottleneck. The protocol does not impose an optimistic-rollup-style multi-day fraud window by design.

This is a core layered-settlement claim of BTX: operator efficiency and credible user exit are not treated as mutually exclusive design goals.

10. Development Model and Governance

BTX adopts a narrow-core governance philosophy. Changes to the reserve layer are intentionally difficult, explicit, and reviewable. Higher-layer systems are expected to evolve more quickly. The protocol design rejects the idea that application demand should routinely rewrite consensus.

This governance posture follows from the settlement role of the base layer. Institutions, operators, and autonomous systems can build above a chain only if the chain's fundamental monetary and settlement surface is predictable. A system whose core rules are continuously contestable under ordinary political or market pressure is a poor settlement base, regardless of how flexible it appears in the short term.

The development model is correspondingly test-driven and specification-oriented. A small number of output types, privacy mechanisms, work rules, and bridge interfaces is easier to specify, test, audit, and refine than a platform whose behaviour expands through an unbounded runtime surface.

BTX also adopts a constrained understanding of technical claims. Confirmation remains probabilistic. Privacy is tactical and bounded, not metaphysical. Decentralisation is a distribution of verification and control, not a slogan. These are engineering properties and should be evaluated as such.

Neutral incentive structure

The issuance rules are uniform. No participant receives units outside the work process by protocol privilege. No operator can redirect subsidy by administrative decision. Any post-hoc change to supply or issuance would require a visible public fork. That neutrality matters for human and non-human participants alike: no one can gain by persuading an administrator to alter the issuance surface in their favour.

AI Alignment at the economic layer

BTX does not solve inner alignment. It does not determine an autonomous system's goals. The narrower and more defen-

sible claim is that BTX provides conditions for *AI alignment at the economic layer*: a participant's financial scope can be bounded before action, its on-chain financial behaviour can be audited afterward, and the rules governing its participation are the same rules governing everyone else. For autonomous systems expected to operate under externally reviewable financial constraints, that is a meaningful infrastructure property.

11. Security Model and Confirmations

BTX inherits Bitcoin's longest-chain security model [2]. An attacker controlling less work than the honest network cannot cause honest nodes to accept invalid transactions or create value from nothing. The standard Nakamoto random-walk assumptions apply: honest and attacker block-production processes are modelled as Poisson processes with rates proportional to their share of total work.

Let p be the probability that the honest network finds the next block, q the probability that the attacker finds it, and q_z the probability that the attacker catches up from z blocks behind:

$$q_z = \begin{cases} 1 & \text{if } p \le q, \\ (q/p)^z & \text{if } p > q. \end{cases}$$

If the recipient waits for z confirmations and the attacker's expected progress is $\lambda = z(q/p)$, the remaining probability of a successful attack is:

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - \left(\frac{q}{p}\right)^{z-k}\right).$$

At 90 seconds per block, confirmation waiting times are shorter than in Bitcoin while preserving the same logic of risk reduction through accumulated work.

q	z	Wait (90 s)	Attack prob.
0.10	6	9 min	$\approx 2.4 \times 10^{-4}$
0.20	11	16.5 min	$< 10^{-3}$
0.30	24	36 min	$< 10^{-3}$
0.40	52	78 min	$< 10^{-3}$

The 50,000-block bootstrap phase increases early history depth and therefore the cost of wholesale reordering. Ordinary settlement security thereafter depends on standard confirmation accumulation. Participants choose depth according to value at risk: short machine-to-machine flows may tolerate shallow depth; large institutional transfers may require substantially more.

BTX isolates many higher-layer failures but does not deny them. A bridge can become insolvent. A venue can misprice risk. A local operator can fail. The chain's role is narrower: it provides a neutral settlement layer beneath those failures so they remain local rather than corrupting final settlement itself.

12. Conclusion

BTX is a constrained public settlement layer for institutions, operators, and autonomous agents that require final settlement under fixed rules rather than administrative discretion. It retains longest-chain settlement while combining four principal design choices: matrix multiplication replaces hash-only work; post-quantum spend paths replace legacy signature assumptions; shielded settlement and selective disclosure replace universal transparency as the default settlement model; and bridges and local ledgers replace the assumption that all application logic must share one global runtime.

The base layer is narrow by design. It orders settlement instructions, enforces spend conditions, maintains transparent and shielded settlement state, and publishes a public work benchmark. Higher-layer systems remain free to specialise above it without surrendering to one shared execution environment.

The resulting network provides a settlement surface with three notable operational properties. First, the work function is economically meaningful outside mining, reducing the degree to which security expenditure is isolated from productive compute markets. Second, valid work is tied to present chain state, making proof of work usable as evidence of current computation. Third, layered systems can settle against a neutral base while preserving harder exit rights than operator-dependent financial stacks typically offer.

The barrier to participation remains minimal: a node, keys, and a use for final settlement. The rest is built above.

References

[1] BTX Developers, "BTX: The Protocol of Record — Context Document," March 2026.

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[3] I. Komargodski and O. Weinstein, "Proofs of Useful Work from Arbitrary Matrix Multiplication," *arXiv preprint* arXiv:2504.09971, 2025.

[4] G. Fanti, S. B. Venkatakrishnan, S. Bakshi, B. Denby, S. Bhargava, A. Miller, and P. Viswanath, "Dandelion++: Lightweight Cryptocurrency Networking with Formal Anonymity Guarantees," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 2, no. 2, article 29, 2018.

[5] National Institute of Standards and Technology, "FIPS 204: Module-Lattice-Based Digital Signature Standard," 2024.

[6] National Institute of Standards and Technology, "FIPS 205: Stateless Hash-Based Digital Signature Standard," 2024.

[7] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, 2nd ed. Wiley, 1957.