Qbit: Post-Quantum Peer-to-Peer Digital Value

www.qbit.org

June 2, 2026

Abstract. Peer-to-peer digital value depends on cryptography that must remain trustworthy over long time horizons. As post-quantum migration moves from theory to engineering reality, open monetary networks need more than a signature upgrade: they need an architecture that accounts for larger witnesses, different exposure surfaces, and the operational cost of carrying post-quantum data through the system. Qbit is built around that requirement. It combines bounded SLH-DSA, P2MR outputs, direct pricing of witness bytes, fast aggregate block cadence, dual-lane mining, and explicit witness-retention modes to preserve self-custodied digital value under post-quantum constraints.

1. Introduction

Open monetary networks rely on digital signatures, public transaction history, and proof-of-work consensus to transfer value without a trusted financial intermediary [1]. Qbit follows that model with a UTXO ledger, work-based chain selection, and direct ownership, but it is a new network rather than a hard fork or continuation of Bitcoin: it has its own genesis block, consensus rules, monetary policy, and asset.

The cryptographic setting for those networks is changing, and timing matters. Shor's algorithm threatens the discrete-log assumptions behind conventional elliptic-curve signatures [2], and public keys or signatures can remain visible on-chain for the lifetime of an asset. Google has framed cryptocurrency quantum exposure as a responsible-disclosure and migration problem with a 2029 transition horizon for vulnerable systems [3], while NIST already recommends beginning post-quantum migration because the transition itself will take time [4].

Rapid advances in quantum computing add uncertainty to an already precarious situation. Across existing cryptocurrency communities, migration policy for dormant coins vulnerable to quantum attack remains an unsettled and ongoing debate. As legacy networks continue to confront technical debt, governance disputes, and the risks of complex upgrade paths, the need for a post-quantum digital asset secured from genesis has never been more clear.

Qbit begins from the premise that post-quantum security must be treated as a base-layer design constraint, not as a later cryptographic retrofit. The question is therefore not only which post-quantum signature primitive to use, but how the chain itself should be built once large hash-based signatures, shorter exposure windows, explicit validation costs, and wallet key-management limits become normal operating assumptions. Qbit answers at the architecture level: output construction avoids long-lived spend-key exposure, witness bytes and validation cost are priced directly, and propagation, storage, and chain formation stay within explicit bounds.

By making post-quantum security part of the initial ledger state, Qbit establishes a clean security boundary from the first block. Every output is created under the same cryptographic assumptions, and ownership never depends on a future migration deadline, custodial coordination, or governance judgment about inactive funds. The result is a protocol whose security model is uniform across its history: no inherited class of vulnerable coins, no retroactive remediation process, and no need to resolve cryptographic risk through social policy.

2. Design Overview

Qbit is organized around a small set of linked design decisions.

Component	Role in the system
bounded `SLH-DSA-SHA2-128s-bounded30`	active post-quantum spend authorization profile
P2MR + `OP_CHECKSIGPQC`	script and address model for post-quantum spends
`WITNESS_SCALE_FACTOR = 1`	direct pricing of witness bytes
2,000,000-byte / 2,000,000-weight blocks	bounded propagation and validation envelope
one-minute aggregate cadence + ASERT	continuous retargeting for a faster chain
permissionless SHA-256 + AuxPoW	one chain with two mining lanes and most-work selection
compound-floor emissions	deterministic 210,000,000 QBT cap with stepwise subsidy decay
witness pruning + archive mode	practical long-term node operation under large witnesses

These choices are not independently selected. Large post-quantum signatures change transaction size. Transaction size changes fee accounting and relay pressure. Relay pressure changes the acceptable block envelope. A tighter block envelope and faster cadence affect mining and retargeting, and historical witness growth then changes the storage model for ordinary nodes. Qbit is defined by that chain of dependencies.

Qbit's implementation is derived from a fork of Bitcoin Core v30.2. That software lineage should not be confused with chain lineage: Qbit does not inherit Bitcoin's history, UTXO set, or balances. The Bitcoin Core base instead contributes a mature node architecture and more than a decade of reviewed security fixes before Qbit-specific consensus and wallet changes are applied.

3. Threat Model

Qbit's post-quantum claim is intentionally narrow. The protocol is designed to protect spend authorization against the failure of classical public-key assumptions, especially the discrete-log assumptions behind ECDSA and Schnorr [2]. Proof-of-work and other hash-based commitments still depend on the post-quantum strength of the hash function and could be affected by Grover-type speedups, while network, storage, and custody risks remain separate.

The threat model centers on three ideas:

- the primary target is classical public-key spend authorization
- the critical exposure surfaces are long-lived revealed keys and short-lived transaction data during propagation
- the system assumes its hash primitives retain useful post-quantum security margins under known attacks, while mining, networking, storage, custody, and historical data availability remain separate concerns

Two exposure windows matter. Long-exposure attacks target keys or outputs that remain revealed over long periods; short-exposure attacks target live transactions while they propagate. Qbit addresses both by using a post-quantum spend model from genesis and by preferring outputs that keep keys and unused branches hidden until spend.

Qbit also keeps the base layer close to Bitcoin's model: a simpler UTXO network rather than an account system, with a limited execution surface and fewer state transitions that must remain secure under post-quantum assumptions.

4. Transaction Model

4.1 Signature profile

NIST standardized SLH-DSA in FIPS 205 as the stateless hash-based digital signature standard [5]. Qbit's active profile, `SLH-DSA-SHA2-128s-bounded30`, is a bounded instantiation chosen for Qbit from the same hash-based design family. Qbit adopts this family because it avoids the discrete-log assumption threatened by Shor's algorithm, keeps public keys compact, and provides a conservative path for base-layer spend authorization.

The tradeoff is explicit. Public keys remain compact, but the active signature size is 3,680 bytes, materially larger than a classical Bitcoin spend. Qbit accepts a `2^30` per-key usage bound in order to reduce validation and relay burden. WOTS+C and FORS+C operate here as compression steps inside the bounded construction rather than as separate signature families. Qbit does not hide the signature cost behind inherited accounting rules. Appendix A lists all active sizes and limits.

Qbit launches with a bounded SLH-DSA-family spend profile because it keeps the base-layer signing model stateless and close to the NIST-standardized SPHINCS+/SLH-DSA line [5]. SHRINCS and SHRIMPS are promising Bitcoin-oriented hash-based designs with materially smaller signatures, but their compact modes rely on signer-state assumptions, device-initialization bounds, and recovery/fallback behavior that would need to be specified across wallets and custody systems [6, 7]. Qbit's witness and leaf reservations leave room to adopt those or similar schemes later through explicit soft-fork upgrade paths.

4.2 P2MR outputs

Spendable outputs on Qbit use Pay-to-Merkle-Root (P2MR), a version-2 witness output type that commits to a Merkle root rather than exposing a spend script or public key directly in the output. When the output is later spent, the witness reveals the chosen leaf script, the Merkle branch proving that leaf belongs to the committed root, and the post-quantum witness data authorizing that path.

This keeps ordinary addresses compact, avoids exposing reusable public keys at output creation time, and keeps unused script branches hidden. It is closely aligned with the existing Merkle-committed post-quantum UTXO proposal, BIP-360 [8].

Qbit's P2MR script surface is centered on a small set of post-quantum and template-oriented opcodes:

Opcode	Role
`OP_CHECKSIGPQC`	verifies a post-quantum signature over the spending transaction
`OP_CHECKTEMPLATEVERIFY`	constrains a P2MR spend to a committed transaction template
`OP_CHECKDATASIGPQC`	verifies a post-quantum signature over an explicit 32-byte message hash
`OP_CHECKDATASIGADDPQC`	accumulates post-quantum data-signature results for threshold-style scripts

The Qbit-specific aspect is that these operations are scoped to the P2MR spend path, rather than reopening the inherited legacy script surface.

4.3 Example transaction flow

Consider a wallet creating an output for a future recipient. It commits one or more authorized spend conditions as Merkle leaves and publishes only the resulting 32-byte root. When that output is spent, the witness reveals the chosen leaf script, its Merkle branch, and the post-quantum witness for that path; a validating node checks the commitment and then executes `OP_CHECKSIGPQC`.

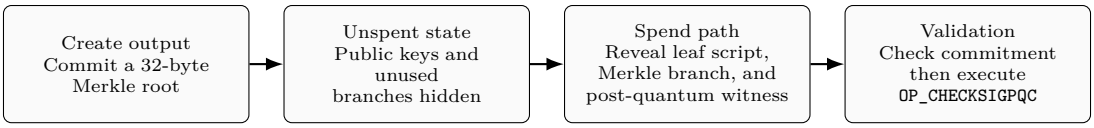

Figure 1: P2MR spend lifecycle. The output commits only to a Merkle root; at spend time the chosen leaf, Merkle branch, and post-quantum witness are revealed for validation under `OP_CHECKSIGPQC`.

5. Resource Accounting

Large post-quantum signatures force the protocol to account for resources more directly than inherited witness discounting would allow. Qbit therefore sets `WITNESS_SCALE_FACTOR = 1`, so weight follows serialized size directly: $W = B$, where block weight and serialized bytes coincide.

That decision is the accounting center of the protocol. In a system where a single signature is 3,680 bytes, the dominant costs are the bytes transmitted, stored, and validated. Qbit therefore aligns three categories more closely:

- propagation cost on the wire
- long-term storage growth
- fee pressure inside mempool and block construction

The same logic extends to validation. Qbit bounds byte growth with a 2,000,000-byte / 2,000,000-weight block envelope, and it bounds script verification through a fixed 50-unit overhead P2MR validation budget: $V_{input} = B_{witness} + 50$, with each passing post-quantum signature consuming $V_{sig} = 3730$. Witness size and validation cost are therefore tied together rather than treated as unrelated concerns.

This makes the current envelope easy to read:

$$N_{block} \approx \frac{B_{max}}{S_{tx}}, \qquad \text{TPS} \approx \frac{N_{block}}{T_{total}} \approx \frac{B_{max}}{S_{tx}\, T_{total}}$$

where N_{block} is the transaction count in the block, $B_{max} = 2{,}000{,}000$ bytes, S_{tx} is the average serialized transaction size, and T_{total} is the aggregate block interval. With 3,680-byte signatures, this implies hundreds of signature-bearing transactions in a full block, not thousands, and a corresponding throughput ceiling even at a one-minute cadence. That is a deliberate result of present post-quantum costs. If future post-quantum spend generations reduce witness size through explicit soft-fork upgrade paths, the same accounting model scales with them automatically.

6. Consensus and Chain Formation

6.1 One chain, two mining lanes

Qbit uses one chain selected by most accumulated work. It does not treat permissionless blocks and AuxPoW blocks as separate ledgers, and it does not use a height-based alternation rule.

The protocol admits two mining paths:

- permissionless native SHA-256 mining
- AuxPoW-backed blocks produced through merge-mining infrastructure

This arrangement preserves two useful properties. Permissionless mining keeps participation open and does not require immediate pool coordination. AuxPoW allows the network to draw security from Bitcoin's global SHA-256 hashrate through merge mining [9] without asking miners to abandon established revenue streams. Bitcoin's global SHA-256 hashrate was running at roughly 1,012 EH/s on a seven-day simple moving average as of June 1, 2026 [10]. As the merged-mining share rises, more of that existing Bitcoin-linked hashrate contributes to honest AuxPoW chainwork, making it harder to replace and raising the cost of a majority-style attack against the combined chain.

The current lane schedule targets a 60-second aggregate cadence through a 75-second target for the

permissionless lane and a 300-second target for the AuxPoW lane:

$$\lambda_{total} = \lambda_{perm} + \lambda_{aux} = \frac{1}{75} + \frac{1}{300}, \qquad T_{total} = \lambda_{total}^{-1} = 60 \text{ s}$$

Because expected lane share is proportional to $1/t_i$, these timings imply a 4:1 target mix in favor of permissionless mining while still allowing AuxPoW to add meaningful security density.

Fork choice still resolves on accumulated work rather than on raw block count. At a high level,

$$W_{block} \approx \frac{2^{256}}{T+1}, \qquad W_{chain} = \sum_i W_{block,i}$$

where T is the encoded proof-of-work target for a block, and the selected chain is the valid chain with maximal accumulated chainwork. In that sense, the "heaviest chain" is the one with the greatest accumulated work. Additional honest hashrate on the AuxPoW lane therefore matters not because it creates a separate fork-choice rule, but because it adds more valid work to the same cumulative-work race.

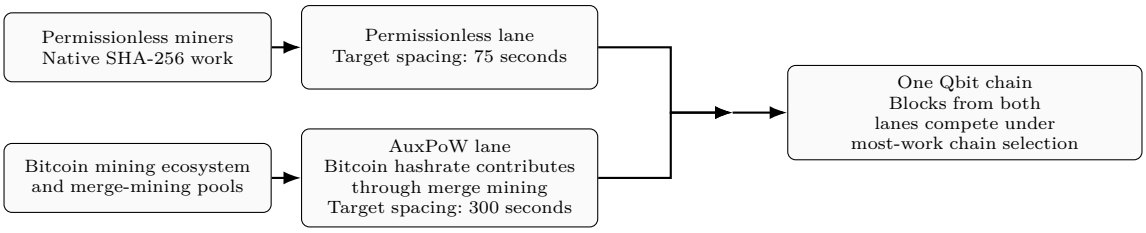

Figure 2: Qbit mining structure. Permissionless mining and AuxPoW blocks feed a single most-work chain. Merge mining lets Qbit draw on Bitcoin hashrate while preserving an open native lane.

6.2 Absolutely Scheduled Exponentially Rising Targets (ASERT)

Qbit uses ASERT with a two-hour halflife to retarget difficulty continuously rather than through long epochs [11]. In the current dual-lane system, ASERT governs both lanes independently: permissionless blocks are retargeted against the permissionless schedule, AuxPoW blocks are retargeted against the AuxPoW schedule, and the resulting valid blocks still compete in a single cumulative-work fork-choice rule.

At a high level, the next target is derived from:

$$\text{target}_{next} = \text{target}_{anchor} \times 2^{(t_{actual} - t_{ideal})/\tau}$$

where τ is the halflife and the time offset is measured against the ideal schedule for the relevant lane.

6.3 Initial Difficulty

Qbit also treats initial difficulty as a lane-specific launch calibration. It first defines a single reference price for QBT from fully diluted value and terminal supply:

$$P_{QBT} = \frac{\text{FDV}}{S_{QBT}}$$

Here `FDV` is the fully diluted value target. Using the intended value of `$100,000,000` and the fixed terminal supply of `210,000,000 QBT`, the reference launch price is approximately `$0.47619` per QBT. At a `210 QBT` permissionless block reward, that implies a `$100.00` permissionless block value. For the permissionless lane, the launch model converts that price into expected SHA-256 work:

$$V_p = R_p \times P_{QBT}, \qquad h_{hash} = \frac{\text{hashprice}}{10^{15} \times 86400}, \qquad D_p = \frac{V_p}{h_{hash}\, 2^{32}}$$

Here R_p is the permissionless block reward, and h_{hash} converts the quoted SHA-256 hashprice from USD per PH per day into dollars per hash. Using a placeholder hashprice of `$32.56 USD/PH/day`, the permissionless model yields an initial launch difficulty of approximately `61.8 billion`.

The choice of FDV and the derived initial difficulty addresses the most common fair-launch criticism of new proof-of-work assets: that the earliest supply was effectively free. By anchoring genesis difficulty to an externally auditable cost of production, Qbit creates a permanent on-chain record that the initial coins were fairly mined. This ensures that Qbit's launch supply begins from a meaningful production-cost floor and then converges rapidly to market equilibrium through the difficulty adjustment algorithm.

For the AuxPoW lane, the launch model starts instead from an assumed share of global Bitcoin hashrate and the AuxPoW lane target spacing:

$$H_{aux} = H_{BTC,global} \times share_{aux}, \qquad D_{aux} = \frac{H_{aux}\, T_{aux}}{2^{32}}$$

Here $H_{BTC,global}$ is the assumed global Bitcoin hashrate, $share_{aux}$ is the assumed merged-mining share, and T_{aux} is the AuxPoW lane target spacing, currently `300` seconds. Using an assumed merged-mining share of `1%` of a rounded `1,000 EH/s` global Bitcoin hashrate, the AuxPoW model implies an assumed `10 EH/s` AuxPoW contribution and an initial launch difficulty of approximately `698.5 billion`. The key point is that the two lanes are calibrated from different assumptions: the permissionless lane from a reference QBT price and production-cost floor, and the AuxPoW lane from an assumed share of Bitcoin hashpower.

6.4 Confirmation and settlement policy

Operator settlement policy should target Bitcoin-equivalent security rather than a static Qbit depth. The security value of a confirmation depends on lane mix, merge-mining participation, native Qbit hashrate relative to Bitcoin, and stale-block rate.

At a high level, each Qbit confirmation contributes some fraction of one Bitcoin confirmation:

$$s_{\text{Qbit}} \approx \phi_{aux}\, m + (1 - \phi_{aux}) \frac{H_{\text{Qbit}}}{H_{BTC}}$$

where ϕ_{aux} is the expected AuxPoW share implied by the cadence schedule, m is the assumed fraction of Bitcoin hashrate merge-mining Qbit, and H_{Qbit}/H_{BTC} is Qbit's native hashrate relative to Bitcoin. Required Qbit confirmations rise when merge-mining participation or native hashrate are weak, and they rise further when stale-block rates increase. Settlement policy is therefore dynamic, not a simple "ten times faster blocks means ten times fewer minutes" conversion.

6.5 Fork choice and issuance

Monetary issuance is deterministic rather than Bitcoin-style halving-based. Qbit starts with a 210 QBT subsidy from genesis, includes transaction fees in block reward economics, and steps the subsidy down every 43,200 blocks on mainnet by multiplying the prior subsidy by 598/625 and rounding down. The consensus money-range cap is 210,000,000 QBT; actual subsidy emission is slightly below that cap because each step is floored by integer arithmetic. Coinbase outputs mature after 1,000 blocks. The schedule is fixed in the consensus rules.

Qbit also pairs that schedule with an explicitly open and fair distribution path. No tokens are assigned or distributed outside the mining process, and no later migration event is required to convert legacy balances into post-quantum-safe balances. Under the current mainnet schedule and one-minute aggregate target cadence summarized in Appendix A, more than 70% of terminal supply

would be distributed by January 1, 2029. That means most ownership would already be in circulation before the 2029 post-quantum migration timeline Google has articulated more broadly, while avoiding the neutrality and legacy-output dilemmas that older chains may face [3]. This earlier distribution reduces the risk of a prolonged low-float, high-FDV dynamic that can distort price discovery.

7. Network and Node Operation

Network behavior and disk-space management are core protocol concerns, not implementation details. Large witnesses affect not only transactions but also propagation, historical validation, and the ordinary operating model of nodes.

The protocol defines both archive-retaining and witness-pruned operating modes. Current software defaults to retaining historical witnesses, while witness-pruned nodes remain an explicit operating mode for nodes that validate the active chain and maintain current chainstate without keeping every historical witness forever. Archive-retaining nodes preserve the historical witness surface needed for deep replay and deep-reorg recovery, while witness-pruned nodes rely on archive help for those cases.

Propagation matters directly when blocks arrive every minute and signatures are measured in kilobytes. Qbit therefore assumes tuned relay defaults and compact block usage. Optional relay acceleration can improve that path further: Qbit's PHOTON implementation is a FIBRE-like overlay [12] that reduces propagation delay for miners and infrastructure operators, but it is not part of consensus and it is not required for independent validation.

8. Wallet and Custody Model

Post-quantum signatures alter wallet and custody software as much as they alter consensus. Hash-based signatures do not preserve the public-derivation model used by BIP-32-style wallets [13], so deterministic recovery remains available, but watch-only workflows require explicit public-key export and tracking instead of a single public derivation string.

The active bounded profile also introduces a per-key usage bound of `2^30` signatures. Therefore Qbit approaches per-key usage as a key component of the wallet security model: consensus verifies signatures, but wallet software must track usage, monitor keys as they approach exhaustion, and rotate to fresh keys before any limits are reached.

For exchanges, custodians, and payment processors, this also changes deposit policy. Confirmation thresholds should be set against the chain's Bitcoin-equivalent settlement target rather than against a static raw Qbit block count, as discussed in the confirmation and settlement policy section.

9. Security Tradeoffs and Upgrade Surfaces

Qbit's security model depends on explicit consensus narrowing. Restricted-output networks accept only P2MR, `OP_RETURN`, pay-to-anchor, and, once active, reserved future witness outputs. That removes the broad inherited legacy spend surface and keeps ordinary spend authorization inside Qbit's post-quantum transaction model. P2MR v1 defines live spend semantics; future changes are intended to use activated soft-fork surfaces such as reserved outer witness versions and reserved P2MR leaf codes.

10. Conclusion

Qbit starts from a clear observation: a monetary network that expects value to remain self-custodied over decades cannot take today's public-key cryptography for granted. Existing proof-of-work chains are forced to approach post-quantum security as a migration problem; Qbit approaches it as a founding assumption, avoids the migration and neutrality dilemmas that legacy chains may face, and is designed for cryptographic longevity as the quantum threat becomes harder to ignore.

Appendix A. Selected Parameters

Parameter	Value
active PQC spend profile	`SLH-DSA-SHA2-128s-bounded30`
signature size	3,680 bytes
usage bound	`2^30` signatures per key
spendable output model	P2MR + `OP_CHECKSIGPQC`
block envelope	2,000,000 bytes / 2,000,000 weight
P2MR validation cost per passing signature	3,730
difficulty cadence	60-second aggregate; 75-second permissionless; 300-second AuxPoW
difficulty adjustment	ASERT, 2-hour halflife
initial block subsidy	210 QBT
mainnet subsidy schedule	43,200-block steps with 598/625 stepdown
MAX_MONEY consensus cap	210,000,000 QBT
coinbase maturity	1,000 blocks
historical witness model	archive-retaining by default, with explicit witness-pruned mode

Note on Launch-Calibration Assumptions

The FDV, hashprice, Bitcoin hashrate, and merge-mining participation inputs used for launch calibration are external assumptions used to derive initial difficulty targets. They are not forecasts or commitments about post-launch QBT valuation, SHA-256 hashprice, merge-mining participation, or Bitcoin hashrate.

References

1. Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008, accessed June 2, 2026. Bitcoin whitepaper PDF
2. Peter W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," arXiv:quant-ph/9508027, 1995, accessed June 2, 2026. arXiv preprint
3. Google Research, "Safeguarding cryptocurrency by disclosing quantum vulnerabilities responsibly," March 31, 2026, accessed June 2, 2026. Google Research blog post
4. NIST, "Post-quantum cryptography," accessed June 2, 2026. NIST PQC page
5. NIST, FIPS 205, "Stateless Hash-Based Digital Signature Standard," August 13, 2024, accessed June 2, 2026. FIPS 205 DOI
6. NIST, SP 800-208, "Recommendation for Stateful Hash-Based Signature Schemes," October 2020, accessed June 2, 2026. NIST SP 800-208
7. Blockstream, "Post-Quantum Signatures: SHRIMPS and SHRINCS," accessed June 2, 2026. Blockstream quantum overview
8. Bitcoin Improvement Proposal 360, "Pay-to-Merkle-Root (P2MR)," accessed June 2, 2026. BIP-360 site
9. Bitcoin Wiki, "Merged mining specification," accessed June 2, 2026. Merged mining specification
10. Hashrate Index, "Hashrate Index Roundup (June 1, 2026)," June 1, 2026, accessed June 2, 2026. Hashrate Index roundup
11. Bitcoin Cash Node, "2020-NOV-15 ASERT Difficulty Adjustment Algorithm (aserti3-2d)," version 0.6, August 12, 2020, accessed June 2, 2026. Bitcoin Cash ASERT upgrade specification
12. FIBRE, "Frequently Asked Questions," accessed June 2, 2026. FIBRE FAQ
13. Bitcoin Improvement Proposal 32, "Hierarchical Deterministic Wallets," accessed June 2, 2026. Bitcoin BIPs: BIP-32