Atom Quantum

A Post-Quantum Execution and Settlement Network

Vision Whitepaper — v0.5 (draft) · July 2026

Document status: technical vision grounded in a running implementation. The network operates as a multi-node testnet alpha (public explorer: https://scan.atqm.network/); mainnet parameters are not final. Throughout this paper, capabilities are labeled **Implemented (alpha)** — present and exercised in the running network — or **Vision** — a design objective, not a claim about running software. This document remains a working draft under active revision.

Revision note: v0.4 is the currently published revision (https://atqm.network/documents/atqm-whitepaper-v04.pdf). This v0.5 draft updates it in one respect: it adds the preliminary token allocation (Section 9.1). All other content is unchanged from the published revision.

Abstract

Public blockchains use public-key cryptography for far more than transferring value: it identifies accounts, authenticates validators, establishes peer sessions, and authorizes smart-contract actions. A cryptographically relevant quantum computer breaks the discrete-logarithm assumptions behind all of it — ECDSA, EdDSA, BLS aggregation, and elliptic-curve key exchange — at every one of those layers simultaneously. Because chain data is public and immutable, the exposure is retroactive: every public key ever revealed on-chain remains permanently forgeable once the underlying problem becomes tractable. Retrofitting a live network is therefore not a signature swap but a coordinated migration of accounts, consensus, networking, and wallets, executed against a deadline that regulators have already set.

Atom Quantum is a Layer-1 designed around this problem from genesis. Its running implementation combines an EVM-compatible execution engine with post-quantum transaction authentication (ML-DSA, FIPS 204; FN-DSA/Falcon), a HotStuff-family Byzantine-fault-tolerant finality protocol in which every validator vote is an individual post-quantum signature, and ML-KEM-secured (FIPS 203) node sessions. Finality is deterministic and block-bound: public clients accept only blocks accompanied by a valid quorum-certificate proof, and a finalized block cannot be reorganized. Programmable accounts — delegation, fee sponsorship, batching, and recovery policies — and on-chain post-quantum signature verification for smart contracts are native capabilities rather than add-on layers.

This paper presents the threat model with its mathematics, the lattice-based cryptographic foundation, the consensus design and its cost model, the account and execution architecture, the state of the running network, and the application thesis that motivates the design: deterministic post-quantum settlement for on-chain trading, cross-chain security, stablecoin rails, and tokenized real-world assets — the classes of value that must outlive the cryptography they are issued on.

Contents

1 Introduction: The End of Curve-Based Settlement **4**

1.1 One assumption, every layer . . . 4
1.2 Two clocks, failing differently . . . 4
1.3 The closing window . . . 4
1.4 Migration is not a retrofit . . . 5

2 Design Principles **6**

3 Cryptographic Foundation **7**

3.1 Why lattices resist quantum attack . . . 7
3.2 ML-DSA — the default signature *(Implemented, alpha)* . . . 7
3.3 FN-DSA / Falcon — the compact option *(Implemented, alpha; opt-in)* . . . 8
3.4 ML-KEM — transport confidentiality *(Implemented, alpha)* . . . 9
3.5 The symmetric layer . . . 9
3.6 NIST security categories . . . 9
3.7 Size as a protocol parameter . . . 9
3.8 Account identity . . . 10

4 Consensus and Deterministic Finality **10**

4.1 Protocol shape *(Implemented, alpha)* . . . 10
4.2 Quorums and their intersection . . . 11
4.3 Safety and liveness . . . 11
4.4 Progress by evidence, not by clocks . . . 12
4.5 The cost model, and why the committee is bounded . . . 12
4.6 Finality latency . . . 12

5 Execution and Accounts **12**

5.1 EVM-compatible, deliberately not Ethereum-compatible *(Implemented, alpha)* . . . 12
5.2 Programmable accounts *(Implemented, alpha)* . . . 13
5.3 Post-quantum verification for contracts *(Implemented, alpha)* . . . 13

6 Post-Quantum Networking **13**

7 Security Model **14**

7.1 Assumptions . . . 14
7.2 Guarantees . . . 14
7.3 Out of scope . . . 14
7.4 Positioning in the post-quantum landscape . . . 15

8 Application Vision **15**
8.1 On-chain trading . . . 16
8.2 Quantum Vaults: post-quantum protection for other chains . . . 16
8.3 Dollar rails . . . 17
8.4 Tokenized real-world assets . . . 17

9 Economic Framework **17**
9.1 Preliminary token allocation *(draft; subject to change)* . . . 18

10 Roadmap **19**

11 Conclusion **19**

References **20**

A Parameter Reference **21**

B Glossary **21**

1 Introduction: The End of Curve-Based Settlement

1.1 One assumption, every layer

The security of nearly every production blockchain reduces to the hardness of the elliptic-curve discrete-logarithm problem (ECDLP). Bitcoin and Ethereum authenticate transactions with ECDSA over secp256k1; Ethereum's consensus aggregates validator votes with BLS signatures over pairing-friendly curves; peer-to-peer transport is keyed by elliptic-curve Diffie–Hellman. In 1994 Peter Shor exhibited a polynomial-time quantum algorithm for discrete logarithms and factoring [1]. A cryptographically relevant quantum computer (CRQC) running Shor's algorithm therefore does not merely weaken these systems — it collapses transaction authenticity, validator authentication, and transport confidentiality at once, and pairing-based aggregation offers no graceful degradation path.

1.2 Two clocks, failing differently

The quantum threat to a ledger runs through two mechanisms that are often conflated but fail differently:

- **Record now, forge later** is an *authenticity* failure. Public keys and signatures recorded on-chain remain available to any adversary indefinitely. Once the relevant discrete logarithm becomes tractable, the adversary derives the signing key of any exposed account and forges valid transactions. On an account-based ledger this exposure is structural: an account is a long-lived, key-reusing identity whose public key is revealed at first use, and an attacker can work against an exposed, dormant account for as long as its funds remain valuable — the attack does not need to fit inside a settlement window.
- **Harvest now, decrypt later** is a *confidentiality* failure. Network traffic captured today can be decrypted later if its session establishment rested only on quantum-vulnerable key exchange. The U.S. Federal Reserve's analysis of distributed-ledger networks examines precisely this risk and its permanence: data recorded now remains vulnerable even after a network later migrates [4].

A third, subtler mode targets *trusted setups*: an adversary who recovers the secret randomness ("toxic waste") behind a pairing-based ceremony — as used by KZG commitments — mints a reusable forgery capability that thereafter runs on classical hardware. In the resource-estimation literature these modes are catalogued as **on-spend**, **at-rest**, and **on-setup** attacks, distinguished by how fast the attacking machine must run: forging an in-flight transaction inside its settlement window demands a *fast-clock* CRQC (superconducting, photonic), while an exposed dormant key can be worked on for days by a *slow-clock* machine (neutral-atom, ion-trap) two to three orders of magnitude slower per operation [23]. Atom Quantum's design closes all three modes: authenticity rests on lattice signatures, confidentiality on lattice key exchange, and the protocol contains no pairings and no trusted setup (Fig. 1).

1.3 The closing window

Let C_{build} denote the fixed, one-time cost of constructing a post-quantum network, and $C_{\text{attack}}(t)$ the falling cost of fielding a CRQC. The safety margin is

$$M(t) = C_{\text{attack}}(t) - C_{\text{build}}.$$

Published resource estimates place the 256-bit ECDLP at a few thousand logical qubits with a large but finite gate budget [2], and successive results on fault-tolerant architectures continue to revise attack constants

Attack mode	Exploited surface	CRQC required	How Atom Quantum closes it
On-spend	Transaction signature in flight, inside the settlement window (ms–minutes)	Fast-clock (superconducting, photonic)	No curve signature exists to forge
At-rest	Public key exposed on-chain; attacker has days or more	Slow-clock suffices (neutral-atom, ion-trap)	No curve public key is ever exposed
On-setup	Trusted-setup "toxic waste"; one CRQC run mints a reusable classical exploit	One-time access	No trusted setup, no pairings, no KZG

Figure 1: Taxonomy of quantum attacks on a ledger [23]: the surface each mode exploits, the class of machine it requires, and the structural reason each is absent from Atom Quantum.

downward — recent work reduced the estimated machine for RSA-2048 to under a million noisy qubits [3]. For the secp256k1 instance specifically, the most recent published estimates demonstrate Shor circuits at $\leq$ 1,200 logical qubits and $\leq$ 90 million Toffoli gates — or, trading space against gates, $\leq$ 1,450 logical qubits and $\leq$ 70 million Toffoli — executable on a superconducting architecture with 10^{-3} physical error rates and planar connectivity using **fewer than half a million physical qubits**, roughly an order of magnitude below prior estimates for a single key extraction [23]. These are demonstrations of a finite engineering problem, not a present-day capability, and Atom Quantum does not predict a "Q-day." Its design premise is narrower and more defensible: **a ledger intended to preserve value for decades should not require an accurate forecast of when its account authentication fails.** The rational posture is to construct while $M(t) > 0$ is comfortably large, because the alternative — migrating a live network after the margin narrows — must be executed under adversarial time pressure (Fig. 2).

1.4 Migration is not a retrofit

A live curve-based chain cannot transparently swap its signature scheme. Addresses are derived from curve public keys; finality is signed on the curve; transport is keyed by curve Diffie–Hellman; a decade of wallets, custody systems, and audit tooling assumes all of it. Replacing the primitive changes the address surface and the consensus messages simultaneously, and no migration can retroactively protect keys already exposed. The regulatory calendar sharpens the engineering argument: NIST finalized ML-KEM, ML-DSA, and SLH-DSA in August 2024 [5–7], and its draft transition guidance deprecates quantum-vulnerable public-key cryptography after 2030 and disallows it after 2035 [8] — a window inside the maturity of the long-dated instruments being tokenized on curve-based chains today. Parallel mandates (CNSA 2.0; EU critical-infrastructure requirements) point at the same decade.

Atom Quantum's thesis follows: **post-quantum security is a property a network has from genesis, or acquires through a migration whose difficulty grows with everything built on top of it.** This paper describes a network built the first way.

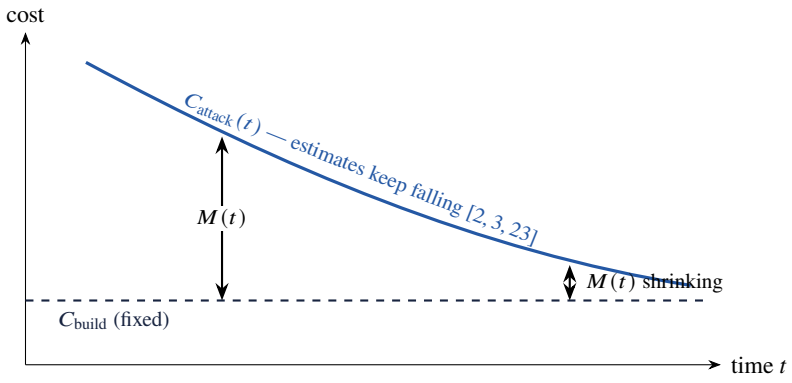

Figure 2: The closing window (schematic, not to scale). $C_{\text{attack}}(t)$ falls as algorithms and error correction improve; C_{build} is fixed. The safety margin $M(t)$ is the shrinking gap between the curves. Building while $M(t) \gg 0$ is a choice; migrating after it narrows is an emergency executed under adversarial time pressure. The curve's shape is qualitative — no "Q-day" is predicted; only the monotone direction of published estimates [2, 3, 23] is asserted.

2 Design Principles

Post-quantum security is a system property. A chain is not post-quantum-secure if its transactions use lattice signatures while its validator votes, node identity, or transport still depend on a curve. Atom Quantum applies post-quantum primitives to account authentication, consensus signatures, node sessions, and contract-level verification alike; no ECDSA, EdDSA, BLS, or pairing-based construction exists in any native security boundary.

NIST standards, not experiments. Every asymmetric primitive is a finalized NIST standard or a NIST-selected scheme in active standardization: ML-DSA (FIPS 204) and FN-DSA/Falcon (FIPS 206, in development [9]) for signatures; ML-KEM (FIPS 203) for key encapsulation; AES-256 and Keccak for the symmetric layer. The NIST process subjected these schemes to roughly eight years of open, global cryptanalysis; Atom Quantum builds on its output rather than on bespoke constructions.

Execution compatibility, not protocol compatibility. Atom Quantum retains EVM state-transition semantics — Solidity, 20-byte addresses, dynamic fees [20], typed transactions, access lists — so contracts and developer tooling carry over. It does not claim compatibility with Ethereum's signers, networking, fork choice, BLS consensus, or blob/KZG data availability [22]; those are precisely the surfaces the design exists to replace.

Deterministic finality. Settlement is a certificate, not a probability. Public clients accept only finalized blocks carrying valid proof, and a finalized block cannot be reorganized.

Costs are first-order. Post-quantum keys and signatures are one to two orders of magnitude larger than their curve predecessors. This paper treats the byte, bandwidth, and verification costs as explicit design parameters (Sections 3.7, 4.5) rather than footnotes.

Cryptographic agility. Every key and signature carries a scheme identifier, and signing is domain-separated by protocol role, so an algorithm can be replaced under a controlled upgrade without ambiguous key interpretation — a necessary hedge against future cryptanalysis of any single family.

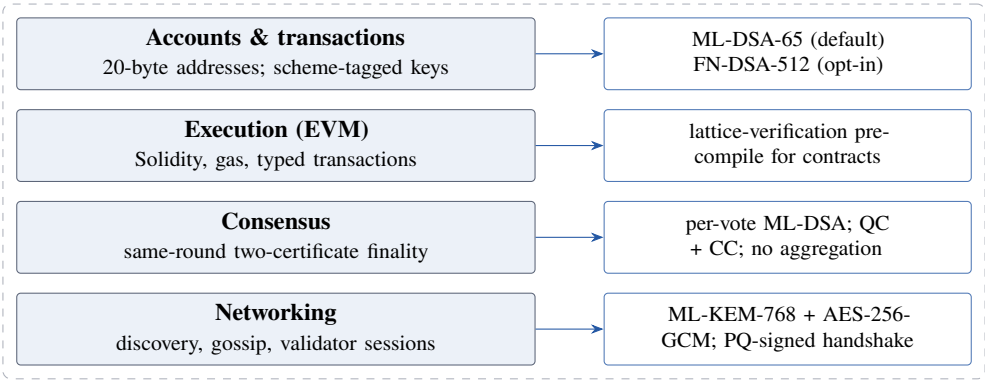

Figure 3: The post-quantum boundary, layer by layer. Accounts and transactions are authenticated by ML-DSA-65 (default) or FN-DSA-512 (opt-in); the EVM execution layer exposes lattice-signature verification to contracts through a precompile; every consensus vote and certificate is an individual ML-DSA signature; node sessions are keyed by ML-KEM-768 under AES-256-GCM with a post-quantum-signed handshake.

3 Cryptographic Foundation

3.1 Why lattices resist quantum attack

Shor's algorithm succeeds against factoring and discrete logarithms because those problems have hidden periodic structure that the quantum Fourier transform exposes. Lattice problems have no known analogous structure. Their security rests on the conjectured hardness of finding short vectors in high-dimensional lattices — problems studied since the 1990s [10] with no efficient quantum algorithm known, and equipped with worst-case-to-average-case reductions [13]: breaking random instances of the schemes below is provably as hard as solving the underlying lattice problem in its *worst* case.

Two problems over the module ring $R_q = \mathbb{Z}_q[X]/(X^{256} + 1)$ carry the load:

Module-LWE (Learning With Errors). Given a public matrix A and

$$t = As + e \pmod{q}$$

with short secret vectors s, e, recovering s is intractable. This hides secret keys inside public keys.

Module-SIS (Short Integer Solution). Given A, finding a short nonzero z with

$$Az = 0 \pmod{q}$$

is intractable. This underpins unforgeability: forging a signature without the trapdoor requires solving it.

3.2 ML-DSA — the default signature *(Implemented, alpha)*

ML-DSA (FIPS 204, formerly CRYSTALS-Dilithium [14]) is the network's default signature for accounts and validators, at parameter set **ML-DSA-65** (NIST category 3). Key generation expands a matrix $A \in R_q^{k \times \ell}$ from a seed, samples short secrets s_1, s_2, and publishes a commitment to

$$t = As_1 + s_2 \pmod{q}.$$

Signing follows Lyubashevsky's *Fiat–Shamir with aborts* paradigm [12]: sample a masking vector y, commit to the high-order bits of Ay, derive a sparse challenge polynomial c (exactly τ coefficients in $\{\pm 1\}$) by hashing, and compute the response

$$z = y + c\, s_1.$$

The signature is released only if z passes rejection bounds ($\|z\|_\infty < \gamma_1 - \beta$, among others); otherwise the attempt aborts and restarts. Rejection sampling is the essential idea — it makes the distribution of published signatures statistically independent of the secret key. Verification reconstructs the commitment from public data, $w' \approx Az - c\, t$, and checks challenge consistency and norms.

Parameter set	Category	(k, ℓ)	η	τ	γ_1	Public key	Signature
ML-DSA-44	2	(4,4)	2	39	2^{17}	1,312 B	2,420 B
ML-DSA-65	**3**	**(6,5)**	**4**	**49**	$\mathbf{2^{19}}$	**1,952 B**	**3,309 B**
ML-DSA-87	5	(8,7)	2	60	2^{19}	2,592 B	4,627 B

with $q = 8{,}380{,}417 = 2^{23} - 2^{13} + 1$, chosen for efficient NTT arithmetic. Two properties matter operationally. First, ML-DSA uses only integer arithmetic and uniform sampling — no floating point, no Gaussian sampler — which makes constant-time implementation tractable and avoids the side-channel class that has affected other lattice schemes. Second, signing is randomized ("hedged" per FIPS 204), providing defense in depth against weak randomness and fault attacks.

3.3 FN-DSA / Falcon — the compact option *(Implemented, alpha; opt-in)*

FN-DSA (the standardization name for Falcon [15]) offers the smallest combined key-plus-signature of any NIST lattice signature — decisive where bandwidth dominates. It operates over $R = \mathbb{Z}[X]/(X^n + 1)$, $q = 12{,}289$, with a secret NTRU basis [10] of short polynomials (f, g, F, G) satisfying

$$fG - gF = q \pmod{X^n + 1}, \qquad h = g \cdot f^{-1} \pmod{q},$$

where h is the public key. Signing follows the Gentry–Peikert–Vaikuntanathan hash-then-sign framework [11]: hash the message to a target $c \in R_q$ and use the trapdoor basis to sample a *short* pair (s_1, s_2) with

$$s_1 + s_2 h = c \pmod{q};$$

verification recomputes the equation and checks a squared-norm bound. The asymmetry is the security: sampling a short solution is easy with the secret basis and conjectured infeasible with only h.

Parameter set	Category	Public key	Signature (padded)
FN-DSA-512	1	897 B	666 B
FN-DSA-1024	5	1,793 B	1,280 B

Compactness carries implementation risk: Falcon's signer samples from a discrete Gaussian using high-precision floating-point arithmetic, historically a side-channel vector — single-trace key recovery has been demonstrated on embedded targets [16]. Constant-time behavior in available implementations is best-effort rather than guaranteed. FN-DSA is therefore an **opt-in** scheme on Atom Quantum, with ML-DSA — which has no Gaussian sampler at all — as the default; production custody of Falcon keys warrants platform-specific side-channel review.

3.4 ML-KEM — transport confidentiality *(Implemented, alpha)*

Node sessions are keyed by ML-KEM (FIPS 203, formerly CRYSTALS-Kyber [5]), a module-lattice key-encapsulation mechanism hardened to chosen-ciphertext security by a Fujisaki–Okamoto-style transform. For the default **ML-KEM-768**, the module has rank 3 over $\mathbb{Z}_{3329}[X]/(X^{256}+1)$. Peers derive a shared secret

$$(ct,\ ss) \leftarrow \text{Encaps}(ek), \qquad ss' \leftarrow \text{Decaps}(dk,\ ct), \qquad ss = ss',$$

which feeds a transcript-bound key schedule for an authenticated AES-256-GCM channel — the shared secret is never used directly as a traffic key.

Parameter set	Category	Public key	Ciphertext	Shared secret
ML-KEM-512	1	800 B	768 B	32 B
ML-KEM-768	**3**	**1,184 B**	**1,088 B**	**32 B**
ML-KEM-1024	5	1,568 B	1,568 B	32 B

3.5 The symmetric layer

Hashing uses Keccak-256; session encryption uses AES-256-GCM. The relevant generic quantum attack is Grover search, which reduces an exhaustive search of 2^n to on the order of

$$O\left(2^{n/2}\right)$$

oracle calls — a square-root speedup, not an exponential break. A 256-bit key thus retains an effective ~128-bit quantum search margin, and a 256-bit hash retains ~128-bit preimage resistance; collision resistance remains ~128 bits by the classical birthday bound, with known quantum collision algorithms impractical once memory costs are accounted for. This is why NIST retains well-sized symmetric primitives through the transition — and why hash commitments are a sound quantum-safe building block (Section 8.2).

3.6 NIST security categories

Category	Reference attack the scheme must resist
1	Key search on AES-128
3	Key search on AES-192
5	Key search on AES-256

Categories compare against reference attack costs; they are not literal counts of quantum gates, and NIST stresses that concrete quantum cost estimates remain model-dependent. Atom Quantum's defaults (ML-DSA-65, ML-KEM-768) target **category 3**; category-5 variants are available where maximum margin is wanted, and the category-1 FN-DSA-512 trades margin for compactness explicitly rather than silently.

3.7 Size as a protocol parameter

A single ML-DSA-65 authentication object — key plus signature —

$$1{,}952 + 3{,}309 = 5{,}261 \text{ bytes}$$

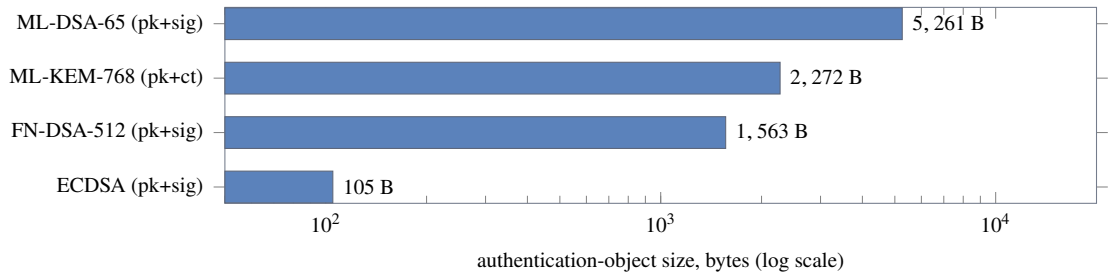

Figure 4: Authentication-object sizes, bytes (log scale): ECDSA pk + sig ≈ 105 B; FN-DSA-512 pk + sig = 897 + 666 = 1,563 B; ML-KEM-768 handshake objects pk + ct = 1,184 + 1,088 = 2,272 B; ML-DSA-65 pk + sig = 1,952 + 3,309 = 5,261 B. Sizes per FIPS 203, FIPS 204, and the Falcon specification (padded signature format).

is roughly **50× larger** than its ECDSA equivalent (∼105 bytes). FN-DSA-512 reduces the pair to 1,563 bytes at a lower security category (Fig. 4). The cost propagates to transaction size, block capacity at a fixed byte budget, gossip bandwidth, historical storage, and verification CPU — but **not** to account state: the public key travels in the transaction and is not persisted in the account record, so per-account state does not grow with signature size. Two design decisions follow directly from this table: the consensus committee is deliberately bounded (Section 4.5), and a compact signature family is offered at all (Section 3.3).

3.8 Account identity

An account address is a 20-byte commitment to its scheme-tagged public key: with a fixed derivation domain D and length-prefixed encoding,

$$\text{addr}(pk) = \text{LSB}_{160}\big(\text{Keccak256}(D \parallel \text{len}(pk^*) \parallel pk^*)\big),$$

where pk^* is the scheme-prefixed key. Addresses remain 20-byte and EVM-native — contracts, explorers, and wallets key off the same identifier format they always have — while the derivation commits to the signature scheme itself, so identical raw key bytes under two schemes yield different addresses. An address is a commitment, not a recoverable key: transactions supply the public key and prove that it derives the claimed address.

4 Consensus and Deterministic Finality

4.1 Protocol shape *(Implemented, alpha)*

Atom Quantum finalizes blocks with a **custom HotStuff-family Byzantine-fault-tolerant protocol** [17]. Within a single round, the scheduled leader proposes an executed block; validators verify and vote, forming a **quorum certificate (QC)**; validators then issue commit votes, forming a **commit certificate**; the pair finalizes *that same block*:

$$\text{proposal} \rightarrow \text{votes} \rightarrow QC \rightarrow \text{commit votes} \rightarrow CC \rightarrow \text{finality}.$$

Two properties distinguish the design. First, **every vote and certificate is an individual post-quantum (ML-DSA) signature** — there is no BLS and no cryptographic aggregation, so no pairing assumption has a path into the safety argument, and every signature is independently attributable, which makes equivocation

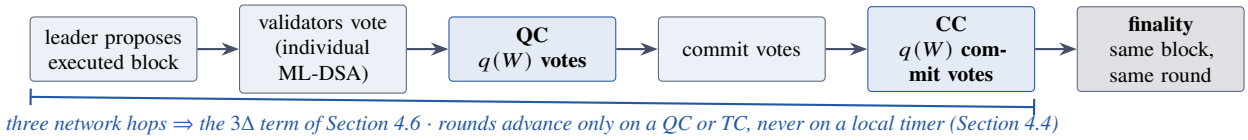

Figure 5: Same-round, two-certificate finality. Within one round: the scheduled leader broadcasts an executed proposal; each validator verifies and returns an individual ML-DSA vote; $q(W)$ votes assemble into a quorum certificate; validators answer the QC with commit votes; $q(W)$ commit votes assemble into a commit certificate; QC + CC finalize that same block.

evidence self-contained. Second, the commit rule is **same-round and two-certificate** rather than the chained rule of Jolteon or DiemBFT, in which phases pipeline across rounds and a certified descendant commits an ancestor [18, 19]. The protocol draws on that lineage but is its own state machine; correctness results proved for those protocols are informative, not inherited, and a formal specification with safety and liveness proofs is a stated pre-mainnet milestone (Section 10).

The finality proof binds the exact execution block hash and height, the epoch, the validator-set identity, and an execution-output commitment (state root, receipts root, gas), so finality certifies not just an ordering but the executed result.

4.2 Quorums and their intersection

Let validator i hold voting power $w_i > 0$ with total $W = \sum_i w_i$. The quorum threshold is

$$q(W) = \left\lfloor \frac{2W}{3} \right\rfloor + 1.$$

For any two quorums Q_1, Q_2:

$$w(Q_1 \cap Q_2) \;\geq\; w(Q_1) + w(Q_2) - W \;\geq\; 2\,q(W) - W \;>\; \frac{W}{3}.$$

If Byzantine power is strictly below $W/3$, every pair of quorums intersects in at least one honest validator — the structural fact from which BFT safety is built. Quorum intersection alone is necessary rather than sufficient; the protocol's voting and locking rules must ensure honest validators never contribute to conflicting commit paths, which is precisely what the planned formal analysis must establish.

4.3 Safety and liveness

The guarantees split in the standard BFT way:

- **Safety** — no two conflicting blocks are ever finalized — is designed to hold under *full asynchrony*: an adversary with complete control of message timing cannot cause a finalized block to be reverted, only delay progress.
- **Liveness** — the chain keeps finalizing — holds under *partial synchrony*: after an unknown stabilization time beyond which message delay is bounded by some Δ, an honest leader and a responsive two-thirds quorum suffice to finalize.

Both require Byzantine voting power strictly below $W/3$.

4.4 Progress by evidence, not by clocks

A defining liveness property: **rounds advance only on quorum evidence — a QC or a timeout certificate (TC) — never on a local timer.** A validator whose timer fires does not advance; it signs a timeout share for its current round and broadcasts it. Only when $q(W)$ timeout shares aggregate into a TC does every validator advance together. This eliminates the classic failure mode in which validators drift onto different rounds and reject each other's honest proposals; progress is a collective, evidence-driven decision. The pacemaker derives its timeout from the configured block interval and widens it with bounded backoff under repeated stalls, so it cannot time out before an honest leader could plausibly have produced a block.

4.5 The cost model, and why the committee is bounded

Individual post-quantum signatures cost more than one aggregate. With committee size n and quorum q:

$$|QC| = O(q), \qquad |CC| = O(q), \qquad C_{\text{verify}} = O(q), \qquad C_{\text{messages}} = O(n^2),$$

the last because votes, commit votes, and timeout shares are exchanged among all validators so that any of them can assemble a certificate. At 3,309 bytes per ML-DSA-65 signature, a certificate over even a small committee measures in tens of kilobytes, and finality carries two certificates. Atom Quantum therefore **bounds the active signing committee** and scales *economic* participation through delegation (Section 9) — many stakers backing a small, accountable signer set — rather than by enlarging the set itself. This is a deliberate trade: pairing-free, individually-attributable, fast finality in exchange for a bounded committee. Techniques for much larger signer sets (leader-side collection, committee sampling) are identified future work, not present claims.

4.6 Finality latency

Under a stable leader and post-stabilization delay bound Δ, the network component of same-round finality is approximately

$$T_{\text{final}} \;\approx\; T_{\text{build}} + T_{\text{execute}} + 3\Delta + T_{\text{verify}} + T_{\text{persist}},$$

the 3Δ term covering proposal, vote, and commit-vote propagation. On the running alpha, payload construction measures in single-digit milliseconds against a 250 ms budget; the block interval is a genesis configuration — conservative on the current testnet, one second on development networks — and the path toward sub-second operation is a matter of committee size, transport latency, and timing configuration rather than new research. Production latency figures will be published from measured runs identified by commit, topology, and genesis profile, not asserted in advance.

5 Execution and Accounts

5.1 EVM-compatible, deliberately not Ethereum-compatible *(Implemented, alpha)*

Atom Quantum retains the parts of the EVM ecosystem that carry developer value: EVM bytecode and Solidity; balances, nonces, and 20-byte addresses; gas accounting and EIP-1559-style dynamic fees; typed transactions and access lists [20]; transaction and receipt commitments; logs and events. Contracts and audit lineage deploy unchanged.

It deliberately replaces or omits everything bound to the old cryptography: secp256k1 sender recovery; Ethereum's transaction types as a public surface; devp2p transport and node identity; Gasper, fork choice, and BLS consensus; and blob transactions with KZG commitments — the latter because KZG rests on pairings and a trusted setup, the exact assumptions the network exists to avoid (Section 1.2). Should high-volume data availability be required later, the design constraint is fixed even where the mechanism is not: hash-based commitments and transparent, post-quantum-sound proof systems only.

5.2 Programmable accounts *(Implemented, alpha)*

Every account is authenticated by a single post-quantum key, and any account can *opt in* to delegation: authorizing code to execute in its context, with the authorization itself a post-quantum signature. Delegation is the base-layer mechanism beneath the account features users actually experience:

- **fee sponsorship** — an application or relay pays for a user's transaction; the fee payer and the authorizing account are cryptographically distinct roles;
- **batching** — approve-and-act sequences in one atomic transaction;
- **session keys** — narrowly scoped, expiring signing authority for an application;
- **recovery policies** — guardian- or policy-based restoration of account control, enforced by the account's own code.

This is a post-quantum analogue of Ethereum's account-delegation direction [21], native to the protocol rather than routed through an external bundler network. A plain account that never opts in remains a plain account.

5.3 Post-quantum verification for contracts *(Implemented, alpha)*

A dedicated precompile exposes lattice-signature verification to the EVM at a flat gas cost comparable to Ethereum's `ecrecover`. Any Solidity contract can verify an ML-DSA or FN-DSA signature and its binding to a claimed address — the primitive beneath permits, meta-transactions, signed off-chain orders, oracle attestations, and governance votes. On curve-based chains each of these patterns is quantum-exposed; here they inherit the network's assumption base.

6 Post-Quantum Networking

(Implemented, alpha.) Every connection between nodes is an authenticated, encrypted session: ML-KEM establishes the shared secret; a transcript-bound key schedule derives directional AES-256-GCM keys; post-quantum signatures authenticate the handshake transcript, preventing downgrade; records carry strictly ordered counters with deterministic nonces and bounded rekeying. Captured traffic yields nothing to a future quantum adversary — closing the harvest-now, decrypt-later channel at the network layer.

Traffic is segregated by role over this one substrate. Discovery peers exchange only signed node records; public execution peers carry transaction gossip, finalized blocks, and checkpoint state sync; validator sessions alone carry consensus traffic. Faults and reputation in the public planes are isolated from consensus liveness.

Public canonical history is the sequence of finalized blocks. A full client accepts a block only with its finality proof — verifying certificate membership, quorum weight, signatures, and continuity — and public

reads are served from that finalized view. New nodes bootstrap by checkpoint state sync anchored to a finality proof: verify the certificate chain, import state under it, join in minutes. Unfinalized proposals never leave the validator plane; there is no fork-choice ambiguity for the public to observe.

7 Security Model

This section consolidates the assumptions under which Atom Quantum's guarantees hold, the guarantees themselves, and — equally deliberately — the threats the protocol does **not** claim to address.

7.1 Assumptions

1. **Cryptographic.** The conjectured hardness of Module-LWE and Module-SIS (Section 3.1) at the stated NIST categories for ML-DSA and ML-KEM; the NTRU assumption for FN-DSA; the standard security of Keccak-256 and AES-256-GCM against Grover-bounded quantum search (Section 3.5). Cryptanalysis of any single family is hedged — not eliminated — by scheme identifiers and domain separation (Section 2), which make a controlled algorithm replacement possible.

2. **Consensus.** Byzantine voting power strictly below $W/3$ (Section 4.2). Safety requires no timing assumption; liveness additionally assumes partial synchrony (Section 4.3).

3. **Operational.** Validators custody their signing keys correctly; the current alpha additionally trusts the authority-bootstrapped validator set — an explicit, temporary centralization with an open, stake-selected transition on the mainnet track (Section 9).

7.2 Guarantees

Under those assumptions, the protocol provides: transaction authenticity and validator accountability against a CRQC-equipped adversary; irreversibility of finalized blocks under full asynchrony; transport confidentiality against harvest-now-decrypt-later capture; the absence of any pairing assumption or trusted setup from every native security boundary; and self-contained, independently attributable equivocation evidence (Section 4.1).

7.3 Out of scope

Atom Quantum does **not** claim to defend against:

- **Smart-contract defects.** Bugs, unsafe upgrade patterns, or flawed economic design in applications deployed on the EVM — including, for Quantum Vaults, everything about a home-chain vault contract other than the commitment scheme itself (Section 8.2).

- **Endpoint and key compromise.** Theft, malware, phishing, social engineering, or coercion at the user's or validator's device, process, or HSM. Recovery policies (Section 5.2) mitigate loss; they do not prevent compromise.

- **Side channels on uncontrolled platforms.** Implementation leakage on third-party hardware; FN-DSA custody in particular carries the explicitly stated side-channel caveats of Section 3.3.

- **Byzantine power at or above $W/3$.** The safety argument holds strictly below the threshold; no BFT protocol survives a supermajority-capturing adversary, and slashing is an incentive, not a safety mechanism (Section 9).
- **Targeted denial of availability.** Liveness is guaranteed after stabilization under partial synchrony (Section 4.3), not during an ongoing partition or eclipse of specific nodes.
- **Transaction-ordering fairness (MEV).** Leader rotation and timeout certificates bound censorship (Section 8.1), but the base layer does not implement an ordering-fairness or encrypted-mempool mechanism; venue-level ordering policy is application-layer work.
- **Economic-parameter risk.** Token economics are deliberately unspecified (Section 9); no incentive-compatibility claim precedes the published simulations.

7.4 Positioning in the post-quantum landscape

The industry is in an early, pre-dominant-design phase: several networks have deployed *partial* post-quantum protection, typically at the transaction layer only, as surveyed in [23]. Table 1 summarizes that survey and public project documentation as of mid-2026; "—" means the layer is not addressed by the project's deployed post-quantum work.

Network	Transaction authentication	Consensus authentication	Transport	EVM execution	Status
Ethereum	ECDSA (secp256k1)	BLS aggregation (pairing-based)	ECDH-keyed	native	PQ migration is an open research agenda
QRL	hash-based XMSS (since 2018), ML-DSA transition underway	—	—	no	longest-running PQ chain
Mochimo	hash-based WOTS+ (one-time)	—	—	no	live
Abelian	lattice-based	—	—	via QDay zk-rollup	live
Algorand	Falcon deployed (state proofs; first PQC-signed transaction 2025; key-rotation migration path)	—	—	no	partial, in production
XRP Ledger	ML-DSA	—	—	no	AlphaNet testnet
Solana	Winternitz Vault (hash-based, opt-in)	—	—	no	experimental
Atom Quantum	**ML-DSA-65 / FN-DSA-512**	**per-vote ML-DSA, no aggregation**	**ML-KEM-768 + AES-256-GCM**	**native**	**testnet alpha**

Table 1: The post-quantum chain landscape, per the survey in [23] and public project documentation as of mid-2026.

Within this landscape, Atom Quantum's position is the row no incumbent covers: post-quantum authentication at *every* layer — transactions, consensus, transport, and contract-level verification — combined with native EVM execution and deterministic, certificate-based finality.

8 Application Vision

Everything in this section is labeled **Vision**: designed against the implemented base layer, not yet deployed. It is included because it explains *why the base layer is shaped as it is* — deterministic post-quantum settlement is the common requirement of the four largest classes of on-chain value.

8.1 On-chain trading

The strongest recent demonstration in market structure is that a purpose-built L1 running a fully on-chain order book, on a HotStuff-family consensus engine, can capture the majority of decentralized perpetuals volume — with public reporting through mid-2026 placing the leading venue's lifetime volume in the trillions of dollars and its protocol fees beyond a billion at software margins. The architectural lesson generalizes: **an order book is a settlement-latency product**, and deterministic finality is the strongest settlement guarantee a venue can offer.

Atom Quantum's roadmap includes a native central-limit order book — spot and perpetuals — in the same architectural family, with properties the incumbents cannot retrofit:

- a fill is final in the round that matched it, with irreversibility bounded by the T_{final} model of Section 4.6 — no confirmation depth, no reorg risk on an open position;
- the full trade lifecycle — orders, cancellations, off-chain quotes verified through the contract-level precompile, oracle attestations, settlement — is authenticated by lattice signatures end to end;
- the sequencing set is the validator set itself: leader rotation and timeout certificates bound any single leader's ability to stall or censor, without a trusted external sequencer.

A production venue additionally requires a matching-determinism specification, oracle fault tolerance, margin and liquidation design, and MEV policy; these are acknowledged as their own engineering discipline, not corollaries of fast finality.

8.2 Quantum Vaults: post-quantum protection for other chains

Cross-chain infrastructure has lost more than $2.8B to attacks that overwhelmingly reduce to one flaw: a *standing secret* — a bridge key or multisig — whose compromise surrenders everything, and which is additionally ECDSA and therefore quantum-forgeable. Chain-analytics reporting placed key compromise as the largest single theft vector of 2024.

The Quantum Vault primitive removes the standing secret. Assets remain on their home chain in a vault contract that stores only a hash commitment. For a withdrawal with parameters (v, r, a, n) — vault, recipient, amount, nonce — and a one-time key k:

$$C = H(v \,\|\, r \,\|\, a \,\|\, n \,\|\, k).$$

To withdraw, the owner authorizes the exact parameters with a lattice signature *on Atom Quantum*; finality seals the authorization in a quorum certificate; a post-quantum oracle committee independently verifies the certificate and threshold-releases k; the vault checks the commitment and pays out precisely (r, a). The security argument is structural:

- **the lock is a hash**, and by Section 3.5 Grover's square-root speedup is absorbed by output size — so the vault is quantum-safe *even though it lives on a quantum-vulnerable chain*;
- **no persistent secret exists** to steal, phish, or subpoena — k exists only for the instant of one authorized withdrawal and is spent on use;
- **a captured k redirects nothing**, because the commitment binds recipient and amount;

- **release is finality-gated, not signature-gated** — it requires an objective quorum-certificate proof, not the say-so of a relayer that can be socially engineered.

The primitive generalizes: any two chains can adopt Atom Quantum finality as a shared root of trust in place of the multisig signer sets that have repeatedly failed, positioning the network as security infrastructure *between* chains rather than only a destination.

8.3 Dollar rails

Stablecoins settled more value on-chain in 2025 than the major card networks combined, on circulating supply around $315B, and now carry U.S. federal legal recognition; the market has priced the *rails* themselves in the billions. Every incumbent rail runs on quantum-vulnerable curves — while its target customers, banks and regulated institutions, operate under explicit post-quantum migration mandates (Section 1.4), including EU requirements for critical financial systems by 2030.

The mechanics a durable dollar rail requires are already native to Atom Quantum's account layer: fee sponsorship makes a stablecoin transfer free to the sender as a protocol capability rather than a subsidy hack; deterministic finality settles it irreversibly in seconds; account-level policy (limits, session keys, recovery) gives self-custody bank-account ergonomics under lattice keys. The remaining work is productization and distribution — issuer integrations, card-network partnerships at the edge, and ISO 20022 compatibility at the RPC boundary, where SWIFT reports the standard now carries roughly 90% of global transaction value. The strategic question a stablecoin rail must answer is no longer only *how fast* — it is *which of these systems still satisfies its own regulators in 2035*.

8.4 Tokenized real-world assets

Institutional forecasts for tokenized securities run from the trillions (Citi, 2030) to tens of trillions (Standard Chartered, 2034) from a base of roughly $33B on-chain today; institutional surveys name **security** as the leading adoption barrier, and the IMF identifies *legally recognized settlement finality* as a precondition for scale. Both requirements point at the same design: deterministic finality — a cryptographic certificate of settlement rather than a probability — on cryptography that regulators are mandating *toward* rather than away from. A ten-year instrument tokenized on a chain whose signature scheme is scheduled for deprecation within four is a compliance liability with a date attached; the unmodified EVM means issuers' existing Solidity stacks deploy on Atom Quantum unchanged.

9 Economic Framework

The native asset **ATQM** is the unit in which gas is paid, consensus security is bonded, and protocol fees settle. This revision publishes the **preliminary token allocation** (Section 9.1) as provided by the project. Issuance rate, burn parameters, and staking targets remain **unspecified**: they require economic simulation and legal review before being represented as protocol facts. All allocation figures are draft values in a document still under revision and remain subject to change until genesis economics are published with the simulations behind them (Section 10, Phase 3).

Security budget. Consensus security scales with the value bonded to it. For circulating supply S, staked fraction σ, issuance rate ρ, aggregate fees F, and burned fee fraction b, the staking yield satisfies

approximately

$$y \approx \frac{\rho S + (1-b)\,F}{\sigma S}.$$

The design levers — issuance ρ, burn fraction b, and target staking ratio σ^* — jointly determine the security budget and net issuance $\rho S - bF$, which a fee-burn component ties to network usage so that security is funded without unconditional inflation. Post-quantum signature bytes are priced explicitly in the fee schedule (Section 3.7), so the chain's dominant physical cost is borne by its users rather than externalized.

Staking architecture. Staking is separated from consensus: economic state (delegations, bonds, commissions, unbonding queues) determines, at each epoch boundary, the active validator set and voting-power vector that consensus consumes. Consensus never sees individual delegators. Equivocation — double voting or double commit-voting — yields self-contained cryptographic evidence (Section 4.1) that feeds slashing; safety never depends on slashing, which is an incentive mechanism, not a safety mechanism. The current alpha runs an authority-bootstrapped validator set — an explicit, temporary centralization stated as such — with the open, stake-selected transition a mainnet-track milestone.

9.1 Preliminary token allocation *(draft; subject to change)*

Total supply is **17,000,000,000 ATQM**. The projected market capitalization at the token generation event (TGE) is **$5,355,956,187**; the initial circulating supply at TGE is **9,698,818,750 ATQM (57.05%)**. Figures are as provided by the project's allocation schedule; percentages are rounded, token counts are exact and sum to the total supply.

Allocation	Purpose	Share	Tokens	Initial release	TGE tokens	Cliff (mo)	Vesting (mo)
Seed / early investors	Boostyfi early stage	0.43%	72,250,000	100.0%	72,250,000	0	0
Private round	Strategic investor allocation	4.00%	680,000,000	100.0%	680,000,000	0	0
Convertible round	ATLA, JGGL, TEKI, IMBA — converts	41.50%	7,055,000,000	100.0%	7,055,000,000	0	0
OTC round	Private placement to a limited number of investors	10.00%	1,700,000,000	100.0%	1,700,000,000	0	0
Team & advisors	Core contributors and consultants allocation	10.50%	1,785,000,000	0.0%	0	48	60
Ecosystem growth	Administered by the Ecosystem Council: incentives, liquidity, listings, market making, staking rewards	8.50%	1,445,000,000	2.5%	36,125,000	6	78
Liquidity programme fund (DEX + staking)	LP farming and staking rewards	5.00%	850,000,000	5.0%	42,500,000	24	60
Emergency security reserve	Critical protocol exploits, validator recovery, bridge/security incidents, network stabilization	2.50%	425,000,000	10.0%	42,500,000	0	84
Post-quantum R&D, audits & security	Post-quantum cryptography research, independent security audits, formal verification	5.00%	850,000,000	2.0%	17,000,000	0	84
Foundation treasury fund	Long-term protocol treasury: ecosystem grants, strategic partnerships, protocol operations	12.58%	2,137,750,000	2.5%	53,443,750	0	84
Total		**100.00%**	**17,000,000,000**	**57.05%**	**9,698,818,750**		

Table 2: Preliminary token allocation and unlock schedule (cliff and vesting in months; draft, subject to change).

Two structural properties are worth stating alongside the raw schedule. First, the allocations that fund long-horizon security obligations — the emergency security reserve, post-quantum R&D and audits, and

the foundation treasury — carry the longest vesting (84 months), aligning their release with the multi-year security program of Section 10. Second, the team allocation unlocks nothing at TGE and sits behind a 48-month cliff, so no insider supply precedes the protocol-hardening and mainnet milestones.

10 Roadmap

Milestones are ordered by security dependency: protocol correctness precedes economic opening, which precedes application layers.

Phase 1 — Alpha *(current).* Post-quantum accounts, transactions, contract verification, and transport; deterministic finality; programmable accounts; checkpoint sync; explorer (https://scan.atqm.network/) and wallet; sustained adversarial testnet operation.

Phase 2 — Protocol hardening. Frozen wire formats and signing rules; a published, implementer-grade consensus specification with safety and liveness analysis and adversarial model checking; independent cryptographic, consensus, and networking audits; reproducible performance benchmarks identified by commit, hardware, genesis profile, committee size, and workload.

Phase 3 — Mainnet and open participation. On-chain staking, delegation, rewards, unbonding, and slashing; epoch-driven validator-set transitions from staking state; genesis economics published with the simulations behind them.

Phase 4 — The exchange. Native order book, spot and perpetuals; validator-attested oracle quorum; margin and risk engine; market-maker onboarding.

Phase 5 — Settlement products. Quantum Vault contracts and oracle network on major external chains; stablecoin sponsorship rails; ISO 20022-compatible institutional access.

Each phase gates on published evidence — audits, proofs, and measurements — rather than on announcements.

11 Conclusion

The cryptography under every major blockchain has a mandated retirement window, and the exposure it leaves behind is retroactive: keys already public stay forgeable forever. A durable ledger must therefore replace quantum-vulnerable assumptions everywhere they appear — accounts, consensus, networking, and contract authorization — and must do it before the margin narrows, because migration under pressure is the failure mode, not the plan.

Atom Quantum demonstrates that this replacement is compatible with the things that make blockchains useful today: EVM execution and its tooling, programmable accounts, familiar fees, and settlement measured in seconds — delivered as deterministic certificates rather than probabilities, on lattice problems with three decades of cryptanalytic history and NIST standardization behind them. The network runs; its mathematics are stated in this paper; its open problems are listed next to its claims. On that foundation, the applications that most need settlement to outlive the decade — trading venues, cross-chain security, dollar rails, tokenized assets — have a place to be built.

The quantum clock is running. The purpose of this work is that, when it matters, the destination already exists.

References

[1] P. W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," *FOCS*, 1994. https://doi.org/10.1109/SFCS.1994.365700

[2] M. Roetteler, M. Naehrig, K. M. Svore, K. Lauter, "Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms," *ASIACRYPT*, 2017. https://eprint.iacr.org/2017/598

[3] C. Gidney, "How to factor 2048-bit RSA integers in under a week using fewer than a million noisy qubits," 2025.

[4] M. D. Mascelli, T. Rodden, "Harvest Now, Decrypt Later: Examining Post-Quantum Cryptography and the Data Privacy Risks for Distributed Ledger Networks," Federal Reserve FEDS 2025-093, 2025.

[5] NIST, *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM)*, August 2024. https://doi.org/10.6028/NIST.FIPS.203

[6] NIST, *FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA)*, August 2024. https://doi.org/10.6028/NIST.FIPS.204

[7] NIST, *FIPS 205: Stateless Hash-Based Digital Signature Standard (SLH-DSA)*, August 2024. https://doi.org/10.6028/NIST.FIPS.205

[8] NIST, *IR 8547 (Initial Public Draft): Transition to Post-Quantum Cryptography Standards*, 2024. https://doi.org/10.6028/NIST.IR.8547.ipd

[9] NIST, Post-Quantum Cryptography Standardization: FIPS 206 (FN-DSA) status. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography

[10] J. Hoffstein, J. Pipher, J. H. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem," *ANTS-III*, 1998.

[11] C. Gentry, C. Peikert, V. Vaikuntanathan, "Trapdoors for Hard Lattices and New Cryptographic Constructions," *STOC*, 2008. https://eprint.iacr.org/2007/432

[12] V. Lyubashevsky, "Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures," *ASIACRYPT*, 2009.

[13] A. Langlois, D. Stehlé, "Worst-Case to Average-Case Reductions for Module Lattices," *Designs, Codes and Cryptography*, 2015.

[14] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé, "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme," *TCHES*, 2018. https://eprint.iacr.org/2017/633

[15] Falcon Project, *Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU — Specification v1.2*, 2020. https://falcon-sign.info/falcon.pdf

[16] J. Qiu, A. Aysu, "SHIFT SNARE: Uncovering Secret Keys in FALCON via Single-Trace Analysis," 2025. https://arxiv.org/abs/2504.00320

[17] M. Yin, D. Malkhi, M. K. Reiter, G. Golan-Gueta, I. Abraham, "HotStuff: BFT Consensus in the Lens of Blockchain," *PODC*, 2019. https://arxiv.org/abs/1803.05069

[18] R. Gelashvili, L. Kokoris-Kogias, A. Sonnino, A. Spiegelman, Z. Xiang, "Jolteon and Ditto: Network-Adaptive Efficient Consensus with Asynchronous Fallback," *Financial Cryptography*, 2022. https://arxiv.org/abs/2106.10362

[19] The Diem Association, *DiemBFT v4: State Machine Replication in the Diem Blockchain*, 2021.

[20] Ethereum Improvement Proposals: EIP-1559 (fee market); EIP-2718 (typed transaction envelope). https://eips.ethereum.org

[21] V. Buterin, S. Dietrichs, M. Schor, A. Wahrstätter, "EIP-7702: Set Code for EOAs," Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-7702

[22] G. Wood, *Ethereum: A Secure Decentralised Generalised Transaction Ledger* (Yellow Paper). https://ethereum.github.io/yellowpaper/paper.pdf

[23] R. Babbush, A. Zalcman, C. Gidney, M. Broughton, T. Khattar, H. Neven, T. Bergamaschi, J. Drake, D. Boneh, "Securing Elliptic Curve Cryptocurrencies against Quantum Vulnerabilities: Resource Estimates and Mitigations," Google Quantum AI, March 2026.

Market and industry figures in Sections 1 and 8 (tokenization forecasts, stablecoin settlement volumes, bridge-loss totals, migration mandates) are drawn from public reporting by NIST, the Federal Reserve, the IMF, Citi Institute, Standard Chartered, State Street, Chainalysis, and SWIFT, as of July 2026, and are contextual rather than normative.

A Parameter Reference

Rings and moduli. ML-DSA: $R_q = \mathbb{Z}_q[X]/(X^{256}+1)$, $q = 8{,}380{,}417$. FN-DSA: $R = \mathbb{Z}[X]/(X^n+1)$, $q = 12{,}289$, $n \in \{512, 1024\}$. ML-KEM-768: rank-3 module over $\mathbb{Z}_{3329}[X]/(X^{256}+1)$.

ML-DSA-65 (default): $(k, \ell) = (6, 5)$, $\eta = 4$, $\tau = 49$, $\beta = 196$, $\gamma_1 = 2^{19}$, $\omega = 55$; category 3.

Consensus. Quorum $q(W) = \lfloor 2W/3 \rfloor + 1$; safety envelope: Byzantine power $< W/3$; certificates $O(q)$; vote traffic $O(n^2)$; finality $T_{\text{final}} \approx T_{\text{build}} + T_{\text{execute}} + 3\Delta + T_{\text{verify}} + T_{\text{persist}}$.

Sizes. See tables in Sections 3.2–3.4 and 3.7.

Economics. Yield model $y \approx (\rho S + (1-b)F)/(\sigma S)$; issuance/burn/staking parameters unset (Section 9). Preliminary allocation: total supply 17,000,000,000 ATQM; TGE circulating 9,698,818,750 (57.05%); see Section 9.1 (draft, subject to change).

B Glossary

Term	Meaning
BFT	Byzantine Fault Tolerance
CC / QC / TC	Commit / Quorum / Timeout Certificate
CRQC	Cryptographically Relevant Quantum Computer
ECDLP	Elliptic-Curve Discrete-Logarithm Problem
FN-DSA	Falcon lattice signature (FIPS 206, in development)
KEM	Key-Encapsulation Mechanism
ML-DSA	Module-Lattice Digital Signature Algorithm (FIPS 204)
ML-KEM	Module-Lattice Key-Encapsulation Mechanism (FIPS 203)
Module-LWE / Module-SIS	Lattice problems underlying ML-DSA and ML-KEM
NTRU	Lattice family underlying Falcon

This document is a draft under active revision. It describes design intent and a running test network; it is not an offer of securities, and token parameters — including the preliminary allocation of Section 9.1 — remain subject to change. Forward-looking items are labeled Vision and are plans, not commitments.