Quantum Research Implementation and Verification Plan

Phexora Cash — unreleased evidence-plan clarifications to research design 0.5.4

Phexora AI · phexora.ai

2026-11-08

Abstract

This document turns the Quantum research requirements into a human-owned, dependency-ordered implementation and verification plan. It defines reviewer roles, evidence classes, task gates, adversarial scenarios, performance reporting, and release criteria for a candidate private post-quantum DAG protocol. It is a planning and assurance document; it does not report a conformant implementation or completed verification.

Keywords: post-quantum cryptography; protocol verification; evidence gates; reproducible benchmarking; independent review.

1. Purpose and status

Programme identity: This is the Phexora Cash verification plan. Quantum remains the historical protocol codename used in the plan and stable filenames. The display-name change leaves protocol identifiers, task and scenario IDs, reviewer roles, requirements and acceptance gates unchanged.

This working copy contains unreleased clarifications to the 0.5.4-research baseline, dated 2026-11-08. Campaigns must pin exact source hashes and obtain the existing owner/reviewer approvals; these edits do not approve a profile.

This guide turns the Quantum research requirements into an ordered body of work. It does **not** certify the design, prove security, or report completed tests. At this revision:

- there is no protocol implementation in this repository;
- none of the task acceptance gates has passed;
- no benchmark has been run;
- no external cryptographic or consensus review has been completed;
- there is no conformant testnet or production network.

The goal is a protocol that simultaneously satisfies:

1. end-to-end post-quantum security;
2. private ledger semantics and network anonymity by default;
3. at least 1,000 accepted layer-1 transactions per second;
4. independently bounded executing validation and recovery; and
5. contestable block production with explicitly reviewed long-run security funding.

These are hard release requirements. They are not current capability claims. The initial product boundary is private post-quantum cash and settlement with bounded payment policies, not a general-purpose smart-contract platform.

2. Governance and reviewer roles

AI systems may help draft code, tests, vectors, and analyses. They MUST NOT approve their own cryptographic design, formal argument, benchmark, audit finding, or release.

Every task has:

- an accountable human owner;
- an independent reviewer who did not author the work;

- explicit dependencies;
- versioned artifacts;
- a pass/fail decision with evidence;
- unresolved-risk and change-control records.

The minimum specialist review roles are:

Role	Approval boundary
Cryptography reviewer	primitives, parameters, KATs, composition
Proof-system reviewer	AIR, transcript, FRI, ZK, soundness
Consensus reviewer	GHOSTDAG profile, DAA, ordering, finality
Data-availability/storage reviewer	current block data, snapshots, recovery encoding, reconstruction, repair
Mining/pool-incentive reviewer	templates, job declaration, shares, payouts, ordering, censorship, pool formation
Monetary-economics reviewer	cap semantics, issuance, fees/burns, miner entry/exit, security-budget models
Network-privacy reviewer	observer model, routing, traffic analysis
Implementation-security reviewer	parsers, side channels, supply chain
Performance reviewer	operability profile, workload, topology, measurement validity
Legal counsel	launch, token and monetary claims, privacy, sanctions and jurisdiction questions

No single reviewer may sign all technical boundaries. Any change to cap semantics, token issuance, burns, or the public 21-million statement requires the named Product Owner, consensus reviewer, monetary-economics reviewer, and Legal Counsel to sign the exact revision.

3. Evidence taxonomy

Reports MUST label evidence accurately:

Evidence type	What it can support	What it cannot support alone
Analytical proof	a stated theorem under stated assumptions	implementation correctness
Formal verification	properties encoded and proved in a named formal tool	omitted properties or real hardware behavior
KAT/conformance vector	exact algorithm and encoding agreement	general security
Unit/property test	exercised implementation behavior	negligible failure probability or a reduction
Differential/fuzz test	cross-implementation mismatch and parser defects	absence of defects
Statistical experiment	empirical distribution or attack result	binding, zero knowledge, or anonymity as a theorem
Benchmark	performance for a pinned environment/workload	performance on other environments
Code review	implementation findings	cryptographic proof
External audit	independent review within its stated scope	guarantees outside that scope

Miri, sanitizers, linters, tests, and manual review are valuable assurance tools but MUST NOT be called formal verification. One million collision-free trials do not prove commitment binding. A prover test with no observed false acceptance does not establish 2^-128 soundness. Uniform-looking ciphertext bytes do not prove privacy.

4. Requirements traceability

Requirement	Primary tasks	Required release evidence
R1 Private ledger	T006, T103–T104, T301–T305, T401–T405	note-instance spendability, security argument, vectors, two implementations, audit
R2 Network anonymity	T006, T601–T602, T604	early cost screen, threat model, simulation, adversarial experiments, review
R3 Post-quantum security	T001, T103–T104, T201–T204, T303–T306, T505, T510, T601, T701	composed 128-bit PQ analysis and KATs
R4 Authorization/supply	T202, T301, T305, T402–T405, T504, T506	prevention of unauthorised issuance, exact accounting, and selected monetary-invariant proof
R5 Consensus safety	T403, T501–T505, T507–T508, T603	deterministic models, vectors, state/reorg and template tests
R6 Scalability	T004–T006, T304–T306, T603–T605	pre-node receiving/transport and transaction feasibility; 1,000 accepted tx/s end-to-end artifacts
R7 Setup/upgrade safety	T002, T202, T204, T302–T306, T403, T501–T502, T505, T510, T701	setup/accumulation analysis, historical-note/nullifier continuity, versioned PQ upgrade-governance evidence
R8 Verifiability	T003–T005, all implementation tasks, T701	trace report, independent implementations, external review
R9 Operability	T004–T005, T305, T403–T404, T505, T603–T605	frozen profile, paired G7/G10 artifacts, bootstrap/restart evidence
R10 Current data/recovery	T403–T404, T505, T603–T605	complete-current-data, snapshot, withholding, reconstruction, and repair evidence
R11 Producer contestability	T005, T501–T503, T507–T509, T604	pre-registered thresholds, miner templates, pooled-mining comparator, incentive and hardware evidence
R12 Monetary security	T504, T506, T509–T510, T701	selected policy/schedule/fee/burn decision, invariant proof, protected-upgrade boundary, pre-registered stress model, signed reviews

A task is not complete if its output is not linked from this matrix through a machine-readable evidence manifest.

5. Acyclic task graph

Tasks MUST be completed in dependency order. A later experiment may send an earlier task back to design, but dependencies MUST NOT be bypassed.

```
Layer 0 - Scope and evidence
T001 Security profile and threat model
```

```
T002 Common encoding rules and domain registry          <- T001
T003 Test-vector and interoperability harness           <- T002
T004 Benchmark and artifact harness                     <- T001, T002
T005 Operability-profile freeze                         <- T001, T002, T004

Layer 1 - Algebra and commitment
T101 Field and extension-field arithmetic               <- T001, T003
T102 Polynomial and NTT layer                            <- T101
T103 Samplers and commitment candidate                  <- T001, T002, T101; T102 if required
T104 Commitment security gate                           <- T001, T003, T103

Layer 2 - Standardized cryptography and keys
T201 SHAKE256 and domain hashing                        <- T002, T003
T202 Hash-based authorization comparison               <- T201
T203 ML-KEM, KDF and authenticated encryption           <- T201
T204 Wallet key hierarchy and address profile           <- T202, T203

Layer 3 - Validity proof
T301 Integer transaction relation and AIR               <- T101, T104, T201, T202, T401, T402
T302 Membership, nullifier and state AIR                <- T301
T303 Fiat-Shamir, FRI and zero-knowledge profile        <- T101, T201
T304 Prover, verifier and aggregation                   <- T301, T302, T303
T306 Proof-system accumulation capability               <- T303, T304, T005

Layer 4 - Transactions and state
T401 Note creation and recipient encryption              <- T104, T203, T204
T402 Transaction codec, builder and authorization hash   <- T002, T201, T202, T401
T006 Early receiving/transport feasibility screen        <- T004, T005, T401, T402
T305 Cryptographic transaction feasibility gate          <- T004, T005, T006, T202, T204, T304, T402
T403 Commitment/nullifier state and atomic application   <- T302, T402, T305
T404 Wallet scanning and spending                        <- T204, T401, T402, T403
T405 Selective disclosure and bounded payment policies   <- T204, T304, T404

Layer 5 - Consensus and issuance
T501 Versioned GHOSTDAG consensus profile                <- T001, T002
T509 PoW hardware contestability study                   <- T001, T004, T005, T201, T501
T502 Header, selected PoW and deterministic DAA          <- T201, T501, T509
T510 Post-quantum upgrade governance and activation      <- T001, T002, T202, T204, T302, T403, T501, T502
T503 Canonical DAG-to-state ordering                     <- T403, T501, T502
T506 Monetary-security decision                          <- T001, T004, T005, T501, T509
T504 Rewards, selected supply policy and genesis         <- T503, T506
T505 Consensus-bound state validity and bootstrap        <- T306, T405, T503, T504
T507 Miner-template and pooled-mining profile            <- T501, T502, T503, T504
T508 Ordering, censorship and incentive gate             <- T005, T503, T506, T507

Layer 6 - Network and integration
T601 Post-quantum authenticated P2P transport             <- T202, T203, T501
T602 Network anonymity protocol                          <- T004, T005, T601
T605 Current-data availability, snapshot and recovery    <- T002, T004, T005, T505, T601
T603 Full node, pruning and recovery                     <- T305, T403, T505, T601, T605
T604 End-to-end system validation                        <- T405, T508, T602, T603

Layer 7 - Independent assurance
T701 Composition review and release decision             <- T510, T604
```

T004 owns the method and artifact harness; T005 freezes the operability profile before later results are interpreted. T104's security decision is independent of those performance thresholds; commitment performance is measured by T305. T306 evaluates only generic proof-system accu-

mulation capability before T505 selects the exact consensus-bound relation after T405, T503, and T504. T509 precedes PoW selection. T510 owns post-quantum upgrade authorisation and activation. T605 remains outside proof soundness and precedes full-node recovery. T701 depends directly on T510 and the integrated T604 result; T604's dependency closure already contains every other implementation and evidence task.

Layers group responsibilities, not execution rounds: cross-layer prerequisites are intentional. In particular, T401 and T402 finish before T301; T006 screens the frozen output/packet envelope before T305; T403 begins state integration only after T305. T002 freezes common rules, while each domain task freezes its own detailed interface before implementation. T401 accepts the abstract encryption context defined in specification Section 6.1; T402 supplies its canonical derivation, so T401 does not depend on the later codec. T402 encodes the effects and an opaque, bounded proof-envelope interface; T304 later freezes and implements its concrete proof representation. There is one codec, reused by T305 and the integrated system, not a prototype-only transaction contract.

T003/T004 pass on independently exercised harness/calibration fixtures, not on future protocol results. Component conformance and real workloads are accepted by their later tasks. T202 reports primitive conformance and a circuit interface; T304/T305 own measured in-proof costs. A domain interface change invalidates its dependent evidence and requires a new campaign, never a hidden back edge. For each monetary campaign, T506 freezes the exact fee, controller, pool, and burn rule before T504. T508 may reject but cannot edit that rule in place; a rejection starts a new versioned T506 campaign and rebuilds dependent T504/T507/T508 evidence. This campaign reset is not a reverse graph edge. The graph is acyclic and retains the earlier correction that polynomial multiplication does not depend on an NTT task which depends on it.

6. Task specifications

T001 — Security profile and threat model

Objective: Freeze the security games, adversaries, corruption thresholds, privacy boundaries, lifetime, multi-user counts, and minimum composed post-quantum security target.

Deliverables:

- cryptographic, consensus, network, side-channel, operational, data- availability, mining/pool-incentive, and monetary adversary models;
- explicit definition of ledger privacy, recipient-key privacy, and network anonymity;
- per-component and composed security budget;
- explicit justification of ML-KEM and authorisation parameters against R3.1, distinguishing NIST security categories from concrete post-quantum attack costs after composition; smaller parameter sets require a separate reviewed profile and cannot inherit the target from their name or category;
- quantum collision, preimage, and second-preimage cost models, including time/memory assumptions, multi-target loss, and required digest lengths;
- wallet target counts and the source-entropy bound required of the complete wallet master secret over the intended protocol lifetime;
- signed prior-art, reproducibility, reuse-rights, and adopt/adapt/replicate decision for every construction selected for implementation;
- exclusions such as compromised endpoints, with user-facing claim wording.

Pass: independent reviewers agree the model is complete enough to analyze R1–R12. **Reject:** "quantum-safe," "anonymous," "decentralised," "MEV-free," "ASIC-resistant," or "sustainable security" appears without a defined game, adversary, metric, and boundary.

T002 — Common encoding rules and domain registry

Objective: Freeze common encoding rules and the domain-registry contract; require each later domain owner to freeze its exact fields before consumer code.

Deliverables:

- endian, length-width, canonicality, rejection, version and chain rules;
- an owner map for detailed fields/maxima: T103 commitments, T202–T204 keys, T401 notes/encryption, T402 transaction effects/digests, T303/T304 proofs, T502 headers and T504 monetary/genesis data;
- registry for transaction ID, authorization, note, nullifier, Merkle, transcript, block, PoW, KDF, address, and upgrade domains;
- registry fields for output length and security property, which each named owner must instantiate and review before that hash domain is consumed;
- parser limits and cost ordering;
- replay and downgrade rules.

Pass: independent teams agree on common-rule vectors and each detailed format has a named freeze task before its consumer. This task cannot certify not-yet-selected proof, commitment or header bytes. **Reject:** implicit serialization, trailing bytes, unbounded vectors, unknown-version fallback, unowned future fields, or one hash reused across purposes.

T003 — Test-vector and interoperability harness

Objective: Build the reproducible owner for official KATs and Quantum cross-language vectors.

Deliverables:

- provenance for each external vector;
- canonical JSON or another frozen, integer-safe vector format;
- positive and negative vectors;
- deterministic generator revision and independently checked expected values;
- runner contract for at least two implementations.

Pass: independent harness implementations agree on the common-rule and calibration vectors and reject their negative cases. Later component tasks must supply and pass their actual protocol vectors. Generated expected values MUST NOT be produced by the same code path being tested without independent confirmation.

T004 — Benchmark and artifact harness

Objective: Make all performance, storage, and network evidence reproducible.

Deliverables:

- pinned operating systems, compilers, CPU features, memory, disks, network topology, clock source, and revisions;
- workload schema, generator interface and calibration fixtures for later private notes, proofs, conflicts, malformed traffic, delay and reorgs;
- separate channels for current transaction/proof/state data, gossip copies, current authenticated state, prunable history, durable recovery data, wallet maintenance, and provider traffic;
- power, thermal, disk-endurance, retail-availability, connectivity, and bootstrap measurement methods;
- miner-template/pool overhead, inclusion/censorship, provider concentration, archive repair/reconstruction, and economic-simulation artifact methods;
- raw metrics, logs, hashes, manifests, and report generator;
- accepted-state-transition throughput as the primary metric.

Pass: an independent lab reproduces the harness's calibration result and validates counters, timing, resource attribution and artifact integrity. Protocol runs must later use real task-owned implementations and pass their own frozen tolerances. **Reject:** "latest stable," unpublished cloud shape, submitted TPS, warm-cache-only measurement, or missing raw artifacts.

T005 — Operability-profile freeze

Objective: Freeze the executing-validator profile and every material operability threshold before implementation results are interpreted.

Deliverables: CPU architecture and physical cores; memory; disk capacity, sustained I/O and endurance; network; power method; retail price and regional availability method; accelerator policy; bootstrap bytes/time; maximum consensus-transaction, external encrypted-payload, total-payment, proof, state/witness and block sizes; wallet-maintenance traffic; payload-provider and retrieval overhead; immutable profile identifier; signed pre-registration; and the G12 metrics, scenario definitions, pass/fail thresholds, and STOP conditions for custom-template latency and revenue penalty, direct-publication success, inclusion delay under defined censorship, pooled-mining overhead, specialised-hardware efficiency, and manufacturer, fabrication, supplier, cloud, and pool concentration.

T005 also freezes the T006 envelope: candidate output-format constraints, conservative not-yet-measured proof/packet sizes, outputs per second, client/provider counts, offline intervals, scan and retrieval bytes/time/energy, missed-note and false-positive limits, observer advantage, cover traffic, idle cost, and delivery/latency bounds. Each modelled component must be identified. This envelope is instantiated with T401/T402 artifacts in T006 and rechecked against measured T305 sizes before integration. T005 does not require the later output codec or complete proof implementation to exist.

At least three independently obtainable observations in multiple regions are required for any hardware, connectivity, or price claim. Calibration evidence must be labelled and cannot be reused as the passing campaign after a profile is changed.

Pass: the accountable owner and independent performance reviewer freeze one profile before T006, T305, T306, T507–T509, or T602–T605 results are interpreted. **Reject:** a threshold follows the observed result; cloud burst capacity is hidden; required accelerator, prover, state provider, or archive work is assigned outside the measured role; or a new profile reclassifies an old campaign.

T006 — Early receiving/transport feasibility screen

Objective: Detect an infeasible receiving or anonymous-transport cost envelope before the complete T305 proving campaign and state/node integration.

Deliverables: T004 method and signed T005 bounds; real T401 output generation and recipient scan/decryption fixtures using T402 context bytes; full-scan baseline and any proposed private-retrieval comparator; online and offline catch-up measurements; client and provider compute, bytes, storage, energy and query-leakage accounting; malicious-output and missed-note cases; packet-size/traffic simulation for batching, mixing, cover traffic, idle clients, failures and correlated retries under the T001 observer; raw artifacts; and independent performance and network-privacy review. Proofs or transport components not yet implemented are explicitly modelled, with a frozen size and cost sweep rather than fictitious throughput or anonymity evidence.

For each retrieval candidate, specify authenticated canonical coverage and malicious-provider omission, false-empty, truncation, stale-range and equivocation cases, including client-visible incomplete status and bounded recovery. A passive privacy argument or matching provider replies do not prove completeness. Window/bucket variants must quantify privacy loss and total work over the same history. The transport model must include the wallet-to-validator path (including any aggregators), individual-proof fanout, batching, retries and outages. Each required service must expose its payer, cost, payment leakage, abuse limits and exit assumptions under specification 9.2.

Pass: return NOT_RULED_OUT only if at least one candidate satisfies all applicable frozen screening thresholds throughout the declared envelope, including required evidence and privacy assumptions. This permits T305; it is not G6 or implementation-level R1/R2 assurance. **Reject:** return STOP if no candidate qualifies, evidence is missing, a private retrieval provider's work is omitted, or simulated cryptography is presented as an implemented protocol. Out-of-envelope T305 measurements require a new reviewed T006 campaign before integration; they cannot widen

the passing envelope retroactively. T404, T602 and T604 retain full implementation and final acceptance ownership.

T101 — Field and extension-field arithmetic

Objective: Select and implement candidate base and extension fields whose use and security are then approved by T303.

Deliverables: canonical field encoding; add/subtract/multiply/invert; extension arithmetic; batch inversion; constant-time policy; reference and optimized implementations; exhaustive small-model and differential tests.

Pass: all identities, boundary values, invalid encodings, and cross-language vectors pass; proof-system reviewer confirms the field supports the soundness analysis. **Reject:** a 64-bit base field is presented as 128-bit soundness by itself.

T102 — Polynomial and NTT layer

Objective: Implement polynomial and NTT operations only after the selected commitment or proof profile requires them.

Deliverables: ring definition, modulus/root derivation, forward/inverse normalization, naive reference multiplication, optimized multiplication, and aliasing/memory rules.

Pass: optimized results equal the independent naive reference across vectors and randomized boundary cases. **Reject:** circular dependency, floating-point roots, undocumented padding, or silent reduction.

T103 — Samplers and commitment candidate

Objective: Implement one precisely cited commitment candidate without inventing a new formula.

Deliverables: construction citation and delta from it; exact parameters; construction-specific parameter/seed generation; canonical encodings; unbiased randomness sampling (including rejection sampling where required); commit/open API; note-field binding; deterministic test mode isolated from production entropy.

Hash-based and lattice-based candidates are eligible under specification 4.4; matrix generation is required only by constructions that use it. Freeze one commitment before the T305 authorisation comparison; changing it starts a separate campaign rather than confounding the authorisation results.

Pass: correctness and vectors pass and the artifact is ready for T104 analysis. **Reject:** byte modulo sampling, reused randomness, invertible-linear "binding" intuition, mismatched randomness length, or carry-dependent value encoding.

T104 — Commitment security gate

Objective: Decide whether the commitment candidate meets R1, R3, and R4.

Deliverables: quantum hiding/binding definitions and reduction; concrete parameter estimates and multi-target/lifetime loss; side-channel review; independent cryptographer report. For lattice candidates include reproducible lattice-estimator inputs; for hash candidates include the randomness/hiding argument, required hash properties, digest lengths and quantum cost models. An assumption inventory must also cover ML-KEM and any homomorphic retrieval scheme; removing a lattice commitment does not remove their assumptions.

Pass: composed security remains at least 128 post-quantum bits and all findings are resolved. **Reject:** collision-search testing is used as proof, the construction is only named "BDLOP-like," or a guessed value can be tested from public data.

T201 — SHAKE256 and domain hashing

Objective: Implement FIPS 202 SHAKE256 and the Quantum QH wrapper exactly.

Deliverables: official FIPS 202 KATs; incremental/one-shot behavior; Quantum vectors; tag registry enforcement; output-length binding; per-domain collision/preimage/second-preimage requirement; concrete quantum and multi-target output-length analysis; canonical error behavior.

Pass: official and Quantum vectors match in both implementations. **Reject:** SHA3-256 is silently substituted, tags are concatenated ambiguously, or XOF length is unbound or a 32-byte research vector is treated as proof that every collision-dependent consensus digest meets R3.

T202 — Hash-based authorization comparison

Objective: Compare exact FIPS 205 SLH-DSA-SHAKE-256f and SLH-DSA-SHAKE-256s with an independently specified TzEL-shaped one-time baseline and, if applicable to the wallet model, one exact NIST SP 800-208 stateful profile.

Deliverables: the signed T305 prior-art and reuse decision; exact construction, parameters, source/standard revisions, APIs, contexts, randomization and state-transition choices for every arm; official KATs where available; canonical key/signature encodings; authorization digest; key-erasure and side-channel rules; state-index allocation, crash, backup/restore, concurrency, exhaustion, recovery, and desynchronisation cases; malformed signature/state corpus; per-arm in-proof verification interface; FIPS 205 message-bound-signature analysis and any reviewed transformation; signed applicability decision for the NIST stateful arm before benchmark results are viewed.

T304/T305 own measured in-proof cost and complete signing-plus-proving latency. Both standardised stateless profiles require full KAT and security treatment; signature size is not a proxy for proof cost. A separately registered SP 800-230 exploratory arm must identify the draft revision and demonstrate how its 2^24-signatures-per-key limit covers every device, retry and backup restore. An inapplicable draft arm may be excluded only by a signed decision before its results are viewed; it cannot replace a required FIPS 205 arm.

Pass: both implementations match applicable KATs and shared vectors, reject malformed/non-canonical inputs and state misuse, and expose all retained arms to T305 under the same transaction relation. **Reject:** protocol IDs say SPHINCS+, TzEL code is copied without compatible permission and legal review, a classical fallback exists, signatures authorize a partial transaction, or stateful failure modes are omitted.

T203 — ML-KEM, KDF and authenticated encryption

Objective: Define recipient encryption building blocks with ML-KEM-1024.

Deliverables: FIPS 203 KATs and errata version; decapsulation-failure behavior; transcript-bound KDF; 256-bit AEAD profile; nonce and replay rules; key separation; receiver-anonymity/key-privacy game for note encryption; multi-user, chosen-key, chosen-ciphertext and cross-output analysis; scanning-tag disclosure budget; test vectors.

Pass: chosen-ciphertext, receiver-anonymity and misuse analysis covers the exact note-encryption composition. **Reject:** ML-KEM is treated as authentication, failure creates an oracle, nonce reuse is possible, or an invented Noise pattern name substitutes for a message specification, or the ciphertext identifies its recipient key.

T204 — Wallet key hierarchy and address profile

Objective: Freeze mnemonic import, child-key derivation, address payload, and storage protection.

Deliverables: exact BIP-39 behavior; 24-word generation from 256 bits of uniform entropy as the minimum compatibility profile; T001-derived multi-user/multi-target/lifetime source-entropy bound for the complete wallet master secret; if required, an independently generated, domain-separated supplemental secret or separately reviewed higher-entropy seed format with exact generation, combination, backup, recovery, and failure rules; lower-entropy legacy-import identification and visible security-downgrade warning; domain-separated post-quantum child derivation; spend/nullifier/encryption key separation; incoming-view, full-wallet-view, transaction-disclosure,

and auditor-scoped capability derivation; canonical scope encodings; canonical lower Base32 address vectors; memory-hard wallet-storage KDF; recovery tests; immutable note-creation versus active transaction-profile interpretation; historical signing-key retention and recovery; and the interface for T510 to validate any migration without changing old nullifiers or silently stranding unspent notes.

Pass: BIP-39 official vectors and Quantum derivation/address/disclosure vectors match both implementations; the complete generated master-secret profile meets the T001 entropy bound without double-counting derived output; legacy imports remain explicitly degraded and do not inherit the R3 claim; and each disclosure capability reveals no more than its stated scope. **Reject:** a 256-bit mnemonic is treated as automatic evidence for the composed R3 target, PBKDF2 output length is presented as added source entropy, legacy wallets silently inherit the 128-bit post-quantum claim, PBKDF2 parameters are called BIP-39 while using a different salt/normalization, one key crosses protocol roles, or a universal viewing backdoor is required.

T301 — Integer transaction relation and AIR

Objective: Encode authorization, input/output openings, encryption binding, ranges, and exact conservation.

Deliverables:

- formal relation before AIR translation;
- public-input and witness schemas;
- constraints for every input commitment opening and membership leaf;
- constraints for every output commitment opening;
- complete verification of each retained T202 authorisation profile over the same complete digest;
- 16-bit limb range checks and wide carry-constrained sums;
- fee and reward-type separation;
- AIR degree and trace-size report.

Required negative cases:

- replace an input value while keeping its commitment;
- spend a one-unit note while claiming a near-maximum input value;
- make both sides equal only modulo the proof field;
- cross a 16-bit carry boundary such as 65,535 + 1;
- omit or alter an output opening, fee, chain ID, or authorization field.

Pass: every negative case is unsatisfiable and reviewers trace each R4 clause to constraints. **Reject:** balance is only a field equality or an input opening is absent.

T302 — Membership, nullifier and state AIR

Objective: Prove note membership and correct unlinkable nullifier derivation against the exact state hashes.

Deliverables: depth and empty-root definitions; leaf/internal tags; membership-index constraints; nullifier relation; local uniqueness; anchor/public-state binding; digest length derived from T001/T201 for Merkle collision and second-preimage requirements; invalid-path and alternate-tree vectors.

The nullifier relation MUST implement specification Section 5.6.1: the opened note and bound secret, constrained leaf position, chain and immutable creation profile identify the instance. It must preserve spendability for duplicated commitments at different accepted positions and preserve one spent identity across anchors and transaction-version changes. Exact construction/security analysis remains a gate. Include duplicate notes across transactions, sender-chosen randomness, swapped positions, changed active versions, and the T403/T510 rollback/migration interface in the vector contract.

Pass: all paths and indices are bound, invalid paths fail, and no public field creates a direct note-to-nullifier link. The selected digest length MUST meet R3 after multi-target and lifetime

losses. **Reject:** path direction is unconstrained, empty roots differ across implementations, a 32-byte wrapper vector silently becomes the Merkle security parameter, or tree depth exhaustion is ignored.

T303 — Fiat–Shamir, FRI and zero-knowledge profile

Objective: Select the complete transparent proof profile.

Deliverables: base/extension fields; transcript schedule; challenge sampling; AIR composition; FRI/DEEP-FRI parameters; query/grinding/batching; zero-knowledge masks; classical/QROM soundness calculations; multi-proof loss; denial-of-service limits.

Pass: independent proof-system review shows at least 128 post-quantum bits after composition and no witness leakage under the stated model. **Reject:** the arithmetic "queries × bits" is used as soundness, target is 2^-100, masking is absent, or a generic STARK is assumed zero knowledge.

T304 — Prover, verifier and aggregation

Objective: Implement the T301–T303 profile without accepting ambiguous proofs.

Deliverables: independent prover/verifier interfaces; streaming/resource limits; transcript vectors; malformed-proof corpus; deterministic failure codes; individual and, if selected, aggregate wire mode under the signed T402 proof policy; separate effects and proof-carriage commitments; binding of an aggregate to ordered effects and public inputs; an explicit state-context interface whose exact Quantum relation is supplied by T503–T505; differential/fuzz tests; measured proof generation and verification.

Before T503–T505, any aggregate state relation is a labelled feasibility surrogate, not a claim of canonical DAG-state validity. T305 measures the complete private-transaction relation and the selected aggregation mechanism; the exact consensus relation is subsequently revalidated by T505/T604.

Pass: invalid witnesses and mutated proofs fail; both implementations interoperate; aggregate mode omits replaced individual proofs; performance is fed into T305. **Reject:** verifier panics, work is allocated before size checks, both proof modes are redundantly transmitted, or tests are claimed as soundness proof.

T306 — Proof-system accumulation capability

Objective: Determine whether the selected transparent proof system offers a generic recursion, folding, or accumulation capability inside R3 and the frozen R9 profile. This task does not own Quantum's exact consensus/state relation.

Deliverables: a capability-specific base/step surrogate or accumulator relation; verifier-in-circuit, folding, or equivalent implementation; transcript and field compatibility; composition theorem and concrete post- quantum loss over the intended lifetime; proof size; verifier and prover time/memory; depth and cadence sweep; malformed inner/outer proof corpus; two interoperable implementations; prover concentration/outage model; alternative or degraded path; and a signed CAPABLE/NOT_CAPABLE interface decision for T505. GHOSTDAG canonicality, exact transaction state, rewards, fees, burns, issuance, and reorganisation semantics remain owned by T503–T505.

Pass: the evidence supports either a usable generic accumulation capability or a reviewed rejection with an alternative T505 bootstrap profile. **Reject:** "STARKs are recursive" replaces an implemented capability; the in-circuit verifier or equivalent accumulator is omitted; composition loss is unquantified; the task prematurely freezes the Quantum relation; proof soundness is called data availability; or the only path requires one trusted prover.

T305 — Cryptographic transaction feasibility gate

Objective: Decide whether the complete private transaction is practical before wallet, node, or GHOSTDAG integration consumes implementation effort.

Experiment protocol: The non-normative T305 research protocol records the prior-work boundary, pre-registration, independent-implementation method, evidence package, and stop/go form. It may strengthen experimental detail but cannot weaken this task or the protocol specification.

Required prototype: exactly two inputs and two outputs with input openings, Merkle membership, nullifier derivation, output openings, encrypted-note binding, complete in-proof authorisation for each retained T202 arm, 64-bit range constraints, carry-safe conservation, public fee, zero-knowledge masking, and the selected individual and aggregate proof path. Only authorisation and its required state may differ between arms.

Deliverables: two interoperable implementations; pinned desktop and constrained-client hardware; single-wallet p50/p95/p99 proving latency; peak and steady memory; verifier time and memory; proof bytes; consensus-transaction bytes; external encrypted-note payload bytes; total generated bytes per payment; payload-provider and retrieval bandwidth; payload availability and retention assumptions; parallel-prover scaling; aggregate latency, size, and amortised verification; isolated authorisation contribution; state-management failure results; pre-registered material-benefit criteria; clearly labelled reproduced versus author-reported prior-work measurements; raw artifacts; and a signed GO/ADAPT/REPLICATE/STOP decision.

Break down trace rows/cycles, time and memory for authorisation, commitments, Merkle paths, nullifiers, ML-KEM/KDF/AEAD binding, ranges/balance and proof overhead; report shared work without double counting. Hash-call or permutation counts are analytical estimates, not measured proving costs. Include the specification 4.5 carriage path, per-hop bytes/fanout, aggregator resources and failure behaviour; amortised proof bytes are not complete transaction bytes.

Pass: the run reuses the canonical T402 codec and staged digest vectors, its actual output/packet sizes remain within a NOT_RULED_OUT T006 envelope, and it satisfies every T004 method and every T005 threshold frozen before it began, without mock signatures, omitted relations, disabled zero knowledge, retained individual proofs in aggregate mode, unreported preprocessing, hidden digest-only payload/provider traffic, or post-result benefit criteria. Retaining the stateless profile additionally requires a material pre-registered benefit over every qualifying stateful arm. The 256f incumbent and 256s comparator are evaluated separately; retaining 256s requires a versioned profile decision before integration. **Reject:** no arm meets the frozen client, verifier, wire, and aggregate budgets. A reject returns the signature, commitment, relation, proof profile, or explicit system requirements to research; it MUST NOT be deferred to node optimisation.

T401 — Note creation and recipient encryption

Objective: Create a note that the recipient can discover and later spend without publishing recipient or amount.

Deliverables: note plaintext schema; commitment randomness; encryption witness; diversified recipient data; authenticated binding to the pre-encryption context and output index from specification Section 6.1; immutable committed note-creation profile; receiver-anonymity/key-privacy analysis linked to T001/T203; sender/recipient test vectors; failure and deletion behavior.

An output-creation API MUST receive all data needed by the final note commitment, including spend-authorization and nullifier-key digests. A view public key alone is insufficient.

Pass: intended recipients recover exactly the committed note, non-recipients cannot distinguish which eligible recipient key was targeted beyond the approved disclosure budget, and replay/swapping fails. **Reject:** encryption and commitment can describe different notes, or a ciphertext/scanning tag identifies the recipient.

T402 — Transaction codec, builder and authorization hash

Objective: Construct and parse the exact public transaction while keeping private witnesses off-ledger.

Deliverables: canonical effects codec; maximum counts/sizes; acyclic pre-encryption-context/effects-ID/authorisation-digest schedule from specification Section 6.1; builder state machine; duplicate handling; expiry/anchor/fee-context/authorisation-profile/proof-policy binding; pre-signing resource-weight contract; opaque bounded envelope interface for T304; stage-by-stage vectors and negative parser corpus. The envelope's concrete proof encoding is owned by T304, not guessed here.

Pass: round trips are byte-stable, alternate encodings fail, and every mutable semantic field changes the authorization digest. **Reject:** parser fallbacks, unbounded allocation, signature malleability, or unknown fields silently ignored.

Proof re-randomisation or policy-permitted aggregation must leave effects ID and authorisation unchanged; effect mutation must invalidate the signature, and an envelope bound to different effects must fail proof/block validation. The same codec and builder must be used by the feasibility experiment and integrated wallet. No transaction may require its own ciphertext, proof, or concrete aggregate identifier before those artifacts can be produced.

T403 — Commitment/nullifier state and atomic application

Objective: Implement the single owner for note-tree, nullifier-set, and state-root transitions.

Deliverables: atomic apply/revert; deterministic roots; duplicate nullifier checks; snapshot/recovery format; reorg journal; concurrency model; invariants and crash tests; role-specific current-state representations; and, if witness-carrying state is proposed, exact membership/non-membership, append, update, batch, rollback and anti-replay witnesses plus wire/provider costs.

Compare explicit-set and any witness-carrying storage over the declared lifetime at the same accepted-input rate. State bytes per entry, indexes, replication, witness maintenance, provider retention and offline-wallet recovery separately. A compact root does not erase global spent-state or availability obligations. Expiry/migration proposals require R7.10/T510 review, with no silent loss of old funds or renewed spending of consumed notes.

State indexes and wallet records must retain authenticated note-instance identity. Revert removes orphaned positions and dependent spends before replay; duplicate commitments at different positions cannot overwrite one wallet entry. The T302 nullifier-domain contract persists across active-version changes; T510 later validates the exact upgrade/migration policy against it.

Transaction verification MUST take an explicit immutable state/anchor context. A stateless Transaction::verify() is insufficient for global nullifier and anchor checks.

Pass: crashes and reorgs recover to the same root, two spends cannot both commit, no non-owner bypasses state validation, and an executing validator never silently queries a trusted state service. **Reject:** check-then-write races, partial state commits, hidden empty-state fallback, stale witness replay, or provider work omitted from the R9/R10 role report.

T404 — Wallet scanning and spending

Objective: Build deterministic private wallet behavior over T401–T403.

Deliverables: scan/decrypt pipeline; authenticated note-instance and nullifier tracking, never commitment-only indexing; spend selection; the specification's Section 6.1 encryption/sign/proof construction flow; post-reorg witness refresh and rollback; historical-note recovery; scoped disclosure export and receipt verification; disclosure limitations; secure deletion; user-facing finality and fee semantics; and, if the selected state profile requires wallet witnesses, online updates, offline recovery from multiple untrusted providers, anti-replay verification, and witness-query privacy evidence.

Pass: restore-from-seed and rescan reproduce the same wallet state through reorgs; spent notes cannot be selected; no network identity is derived from wallet keys. **Reject:** missing data is zero-filled, failed decryption is treated as an empty note, or sensitive logs are emitted.

T405 — Selective disclosure and bounded payment policies

Objective: Add useful settlement controls without introducing a general virtual machine, a universal viewing key, or an unbounded proof relation.

Deliverables: incoming-view, full-wallet-view, transaction-specific, and auditor-scoped disclosure flows; canonical scope and receipt encodings; least-privilege and unlinkability tests; documented non-revocation of copied history; a finite registry of accepted payment policies; exact semantics and costs for each enabled timelock, hashlock, threshold/multisignature, escrow, recurring-authorisation, atomic-swap, or conditional-release policy; updated AIR/proof profile, vectors, and resource report for every enabled policy.

Private asset issuance and cross-chain adapters are separate post-core profiles. Any proposed adapter MUST publish its external consensus, custody, MPC/multisignature, oracle, light-client, finality, privacy, and upgrade trust assumptions; it does not inherit Quantum's base-layer assurance.

Pass: disclosure never exceeds its canonical scope, policy execution is deterministic and bounded across reorgs/timeouts, and every enabled policy preserves transaction privacy, authorisation, supply, and T305 performance budgets. **Reject:** a global backdoor key exists, copied disclosure is called revocable, unknown policy bytecode falls back to execution, a policy can mint or bypass authorisation, or an adapter is described with stronger properties than its own evidence supports.

T501 — Versioned GHOSTDAG consensus profile

Objective: Turn the cited GHOSTDAG research into one deterministic, versioned protocol profile.

Deliverables: parent validation; anticone and blue/red rules; selected parent; merge ordering; blue work; finality/pruning assumptions; block limits; reference model; exhaustive small-DAG and adversarial vectors.

Pass: independent implementations produce identical classification, ordering, score/work, and finality across all vectors. **Reject:** "highest blue score wins," a simple topological sort, or upstream behavior without a pinned version is treated as the specification.

T509 — PoW hardware contestability study

Objective: Compare proof-of-work candidates before T502 selects one, using measured contestability rather than an ASIC-resistance label.

Deliverables: exact candidate algorithms and implementations; verification cost; CPU/GPU performance; plausible specialised-hardware efficiency, memory, chip-area and energy models; manufacturer/fabrication and supplier concentration; open miner availability; cloud-rental and botnet exposure; block-rate/propagation interaction; reproducible measurements; independent hardware and mining-incentive review; and results against the G12 metrics, scenario definitions, pass/fail thresholds, and STOP conditions frozen by T005.

Pass: at least one candidate meets every pre-registered applicable T005 threshold and the signed comparison selects it with residual hardware, energy, capital, supply-chain, and rental risks stated. **Reject:** no candidate meets the frozen thresholds but one is selected anyway; T502 selects an algorithm first; "memory-hard," "commodity-friendly," or "ASIC-resistant" substitutes for measurements; verification or network cost is omitted; or author-reported estimates are presented as reproduced results. If no candidate passes, the project stops or creates a new, independently reviewed campaign without reclassifying the failed evidence.

T502 — Header, selected PoW and deterministic DAA

Objective: Specify and implement canonical headers, PoW, and expected difficulty.

Deliverables: exact header bytes and size; the T509-selected PoW identifier and parameters; block/PoW domains; integer target conversion; DAA equations, window, clamps and overflow rules; timestamp rules; genesis-independent vectors; time-warp simulations.

Required negative cases:

- miner supplies an easy target;
- nodes have different local clocks;
- timestamps are reordered or clamped;
- maximum/minimum targets and arithmetic boundaries;
- semantically equal headers use alternate bytes.

Pass: validators recompute and enforce the same expected target without floating point or local-time dependence. **Reject:** the header target is only used to check its own PoW.

T510 — Post-quantum upgrade governance and activation

Objective: Specify and test the mechanism that authorises, activates, recovers, and, where permitted, rejects protocol upgrades without a classical fallback or ambiguous activation state.

Deliverables: exact authorisation principals and post-quantum algorithms; canonical proposal, vote/signature, version, chain, and activation encodings; quorum or other decision rule; timelocks and activation thresholds; key generation, storage, rotation, recovery, revocation, compromise, and succession procedures; normal and emergency state machines; replay, downgrade, equivocation, and persistent-split analysis; principal-concentration, bribery, collusion, strategic-abstention, veto-power, and client-distribution-control analysis; consensus-parameter registry classifying immutable historical invariants, normal-upgrade-only protected parameters, and emergency-changeable parameters; an explicit prohibition on emergency changes to the cap semantic, issuance schedule, fee allocation, fee-pool liabilities, and value-conservation rules; boundary between deterministic protocol activation and voluntary social adoption; rollback boundary; audit trail; canonical positive and negative vectors; two interoperable implementations; and named governance owner plus independent cryptography, consensus, implementation-security, Product Owner, and legal review.

Pass: both implementations derive the same authorised upgrade and activation state for every vector; unauthorised, replayed, downgraded, equivocating, and out-of-window proposals fail closed; compromised-key and emergency paths follow one predeclared rule; the capture and adoption boundaries are explicit; accrued issuance and fee liabilities remain invariant; protected monetary parameters can change only through a new T506 campaign and normal versioned activation; and no hidden unilateral or classical fallback exists. This pass does not prove governance decentralisation or exclude capture by a formally authorised quorum. **Reject:** governance is deferred to unspecified social consensus, cryptographic activation is presented as proof of governance decentralisation, one undisclosed key can activate code, key rotation or compromise recovery is absent, emergency procedure bypasses the authenticated state machine, an emergency or ordinary proposal silently rewrites accrued issuance or fee liabilities, an emergency proposal changes protected monetary rules, or different honest nodes can activate different versions from the same inputs.

T510 additionally owns historical-note continuity under R7.10: immutable creation profiles, stable spent-nullifier identity, historical signature verification, retained key recovery, and any explicit consume-old/create-new migration. Publish activation, offline restore, repeated migration, profile retirement and rollback vectors against T204/T302/T403. A spent note must not become spendable after an upgrade and an unspent note must have the declared spending or migration path. These are approval prerequisites, not a signature downgrade or an implicit historical-format fallback.

T503 — Canonical DAG-to-state ordering

Objective: Prove and implement one deterministic state transition order for parallel blocks.

Deliverables: ordered merge-set algorithm; transaction ordering; duplicate and concurrent-nullifier rule; exact public ordering inputs and candidate- selection boundary; pre/post-state commitments; reorg semantics; model checker or equivalent state exploration; adversarial DAG vectors; interface for T508 to compare ordering/incentive candidates without changing canonical state semantics silently.

Specify the exact header/checkpoint execution context and whether its own body is included or deferred. Distinguish malformed/cryptographically invalid blocks from contextually rejected transactions. Honest parallel conflicts skip only the losing transaction, with no output, nullifier or monetary effect. Instantiate or explicitly replace specification Section 7.3.1's symbolic sibling/merge/reorg example, then generate independent byte/root vectors. T503 freezes an accounting interface; T504 supplies reward/fee/controller semantics and revalidates this model, so T503 does not require future monetary results. A different canonical history can select a different spend only after complete rollback; header roots are never compared across execution contexts.

Pass: all implementations converge on accepted transactions and state root under permutations, delays, conflicts, and reorgs. **Reject:** parallel blocks mutate shared state concurrently or conflict resolution depends on arrival order.

T506 — Monetary-security decision

Objective: Select the exact monetary-policy semantic, integer issuance schedule, fee/burn policy, and long-run proof-of-work security model before T504.

Required comparators: the current 21,000,000 QTM lifetime gross-issuance cap with a smoothed finite schedule and long-run fee funding; within that cap, a static resource-fee baseline and the bounded dynamic-fee/reward-window candidate defined by R12.7–R12.13; fixed-absolute and fixed-percentage tail emission; an outstanding-supply cap with explicit burn and reissuance; and any other non- oracle candidate proposed with exact accounting.

Deliverables: an immutable pre-registration identifier and timestamp before candidate results are interpreted; canonical DAG-time/score schedules; rounding and reorg behavior; monetary-policy and supply trajectories; fee and burn rules; miner-revenue distributions; for the dynamic candidate, canonical transaction weight, fee/reward epoch, fee-rate scale, resource and security controller functions, finalisation lag, minimum/maximum rates, per-window change bounds, exact zero-weight transition, epoch-admission/grace and expiry rules, optional-priority policy, exact resource/security/priority charge-class accounting, exact miner-eligible/burn/non-miner-output destination accounting, the allocation matrix between those views, sustainable miner budget versus claimed-reward accounting, fee-pool liability and maturity buckets, matured- claimable payout capacity, maximum retention, burn, rounding, remainder and reorg rules, exact monetary/weight/rate/accumulator widths and operand maxima, checked arithmetic and quotient/remainder ceil_div vectors, underfunded-state rule, and direct payout, partial/full delayed pool, partial payout plus burn, and separate resource-fee and security-fee allocation variants; stated price, fee, hashrate, hardware, energy, entry/exit, attack-rental and pool-concentration assumptions; sensitivity and adversarial analysis; distributional/user-cost analysis; exact scenario sets, metrics, pass/fail thresholds, and STOP conditions for fee and miner-revenue cases, price and energy shocks, miner exit/hashrate response, security-expenditure or attack-cost proxies, and concentration; exact public wording for every candidate; signed Product Owner decision; independent consensus, monetary- economics, and legal reports.

Pass: one policy meets every pre-registered applicable threshold, is selected on an exact version before T504, its accounting and public claim agree, and no unresolved high-severity economic or legal finding remains. A selected dynamic controller also remains stable inside every applicable demand and adversarial scenario, meets both miner-revenue and user-cost thresholds, and never treats a target as guaranteed revenue. The result MAY retain the current cap. **Reject:** criteria are selected after candidate results are viewed; no candidate meets the frozen STOP conditions but one is selected anyway; engineering selects monetary policy; positive subsidy is called adequate security; a hard cap is said to prove a fee market; zero demand is treated as fee revenue; an external

price/cost oracle or current mempool changes consensus fees; a rate or zero-weight transition is unbounded; total accepted fees or burned/non-miner allocations are counted as sustainable miner revenue; an amount already designated for future burn is labelled miner- eligible; payout capacity includes an immature or non-claimable balance; arithmetic wraps, saturates, underflows, or uses an unspecified width; an unpaid fee disappears instead of remaining in the pool or following an explicit burn rule; foregone subsidy is counted as burn; more usage is assumed to create burn; or burn/reissuance keeps a lifetime gross-issuance claim unchanged.

T504 — Rewards, selected supply policy and genesis

Objective: Implement the exact T506 value-creation decision and immutable network start.

Deliverables: distinct reward transaction; DAA-score or other exact DAG issuance index; blue/red/stale eligibility; selected monetary-policy semantic; integer schedule and rounding; selected fee-epoch/controller state and reward- window allocation if applicable; accepted resource/security/priority fee accounting; miner-eligible, burn, and non-miner-output destination accounting; fee-pool liability, matured-claimable balance, retention, rounding, remainder and reorg accounting; exact non-negative integer widths, checked arithmetic and ceil_div semantics; authorised/claimed/foregone subsidy accounting; burned-existing-value accounting; gross issuance and outstanding supply; proof of the selected supply-policy invariant and prevention of unauthorised issuance; reward maturity and reorg reversal; private reward-output handling; canonical genesis bytes and vectors, including zero initial fee pool, cumulative issuance equal to genesis claimed subsidy, zero genesis subsidy under the retained no-premine baseline, and canonical initial controller epoch and rates.

Pass: exhaustive/bounded-model checks and analytical proof show the selected monetary-policy invariant holds across reorganisations, fees never count as new issuance, every fee enters exactly one canonical acceptance interval on the selected history, unpaid unburned miner-eligible fees remain in outstanding supply, only miner-eligible new fees enter the fee-derived sustainable miner budget, fee-derived payout capacity uses only matured claimable pool value, foregone subsidy is never a burn, burned value is not reissued unless T506 explicitly changes the monetary claim, and all nodes derive identical rewards. **Reject:** linear height is used without a DAG definition; a fee, subsidy, burn or foregone subsidy is negative or counted twice; a fee assignment survives a reorg and is also assigned on the new history; no reward transaction exists; public monetary wording differs from T506; timestamp units differ; genesis monetary/controller state is implicit; or genesis hashes with another function.

T505 — Consensus-bound state validity and bootstrap

Objective: Select and implement the exact consensus-bound state-validity and checkpoint/bootstrap profile after the T306 capability decision.

Deliverables: canonical checkpoint/finality identity; exact relation or authenticated replay boundary; protocol version; canonical GHOSTDAG ordering; ordered accepted transaction set; note, nullifier, and policy pre/post-state; rewards, charge-class and destination-split fees, fee-pool liability and maturity state, fee-epoch/controller state, burns, issuance, and the selected monetary-policy state; reorg and pruning behavior; current-state/snapshot commitments; T605 data boundary; two interoperable implementations. If T306 selects a generic accumulation capability, instantiate and prove the exact Quantum relation here with its fallback. If T306 rejects it, specify the alternative without a succinct-proof claim.

Pass: a new validator derives the same canonical current state using the selected profile and independently validates all subsequent blocks, inside R3, R9, and R10. **Reject:** a proof alone is called bootstrap; canonicality or T504 monetary state or T405 policy semantics are omitted; multiple orders satisfy one checkpoint; current data or a trusted snapshot signer is assumed; or a rejected recursion profile is silently retained.

T507 — Miner-template and pooled-mining profile

Objective: Make miner-controlled transaction selection usable with pooled proof of work without conflating the mining device, Job Declarator, pool, and payout roles.

Deliverables: role and message protocol; custom-template commitment; acknowledged-job/share binding; independent valid-block publication; private reward/payout handling; pool-selected baseline; miner-custom-job profile; non-custodial pooled-mining comparator; latency, bandwidth, variance, privacy, payout and revenue-overhead measurements; cross-implementation vectors.

Pass: a miner-controlled full node constructs its transaction set, receives pooled share accounting bound to that job, and can publish a valid found block, with all overhead below T005 thresholds. **Reject:** the ASIC/mining device is misidentified as the template owner; the pool may replace an acknowledged job; reward privacy bypasses R1; payout requires custody without disclosure; or market adoption is claimed from protocol permission.

T508 — Ordering, censorship and incentive gate

Objective: Select transaction-candidate and ordering rules only after quantifying extraction, censorship, duplicate, reorganisation, and pool- formation incentives under Quantum's hidden and public fields.

Deliverables: exact candidate rules; public-input inventory; fee and fee- density strategies; adversarial validation of the exact T506-frozen fee and finalised-window allocation, including its direct, delayed-pool, burn, and resource/security/priority charge and miner/burn/non-miner destination choices; attempts to reclassify or time fees between those destinations or manipulate resource utilisation, accepted weight, revenue gaps, fee epochs, rate changes, and the underfunded signal through omission, self-fees or coordinated pools; conflicting-nullifier selection; transaction-identifier and timing grinding; duplicate inclusion across parallel blocks; fee sniping; selective/blanket censorship; inclusion-delay metrics; reorg and pool-formation strategies; simulator/model; reproducible attack experiments; independent consensus and mining-incentive review.

Pass: the frozen monetary rule and one ordering rule satisfy pre-registered T005/T506 thresholds and preserve T503 determinism without an unsupported MEV/censorship claim. **Reject:** the monetary rule is modified after its T506 freeze; privacy is said to eliminate extraction; public fees are called mechanical ordering; a fee bucket or tie-breaker is selected without grinding/inclusion analysis; or a generic GHOSTDAG result is transferred without matching Quantum's exact rules.

T508 is a falsification gate, not an in-place monetary-policy editor. If it rejects the frozen fee, pool, burn, or controller rule, a new versioned T506 campaign must pre-register and select the replacement. T504 and T507 must then be reimplemented or revalidated as applicable, and the dependent T508 evidence must be rebuilt; evidence for the rejected campaign cannot be carried forward.

T601 — Post-quantum authenticated P2P transport

Objective: Define link confidentiality/authentication without classical fallback or wallet identity leakage.

Deliverables: exact handshake message sequence; ML-KEM/SLH-DSA transcript binding; KDF/AEAD keys; replay/downgrade/key-compromise analysis; peer identity lifecycle; vectors and state-machine fuzzing.

Pass: independent cryptographic review approves the exact composition and implementations interoperate. **Reject:** "post-quantum Noise" is used without an actual reviewed pattern, ML-KEM is mistaken for authentication, or failed PQ negotiation falls back to X25519.

T602 — Network anonymity protocol

Objective: Meet R2 against the T001 observer rather than merely encrypt links.

Deliverables: routing and cover-traffic protocol; peer/route selection; batch/delay distributions; active tagging, eclipse, Sybil and intersection analysis; simulator; reproducible attack experiments; utility/privacy tradeoff report; mix/relay cost, payer, payment-metadata leakage, abuse controls and provider-exit/outage scenarios under specification 9.2.

Metrics: source-classification advantage, anonymity-set distribution, entropy, precision/recall for origin inference, latency, bandwidth amplification, failure behavior, and confidence intervals.

Pass: thresholds defined in T001 are met under all required scenarios and an independent network-privacy review passes. **Reject:** only ciphertext byte uniformity, payload correlation, or Dandelion++ deployment is offered as proof of full anonymity.

T605 — Current-data availability, snapshot and recovery

Objective: Specify and test complete current-data delivery for executing validators plus bounded, no-trusted-sole-provider bootstrap and recovery.

Deliverables: role/data matrix; complete current block/proof/state-data protocol; consensus commitments; authenticated snapshot and T505 checkpoint integration; any erasure code, availability commitment and sampling profile; reconstruction threshold; repair/re-replication; retention and incentive model; wallet-recovery data; selective-withholding, eclipse, corrupted-provider and correlated-loss experiments; separation of public ledger data from R2 network- origin metadata.

Apply specification 9.2's payer, cost, abuse and provider-exit contract to state/recovery services. With T404, test authenticated retrieval coverage and false-empty/stale responses, including offline-wallet witness recovery; a valid returned note or agreement between providers does not prove that all relevant notes were returned.

Pass: executing validators reject unavailable current bodies, a new node and offline wallet recover through multiple untrusted providers inside T005, and independent implementations reconstruct identical protocol-required data. **Reject:** sampling substitutes for current executing data; a designated foundation/company/signer is required; storage proofs or erasure coding are assumed sufficient; no repair or eclipse path exists; or archive distribution is called a privacy guarantee.

T603 — Full node, pruning and recovery

Objective: Integrate consensus, proof verification, atomic state, T505 bootstrap, T605 current/recovery data, transport, storage, and resource admission inside the frozen executing-validator profile.

Deliverables: validation pipeline with cheap checks first; bounded queues; peer resource accounting; complete current-body handling; selected state or transition-witness validation; state persistence; pruning/snapshot/archive model; T505/T605 bootstrap and recovery; reorg handling; role-specific resources; observability without sensitive data; fault-injection results.

Pass: state roots remain identical under crash, restart, overload, reorder, and recovery, no malformed input causes unbounded work, and the executing node stays inside T005. **Reject:** hidden fallback state, partial or sample-only current validation, archive size presented as pruned-node size, provider work omitted, or snapshot trust is undocumented.

T604 — End-to-end system validation

Objective: Demonstrate R1–R12 together, not in isolated microbenchmarks.

Required run:

- geographically distributed topology;
- private note creation, scanning, spending, proofs, propagation and state application enabled;
- selective-disclosure flows and every payment policy enabled in the tested release profile;
- at least 1,000 accepted layer-1 tx/s for 24 hours;

- conflict, delay, peer loss, malformed traffic, reorg, restart and recovery phases;
- miner-controlled custom templates, the selected pooled-mining comparator, T508 ordering/censorship scenarios, T505 bootstrap and T605 recovery enabled;
- raw proof-size, bandwidth, storage, CPU, memory, latency, rejection and anonymity metrics for every declared role.

Pass: all functional invariants remain true; G7 and G10 pass together; G11, G12 and G13 evidence matches the tested revision; anonymity thresholds pass; and another team reproduces the report. **Reject:** benchmark uses mock proofs, prevalidated transactions, disabled privacy, submitted TPS, a single process, unpublished hardware, sample-only current data, or unreported provider work.

T701 — Composition review and release decision

Objective: Decide whether the combined system—not only its parts—meets the five non-negotiable product requirements and R1–R12.

Deliverables: requirements-to-evidence manifest; independent cryptographic, proof, consensus, data-availability/storage, network-privacy, mining/pool-incentive, monetary-economics, implementation and performance reports; T510 post-quantum upgrade-governance and activation report; finding register; per-capability claim register; residual-risk statement; Product Owner and required legal review; signed go/no-go record.

Pass for testnet: all blocking design gates pass, two implementations interoperate, and no unresolved critical/high issue permits unauthorised issuance, unauthorised spend or upgrade, privacy break, consensus divergence, or target bypass.

Pass for production: testnet evidence remains valid over the approved stability period, external audits are closed, T510 governance/upgrades are post-quantum, replay-safe, compromise-aware, and activation-deterministic, operational recovery is proven, and public claims match evidence.

Reject: an author or AI self-certifies the system, a component review is presented as composition review, or any hard requirement is deferred until after launch.

7. Mandatory adversarial scenarios

Every relevant implementation must preserve a permanent regression test for:

ID	Scenario	Required outcome
A001	Small real input opened with a larger private input value	Proof unsatisfiable
A002	Input/output balance equal only modulo the proof field	Proof unsatisfiable
A003	65,535 + 1 crosses an unconstrained limb carry	Correct integer result only
A004	Output commitment and encrypted output describe different notes	Rejected
A005	Authorization omits fee, fee context, output, chain, anchor, authorisation profile, or proof policy	Rejected
A006	Same nullifier in one transaction, block, parallel block, or reorg	At most canonical first spend accepted
A007	Header declares an easier target than the DAA computes	Block rejected
A008	Honest nodes use different local clocks on the same DAG	Identical consensus result

ID	Scenario	Required outcome
A009	Equivalent DAG delivered in different arrival orders	Identical ordering and state root
A010	Oversized vectors/proofs or unknown version	Rejected before expensive work
A011	Failed PQ negotiation requests classical fallback	Connection rejected
A012	Origin observer combines timing, volume and route metadata	Must remain below T001 attack threshold
A013	Snapshot interrupted or maliciously altered	Recovery rejects or reaches canonical root
A014	Reward rounding, reorganisation, fee replay, or schedule-boundary case	The selected monetary-policy invariant holds; fees are never counted as new issuance, lost from the pool, or claimed twice
A015	Observer compares note ciphertexts or scanning tags against candidate recipient keys	Recipient inference remains below the T001 threshold
A016	Disclosure capability is replayed outside its account, counterparty, transaction, or time scope	No additional note or wallet history is revealed
A017	Unknown payment-policy identifier, unbounded policy input, timeout/reorg edge, or adapter downgrade	Fail closed; deterministic bounded state; no inherited bridge claim
A018	Valid state/transition witness replayed against another parent, order, chain, version, or reorg branch	Rejected; canonical state unchanged
A019	Accumulated proof is valid but current block body or required state data is withheld	Executing validator does not accept; proof is not availability
A020	Prover delays or selectively proves one consensus-valid history	Observable degraded/alternative path follows the selected T505 profile
A021	Snapshot or archive provider equivocates, withholds selected segments, or eclipses recovery peers	Recovery rejects or reconstructs canonical data through independent providers
A022	Pool replaces a miner-acknowledged template or blocks direct publication	Share/job mismatch rejected; valid miner publication path remains available
A023	Producer grinds transaction IDs/timing, selects conflicting nullifiers, self-pays fees, duplicates transactions, or fee-snipes	T508 thresholds and canonical T503 state remain satisfied
A024	Candidate PoW has an undisclosed specialised-hardware or rental advantage	T509 comparison reopens; no ASIC-resistance or contestability claim
A025	Foregone subsidy is counted as burned value or later issuance headroom	Rejected; gross issuance and outstanding supply unchanged except by the exact valid transition

ID	Scenario	Required outcome
A026	Burned existing value is reissued while retaining a lifetime gross-issuance-cap claim	Rejected unless a separately approved cap semantic and public statement replace the claim
A027	G7 reaches target only by raising hardware or moving required work to an unreported provider	G10 fails; release fails
A028	Upgrade proposal is unauthorised, replayed, downgraded, equivocated, out of window, or signed by a compromised/revoked key	T510 rejects it or follows the one predeclared recovery path; honest nodes do not split
A029	A G12 or G13 metric, scenario, threshold, or STOP condition is changed after candidate results are viewed	Evidence is invalid for that campaign; the failed result is not reclassified
A030	A formally authorised quorum colludes, is bribed, abstains strategically, exercises a veto, or controls client distribution	T510 records the outcome inside its capture model; cryptographic validity is not presented as governance decentralisation, and protocol activation remains distinct from voluntary adoption
A031	A miner or pool omits transactions, self-pays fees, changes accepted weight, or times publication to move the next dynamic fee rate	Only prior-finalised canonical inputs affect the bounded transition; no extra issuance occurs; T508/G13 thresholds still pass or the candidate fails
A032	Accepted transaction weight falls to zero while the miner-revenue objective remains positive	The exact zero-weight transition is deterministic and bounded; fee revenue remains zero, the result becomes UNDERFUNDED when its frozen condition is met, and no divide-by-zero, cap bypass, or adequate-security claim occurs
A033	A signed transaction crosses a fee-epoch boundary because of honest delay or strategic withholding	The selected expiry and epoch-admission/grace rule yields one deterministic required fee; nodes do not reinterpret the signature or disagree on acceptance
A034	A fee held for delayed payout is reorged, omitted from outstanding supply, burned implicitly, or paid on two branches	Fee-pool state rolls back and advances atomically; the fee is paid or explicitly burned at most once, otherwise remains a liability, and never increments gross issuance
A035	Burned or non-miner-allocated fees are included in sustainable miner budget, or an immature pool amount is included in payout capacity	Rejected; the destination split balances exactly, sustainable miner budget uses only newly miner-eligible fees, and payout capacity uses only matured claimable pool value

ID	Scenario	Required outcome
A036	Maximum transaction weight/rate, interval accumulation, multiplication, addition, or ceil_div reaches an integer boundary	Exact-width checked arithmetic matches mathematical-integer vectors; overflow, underflow, wrap, saturation, negative values, and zero divisors are rejected
A037	A normal or emergency upgrade rewrites accrued issuance, erases/reclassifies a fee-pool liability, or changes protected monetary rules without a new T506 campaign	T510 rejects the transition; an emergency cannot change the cap semantic, issuance schedule, fee allocation, or value-conservation equations
A038	Genesis uses a non-zero implicit pool, inconsistent cumulative issuance, or implementation-local controller epoch/rates	Genesis is rejected unless the pool is zero, issuance equals the explicitly authorised genesis subsidy, and epoch/rates match canonical constants
A039	A sender repeats a note/commitment across transactions at distinct accepted leaf positions	Distinct authenticated instances remain independently spendable; wallet records do not collapse them
A040	Active transaction version changes or a migration is replayed for an old note	Spent identity remains consumed; an unspent historical note retains its declared spending/migration path
A041	Nullifier uses a swapped leaf position or a position from an orphaned history	Membership and instance checks reject it; rollback/replay matches clean evaluation
A042	Encryption requires the final txid, or signing requires the proof or concrete aggregate identifier produced afterwards	T402 construction is rejected as cyclic; stage vectors use only prior outputs
A043	Proof is re-randomised/aggregated, or effects are swapped under its envelope	Permitted representation changes preserve effects ID and authorisation; changed effects or invalid envelope binding fail
A044	Parallel valid bodies conflict or a sibling root is checked in the merge context	Only canonical first spend has state/fee effects; roots are checked in their declared context; rollback restores all effects
A045	Actual T305 payload/proof sizes exceed the T006 screening envelope	Integration stops pending a new reviewed campaign; no retroactive threshold change
A046	SP 800-230 key-use limit is exceeded through retries, multiple devices or restored backups	Exploratory arm is inapplicable or rejected; no final-standard or full-security claim
A047	Proof encoding or aggregate membership changes after a fee is signed	Fee and effects remain fixed; out-of-profile carriage fails admission instead of repricing the transaction

ID	Scenario	Required outcome
A048	Retrieval provider returns false-empty, truncated, stale or equivocated results	Coverage/completeness check fails or remains visibly incomplete; bounded recovery preserves the declared privacy model
A049	Compact blocks hide wide individual-proof gossip or an aggregator outage triggers unbudgeted full-proof propagation	Per-hop accounting exposes the cost; out-of-budget carriage fails the frozen profile
A050	Retrieval, aggregation, state or mix providers exit, censor, or demand payment outside the declared service model	Required service availability, costs and payment leakage remain within the frozen bounds or the gate fails
A051	Pruning, expiry or witness loss prevents offline recovery or revives a consumed note	R7.10 continuity and authenticated recovery hold; missing evidence cannot count as bounded operability
A052	A stateful A/B authorisation wallet crashes after signing and restores an older backup, then retries on multiple devices	No unsafe key-index reuse; recovery follows the reviewed T202 state contract or the arm fails

These are minimum cases, not a complete security test suite.

The new scenarios have the following explicit requirement/task/gate mapping. These are planned obligations; no row records executed tests or a passed gate.

Scenarios	Requirements	Responsible tasks	Evidence gates
A039	R1.7	T302, T403, T404	G4, G5, G8
A040	R7.10	T204, T302, T403, T510	G4, G5, G9; T701 activation review
A041	R1.7, R4, R5	T302, T403, T503	G4, G5, G8
A042–A043	R4, R7, R8	T304, T401, T402, T305	G3, G3A, G4, G8
A044	R4, R5	T403, T503, T504	G5, G8, G13
A045	R1, R2, R6, R9	T005, T006, T305	G3B; later G6, G7, G10 remain open
A046	R3, R4	T001, T202, T204, T305	G3A, G4, G9
A047	R4, R6, R12	T402, T304, T305, T506	G3A, G4, G13
A048	R1, R2, R10	T006, T404, T605	G3B, G6, G11
A049	R6, R9, R10	T006, T305, T604	G3B, G3A, G7, G10
A050	R2, R9, R10	T006, T305, T403, T602, T604, T605	G3B, G3A, G6, G10, G11
A051	R4, R7.10, R9, R10	T403, T404, T510, T605	G5, G9, G10, G11
A052	R3, R4, R7	T202, T204, T305	G3A, G4, G9

8. Performance and capacity report

Every benchmark report MUST contain:

1. repository revisions and build hashes;
2. exact dependency/toolchain versions;

3. machine and network inventory;
4. workload seed and transaction-shape distribution;
5. accepted, rejected, duplicate, and submitted counts separately;
6. p50/p95/p99 proof, propagation, ordering, application and finality latency;
7. average/p95/max consensus-transaction, external encrypted-payload, total-payment, proof, state/transition-witness, header, and recovery bytes;
8. unique canonical data separately from per-role inbound/outbound bandwidth, gossip amplification, wallet maintenance, bootstrap, and provider traffic;
9. executing-validator, succinct-verifier, producer, prover, state-provider, pool/share-aggregator, and archive-provider CPU, memory, disk I/O, state, prunable history, and durable recovery data separately;
10. power, thermal behavior, disk endurance, retail/connectivity evidence, bootstrap, restart, reconstruction, and repair;
11. miner-template/pool overhead, inclusion/censorship, ordering extraction, PoW hardware, and provider-concentration metrics;
12. anonymity metrics and added bandwidth/latency;
13. all failures, restarts, divergences and discarded samples;
14. commands and immutable artifact locations.

T305 reports MUST additionally separate single-wallet latency from parallel prover throughput and aggregate amortisation, report authorisation cost per arm, and label prior-work numbers as reproduced or author-reported. Consensus bytes, external encrypted-note payload bytes, total generated bytes per payment, payload-provider/retrieval traffic, and availability/retention assumptions MUST be reported separately. A digest-only layer-1 representation MUST NOT hide required payload bytes. A server aggregate rate MUST NOT be reported as client- side transaction latency.

The 1,000 tx/s acceptance target is fixed. Reference hardware and material thresholds are intentionally not fabricated in advance; T004 defines the method and T005 freezes the named profile before a result can be interpreted. A 60-second benchmark limit MUST NOT be quietly substituted for a different specification limit.

9. Security review checklist

Cryptographic correctness

- ☐ FIPS 202, 203, and 205 algorithm names and parameter sets are exact.
- ☐ Official KAT provenance is recorded.
- ☐ Every hash domain has a security property and output length justified against quantum and multi-target attacks.
- ☐ Commitment construction and parameters pass T104.
- ☐ Every random sampler is unbiased and domain-separated.
- ☐ No secret-dependent branch, memory lookup, log, or error oracle remains.
- ☐ Multi-user and protocol-lifetime composition remains 128 PQ bits.

Transaction soundness

- ☐ T402 stages encryption context, effects ID, authorisation and proof carriage without self-reference and fixes resource weight before signing.
- ☐ Duplicate notes across transactions have distinct authenticated instances; nullifiers and historical-note access remain correct across upgrades.
- ☐ Each input value is bound to an opened member commitment.
- ☐ Each output value is bound to its public output commitment.
- ☐ Authorization key is bound to the spent note.
- ☐ Authorization digest covers every semantic field.

- ☐ All ranges and sums use constrained integer limbs and carries.
- ☐ Reward minting is type-separated from ordinary conservation.
- ☐ Global anchor/nullifier checks use explicit state context.

Privacy and anonymity

- ☐ No transparent transfer mode exists.
- ☐ Public ledger fields meet the R1 disclosure budget.
- ☐ Note encryption and scanning metadata meet the recipient-key-privacy game.
- ☐ Note ciphertext and commitment cannot be swapped or replayed.
- ☐ Wallet, transport, and transaction identities are separated.
- ☐ T602 covers global passive and required active attacks.
- ☐ Incoming, full-wallet, transaction-specific, and auditor-scoped disclosure are distinct and least-privilege.
- ☐ Copied disclosure is not described as revocable or as proof of legal compliance.

Product boundary and performance

- ☐ T006 screening thresholds and modelled components are explicit; actual T305 sizes remain in the screened envelope before state/node integration.
- ☐ Both standardised stateless arms (256f and 256s) are measured, and any SP 800-230 arm remains exploratory with its total key-use bound enforced.
- ☐ T305 passes before full-node or GHOSTDAG integration starts.
- ☐ The 2-input/2-output prototype includes every required relation and each retained in-proof authorisation arm.
- ☐ Each retained stateless arm (256f or 256s) independently passes its frozen material-benefit rule against every qualifying stateful comparator; a failing 256f does not exclude 256s.
- ☐ Stateful key reuse, exhaustion, crash, restore, rollback, concurrency, recovery, and desynchronisation cases are tested.
- ☐ Wallet latency, parallel proving, verifier cost, wire size, and aggregate amortisation are reported separately.
- ☐ Digest-only note carriage reports consensus bytes, external encrypted payload bytes, total payment bytes, provider traffic, and retention assumptions separately.
- ☐ Every enabled payment policy has bounded deterministic semantics and an updated proof/resource report.
- ☐ No general-purpose VM, private asset, bridge, or adapter inherits a base security, privacy, or compliance claim.

Consensus and supply

- ☐ GHOSTDAG profile is executable and versioned.
- ☐ DAA is deterministic integer arithmetic.
- ☐ Validators recompute expected target.
- ☐ DAG ordering yields one atomic state history.
- ☐ Reward eligibility and index are DAG-native.
- ☐ Charge-class and destination fee sums, claimed fees, claimed/foregone subsidy, burned existing value, gross issuance, outstanding supply, and reward maturity are exact and non-negative.
- ☐ Any selected dynamic fee profile binds fee epoch, canonical transaction weight, prior-finalised controller state, bounded integer transitions, and finalised reward-window allocation into consensus.
- ☐ The fee-derived sustainable miner budget includes only newly miner- eligible fees; fee-derived payout capacity includes only matured claimable miner-fee-pool value.

- ☐ Monetary widths, maxima, checked arithmetic, and quotient/remainder ceil_div semantics are exact and have boundary vectors.
- ☐ The selected monetary-policy invariant includes fee replay, burn/reissuance semantics, rounding, reorganisation, and prevention of unauthorised issuance.
- ☐ Genesis bytes, timestamp units, zero fee pool, cumulative issuance, and initial controller epoch/rates are canonical.

Operability, validity and recovery

- ☐ T005 is signed before any dependent result is interpreted.
- ☐ G7 and G10 use the same revision, workload, campaign, and profile.
- ☐ Resources are separated for executing validators, succinct verifiers, producers, provers, state providers, pools, and archive providers.
- ☐ T306 proves only generic proof-system accumulation capability; T505 owns the exact Quantum consensus and monetary relation without calling a proof current-data availability.
- ☐ Executing validators receive and validate complete current data.
- ☐ Snapshot, state, wallet, and archive recovery require no trusted sole provider and pass withholding, reconstruction, repair and eclipse tests.
- ☐ Public archive distribution is not presented as network-origin privacy.

Production and incentives

- ☐ T509 precedes PoW selection and no ASIC-resistance claim replaces its measurements.
- ☐ Miner-controlled Template Provider/Job Declarator and pool/share- aggregator roles are distinct.
- ☐ Custom jobs and the non-custodial pooled-mining comparator pass T507.
- ☐ T508 covers fee, conflict, timing, duplicate, censorship, reorganisation, grinding and pool-formation strategies.
- ☐ T508 covers manipulation and free-riding for the T506-frozen direct or delayed payout, burn, and charge-to-destination allocation rule.
- ☐ Privacy is not described as eliminating MEV or transaction-selection incentives.
- ☐ G12 metrics, scenarios, thresholds, and STOP conditions were frozen by T005 before T507–T509 candidate results were interpreted.

Upgrade governance

- ☐ T510 defines exact post-quantum authorisation principals, encodings, activation rules, timelocks or thresholds, key lifecycle, compromise recovery, and any emergency path.
- ☐ Replay, downgrade, equivocation, out-of-window activation, and persistent split cases fail closed in two interoperable implementations.
- ☐ No hidden unilateral key or classical fallback can activate an upgrade.
- ☐ T510 analyses quorum capture and keeps deterministic activation distinct from voluntary social adoption; it makes no governance-decentralisation claim.
- ☐ T510 classifies immutable, normal-upgrade-only, and emergency-changeable parameters; emergency activation cannot change monetary invariants.

Monetary security

- ☐ T506 separates monetary-policy semantic, issuance schedule, fee/burn rule, and economic security model.
- ☐ Every unpaid, unburned miner-eligible fee remains in the consensus fee- pool liability and outstanding supply until deterministic payout or burn.
- ☐ Total accepted fees equal both the resource/security/priority sum and the miner-eligible/burn/non-miner-output destination sum.

- ☐ G13 scenarios, metrics, thresholds, and STOP conditions were registered before candidate results were interpreted.
- ☐ The current 21-million lifetime gross-issuance cap remains the public statement unless the exact replacement receives all required signatures.
- ☐ A positive token subsidy is not presented as a measured attack cost.
- ☐ A finite cap is not presented as evidence that future fees are adequate.
- ☐ The bounded dynamic-fee candidate reports zero-demand and low-demand underfunding truthfully, meets frozen user-cost limits, and never uses an oracle or revenue gap to authorize extra issuance.
- ☐ T508 tests the T506-frozen monetary rule without editing it; rejection starts a new campaign and rebuilds dependent T504/T507/T508 evidence.
- ☐ Product Owner, monetary-economics, consensus and legal approvals match the exact monetary revision and public wording.

Parsing and operations

- ☐ All lengths and counts are bounded.
- ☐ Unknown versions and non-canonical encodings fail closed.
- ☐ Cheap validation precedes expensive cryptography.
- ☐ Queues, memory, CPU and disk work are bounded per peer.
- ☐ Snapshots are authenticated and trustless recovery is tested.
- ☐ Logs and crash artifacts contain no wallet or network-origin secrets.

Unchecked boxes are normal at this research stage. They MUST NOT be prechecked in reports.

10. Evidence manifest and report format

Each task produces a record with:

```yaml
task_id: T301
status: NOT_STARTED  # NOT_STARTED | IN_PROGRESS | BLOCKED | PASS | FAIL
owner: null
reviewer: null
requirements: [R1, R3, R4]
source_revision: null
toolchain_manifest: null
evidence:
  analytical_argument: null
  formal_artifact: null
  test_report: null
  vector_set: null
  benchmark_artifacts: null
  external_review: null
open_findings: []
decision_date: null
decision_signature: null
```

Null fields mean **not supplied**, never “not needed” or “passed.” PASS is allowed only when every mandatory artifact for that task exists and the named reviewer signs the exact revision.

Current aggregate report

Area	Status	Evidence
Requirements draft	IN_PROGRESS	Specification 0.5.4-research

Area	Status	Evidence
Early receiving/transport screen	NOT_STARTED	T006/G3B defined; no measurements or approved envelope
Commitment	NOT_STARTED	No selected construction
Proof profile	NOT_STARTED	No frozen parameters or AIR
Private transaction	NOT_STARTED	Relation drafted; no implementation
DAG/state integration	NOT_STARTED	Requirements only
Operability profile	NOT_STARTED	T005 requirements drafted; no numeric profile frozen
Current data/recovery	NOT_STARTED	Candidate mechanisms only
Miner templates/incentives	NOT_STARTED	Requirements only
PoW hardware contestability	NOT_STARTED	No candidate comparison
Upgrade governance	NOT_STARTED	T510 mechanism and evidence absent
Monetary security	NOT_STARTED	Current lifetime cap retained; bounded dynamic-fee comparator drafted; T506 decision absent
Network anonymity	NOT_STARTED	Requirements only
1,000 accepted tx/s	NOT_STARTED	No benchmark
Independent implementations	NOT_STARTED	None
External audit	NOT_STARTED	None

11. Proposed implementation workspace

The following layout is a proposal for a future implementation repository; it does not exist here and is not a command contract:

```
quantum-protocol/
  spec/                 canonical encodings and generated vectors
  reference/            simple independent reference implementation
  node/                 production-oriented validator implementation
  wallet/               wallet, scanner and proof client
  formal/               models and formal artifacts
  tests/
    kats/
    interoperability/
    adversarial/
    fuzz/
    performance/
  evidence/             immutable manifests and signed reports
```

Build, test, fuzz, benchmark, and vector commands MUST be defined by that repository and pinned in T003/T004. This documentation repository intentionally does not publish fictitious commands that appear runnable.

12. Release labels

Only these labels may be used:

Label	Minimum meaning
Research design	Requirements and candidates; blocking gates may be open

Label	Minimum meaning
Prototype	Non-production code; no security or performance claim
Reviewed testnet candidate	T701 testnet pass on an exact revision
Public testnet	Candidate deployed with explicit residual risks
Production candidate	T701 production evidence complete; no live claim yet
Production	Exact audited revision deployed and continuously monitored

At the current revision the project is a **research design**.

13. Revision record

Unreleased — 2026-11-08

- identified the programme as Phexora Cash while retaining the historical Quantum protocol codename, task IDs and evidence contracts;
- clarified T001/T103/T104 security and construction evidence and T305 component-cost and proof-distribution measurements;
- assigned retrieval completeness, helper-service costs and failure assumptions, and state/witness recovery evidence to their responsible tasks;
- added planned scenarios A048–A052 with requirement, task and gate mappings; and
- retained independent review and exact-artifact requirements without recording a passed task, benchmark, approval or new release.

0.5.4-research — 2026-09-05

- added T006/G3B screening before T305 and retained final wallet/network gates;
- corrected interface and implementation ordering: common rules in T002, harness-only acceptance in T003/T004, canonical T401/T402 before T301/T305, and measured authorisation costs in T304/T305;
- added the 256s comparator and explicit limited-use draft applicability;
- traced note-instance/upgrade continuity and staged transaction digests to their owners, added contextual DAG acceptance, and specified A039–A047;
- advanced the graph to 41 acyclic tasks without recording a passed task, benchmark, specialist approval, or selected monetary-policy change.

0.5.3-research — 2026-08-25

- added the hard-cap dynamic-fee/reward-window comparator with explicit controller parameters, finalised inputs, user-cost limits, underfunded-state semantics, and no-oracle/no-extra-issuance boundaries to T506;
- added fee-pool liability and outstanding-supply accounting for delayed payout, including maturity, retention, burn, rounding, and reorganisation rules;
- separated accepted charge classes from miner-eligible, burn, and non-miner destinations; constrained payout capacity to matured claimable pool value;
- required exact monetary widths, checked arithmetic, canonical fee reassignment after reorgs, and explicit genesis monetary/controller state;
- made T508 a falsification gate for the T506-frozen monetary rule, with a new campaign required after rejection; and
- constrained T510 emergency upgrades from silently changing protected monetary rules and added permanent scenarios A031–A038 without marking T506, T508, T510, G12, or G13 as passed.

0.5.2-research — 2026-08-09

- derived wallet master-secret entropy from the T001/T204 multi-user, multi-target, and lifetime bound instead of treating 256-bit BIP-39 as automatic evidence for R3;
- made T505 depend on T405 before binding policy pre/post-state into the exact consensus relation;
- generalised A014 to the selected monetary-policy invariant and added A030 for governance capture and the activation/adoption boundary;
- expanded T510 with principal-concentration, bribery, collusion, abstention, veto, client-distribution, and social-adoption analysis; and
- made T305 and capacity reports expose external encrypted payloads and provider traffic when layer 1 carries only their digests.

0.5.1-research — 2026-08-09

- expanded the graph to 40 acyclic tasks by adding T510 as the owner of post- quantum upgrade governance, key lifecycle, and deterministic activation;
- required 256-bit mnemonic entropy for the full R3 wallet profile and isolated lower-entropy legacy imports behind explicit downgrade evidence;
- removed benchmark ownership from T104 and separated T306 generic proof-system accumulation capability from the exact T505 Quantum relation;
- reduced T701 to its non-redundant direct dependencies on T510 and T604 while retaining all other tasks through T604's dependency closure;
- made G12 and G13 depend on pre-registered metrics, scenarios, thresholds, and STOP conditions rather than risk descriptions alone; and
- generalised monetary evidence from cap-only/no-inflation language to the selected policy invariant and prevention of unauthorised issuance.

0.5.0-research — 2026-08-09

- expanded traceability from R1–R8 to R1–R12 and the task graph from 31 to 39 acyclic tasks;
- added T005 to freeze a dated, independently obtainable operability profile before dependent results are interpreted;
- added T306 as a SELECT/REJECT feasibility decision for succinct or accumulated validity rather than making recursion an assumed requirement;
- added T505 for the selected consensus-bound state-validity/bootstrap profile and T605 for separate current-data, snapshot, withholding, and recovery evidence;
- added T507/T508 for miner-controlled templates, non-custodial pooled-mining comparison, and ordering/censorship/pool-formation incentives;
- placed T509 hardware contestability before T502 selects the PoW algorithm;
- added T506 as the product-owned monetary decision before T504 implements the cap, schedule, fee, burn, and genesis rules;
- corrected fee, foregone-subsidy, burned-existing-value, gross-issuance, and outstanding-supply evidence requirements;
- introduced paired gates G10–G13 for operability, current data/recovery, producer contestability, and monetary security; and
- retained NOT_STARTED status for every new task and gate.

0.4.0-research — 2026-07-23

- required a signed revision of the T305 prior-art and reuse decision as a T001/T202 deliverable;
- reframed T202 and T305 as a same-relation comparison between the FIPS 205 stateless incumbent, an independently specified TzEL-shaped baseline, and an applicable NIST SP 800-208 stateful profile;

- required state-management failure vectors, isolated authorisation costs, and reproduced-versus-author-reported evidence labels;
- introduced a requirement to pre-register the material-benefit rule and explicit GO/ADAPT/REPLICATE/STOP outcome classes.

0.3.0-research — 2026-07-21

- expanded the acyclic plan to 31 tasks with T305 as the mandatory complete transaction feasibility gate before node/consensus integration;
- separated single-wallet proof latency, parallel prover throughput, aggregate amortisation, and verifier/wire budgets;
- added T405 for least-privilege disclosure and a finite, proof-bounded payment policy registry;
- made private assets and interoperability adapters separate post-core profiles with explicit external trust and claim boundaries;
- added adversarial cases for disclosure-scope escalation and unknown or unbounded payment-policy behavior.

0.2.0-research — 2026-07-09

- changed the guide from autonomous AI implementation instructions to a human-owned research and verification plan;
- defined all 29 task IDs and removed missing-task references;
- made the dependency graph acyclic;
- added requirements traceability and honest NOT_STARTED reporting;
- added direct regression cases for input-value inflation, field wrap, self-declared PoW targets, local-time DAA divergence, DAG conflicts, and anonymity correlation;
- distinguished proof, formal verification, testing, statistics, benchmarks, review, and audit;
- aligned performance acceptance with 1,000 end-to-end accepted tx/s;
- required external specialist and composition review before any release claim.
- added explicit gates for quantum digest lengths and encrypted-note recipient-key privacy;
- defined fee transfer, claimed subsidy, burns, maturity, and reorg accounting as mandatory supply evidence.