

WIZM - Post-Quantum Privacy Blockchain

Architecture, Cryptography, Consensus and Operations of a Five-Validator Production Mainnet

WIZM Protocol Engineering

14 October 2026

Mainnet-grounded: every mechanism in this paper runs in production;
measurements are taken from the live five-validator network.

Abstract

WIZM is a post-quantum, privacy-preserving payments blockchain running as a five-validator proof-of-authority mainnet with ten-second slots. Every signature is Falcon-512, every mempool entry is sealed under ML-KEM-768, every hash is BLAKE3, and shielded transfers prove in Flock, a transparent hash-based zero-knowledge system with no trusted setup and no ceremony. The ledger carries four transaction families (transparent, stealth, shielded deposit/transfer, shielded unshield) under a single flat fee, and settles up to 1,200 transactions per block. Consensus is single-branch PoA with deterministic fork resolution (earlier seal wins, 1-block unwind), blind/begin seal suppression, a stall watchdog, light checkpoints and pipelined sync. Admission is a gap-buffered encrypted mempool with per-sender floors, forfeit jumps, parked lanes, time-based reaping and lane re-grounding. This paper presents the full system: cryptographic foundations with correctness equations, the asset and fee model with the closed-form emission schedule, the state and transaction models, the consensus and mempool machines, networking and storage, the zero-configuration wallet, the operational methodology (including multi-day full-spectrum soak testing with chain-true accounting), measured performance, a security analysis, and the roadmap.

Contents

1	Introduction	4
1.1	Vision	4
1.2	What WIZM is	4
1.3	Design principles	4
1.4	Organization	4
2	System overview	5
2.1	Layers	5
2.2	Production deployment	5
2.3	Throughput envelope	5
3	Cryptographic foundations	5
3.1	Falcon-512 signatures	5
3.2	ML-KEM key encapsulation	6
3.3	BLAKE3 hashing	6
3.4	Flock transparent zero-knowledge proofs	6
3.5	Stealth addresses	6
3.6	Security posture	6

4	Assets and tokenomics	6
4.1	Units	6
4.2	Supply and emission	7
4.3	Fees	7
4.4	Validator economics and slashing	7
5	Transaction model	7
5.1	Families	7
5.2	Lifecycle	8
5.3	Nonces and replay protection	8
5.4	Client discipline	8
6	State model	8
6.1	World state	8
6.2	Shielded pool	9
6.3	Snapshots, journal, rewind	9
6.4	Retention	9
7	Consensus: five-node proof-of-authority	9
7.1	Slots and committee	9
7.2	Proposal, vote, finalize	9
7.3	One branch: fork rules	9
7.4	Seal suppression gates	10
7.5	Watchdog, checkpoints, pipelined sync	10
7.6	Why PoA	10
8	Mempool: gap-buffered encrypted admission	10
8.1	Shape and limits	10
8.2	Floors, jumps, parking	10
8.3	Reaping and re-grounding	11
8.4	Fairness and cost	11
9	Networking	11
9.1	Planes	11
9.2	Client APIs	11
9.3	Propagation discipline	11
10	Storage	12
11	Wallet	12
11.1	Zero-configuration desktop	12
11.2	Spending flows	12
11.3	Backup and recovery	12
12	Operational methodology	12
12.1	Soak design	12
12.2	Chain-true accounting	12
12.3	Recovery rehearsals	13
13	Measured performance	13

14 Security analysis	13
14.1 Threat model	13
14.2 Quantum readiness	13
14.3 Spam, sybil and money economics	13
14.4 Network and operator risks	14
14.5 Privacy boundaries	14
15 Comparison	14
16 Roadmap	14
A Parameter table	15
B RPC catalog (abridged)	16
C Glossary	16

1 Introduction

1.1 Vision

Public blockchains face two slow-moving cliffs at once. First, large-scale quantum computers break the elliptic-curve signatures and key exchange that secure essentially every production ledger; any coin whose history must stay verifiable for decades should already be migrating. Second, transparent ledgers publish every payment graph forever, which is the opposite of money. WIZM is built on the premise that the answer to both cliffs is the same engineering posture: post-quantum primitives everywhere, privacy by default, no ceremonies, no trusted setups, no special roles.

1.2 What WIZM is

WIZM is a payments chain, not a general computer. It does not run user programs. It moves value in four ways: transparent account transfers, stealth transfers to dual-key meta-addresses, shielded notes with zero-knowledge transfers inside a Merkle pool, and note-to-transparent unshielding. Consensus is proof-of-authority among a small validator committee sealing on a ten-second slot schedule. The production network runs five validators on five regions with a single branch, full blocks of up to 1,200 transactions, and a ten-day full-spectrum soak methodology that holds submitted and committed rates equal into the nineties of transactions per second with zero chain-true rejects.

1.3 Design principles

1. **Post-quantum throughout.** Falcon-512 signs, ML-KEM-768 encapsulates, BLAKE3 hashes, Flock proves. No elliptic curves anywhere on the critical path.
2. **No ceremonies.** No trusted setup (Flock is transparent), no key ceremonies beyond ordinary validator key generation, no per-deployment secrets.
3. **Privacy by default, verifiability always.** Shielded notes hide sender, receiver and amount; every proof still verifies against public roots; supply stays auditable through nullifiers and commitments.
4. **Single branch, self-healing.** Forks are resolved deterministically within seconds (earlier seal wins, one-height unwind); stalled lanes, wedged pools and lagging nodes have mechanical recovery paths instead of operator procedures.
5. **Honest accounting.** Every transaction that enters and never seals surfaces in exactly one shed counter; client-side timeouts are never counted as chain rejects. Throughput claims are reconciled by construction, not estimated.
6. **Boring operations.** Fixed ports, fixed cadences, systemd units, a live dashboard, and rehearsed recovery (rewind, wipe-and-resync, rolling restarts) instead of heroics.

1.4 Organization

Section 2 tours the system. Section 3 specifies the cryptography. Sections 4–6 cover assets, transactions and state. Sections 7–10 cover consensus, mempool, networking and storage. Section 11 covers the wallet. Sections 12–13 cover methodology and measured performance. Section 14 analyzes security, Section 15 compares, Section 16 plans. Appendices tabulate every parameter, RPC and term.

2 System overview

2.1 Layers

WIZM is five layers with narrow interfaces:

Layer	Duty	Key mechanism
1 · Network	move bytes	gossipsub + QUIC + direct unicast fan-out
2 · Consensus	one branch	5-node PoA, 10 s slots, deterministic reorgs
3 · Execution	apply money	4 tx families, flat fees, backpressured senders
4 · State	remember	RocksDB, snapshots @2000, rewind journal 32
5 · Wallet	spend	Tauri desktop + browser sidecar, zero-config

A submitted transaction is KEM-encrypted at ingress, gossiped as ciphertext, drained into proposals in nonce order, executed deterministically, sealed under a Falcon-signed header, and voted into finality. Shielded notes additionally flow through the depth-22 Merkle pool with nullifier double-spend protection.

2.2 Production deployment

The mainnet runs five validators (indices 0–4) on five regions and hosts: US, EU, AS, UK and USW. Validator links use TCP port 7777 with a UDP/QUIC relay plane; client APIs sit behind per-node TLS proxies (External port 8546 to local port 8545, Bearer API-key auth on writes) and a gRPC stub on port 50051. Clocks are disciplined by chrony for the slot schedule. Each node runs as a systemd unit with the same release binary fleet-wide; upgrades roll one node at a time with convergence checks, and full wipes re-seed from genesis in minutes when the operators choose a fresh start.

2.3 Throughput envelope

Blocks carry at most 1,200 transactions and seal about every ten seconds, giving a ceiling near 120 transactions per second sustained. Full 1,200-transaction blocks execute end-to-end (ingest, execute, persist) in roughly half a second to four seconds depending on mix. The soak program (Section 12) sustains 60–95 TPS for days with submitted and committed rates equal and zero chain-true rejects, and probes 150–200 TPS bursts with drain phases after every spike.

3 Cryptographic foundations

3.1 Falcon-512 signatures

Every signature in WIZM — transactions, votes, transport authentication — is Falcon-512 over NTRU lattices, NIST security level 1 (comparable to AES-128 against quantum attackers).

Definition 3.1 (Falcon-512 deployment). Key generation outputs (pk, sk) with $|pk| = 897$ bytes. Signing is deterministic over the message; signatures are variable-length in $[512, 690]$ bytes (typically ≈ 666). Verification satisfies

$$\text{Verify}(pk, m, \sigma) = 1 \iff \sigma = \text{Sign}(sk, m). \quad (1)$$

Transparent transactions sign bytes $[0, 56)$ of the body (sender, recipient, amount, nonce and domain tag); admission re-verifies at ingress so invalid signatures never consume block space, and the verdict is cached so execution skips redundant checks. Validator seats bind Falcon keys per epoch with snapshots, so rotation replays deterministically.

3.2 ML-KEM key encapsulation

Two parameter sets, split by duty. The mempool, committee shares and shielded note keys use ML-KEM-512 ($|pk| = 800$, $|sk| = 1632$, $|ct| = 768$ bytes); validator transport links use ML-KEM-768 ($|pk| = 1184$, $|sk| = 2400$, $|ct| = 1088$ bytes). Shared secrets are 32 bytes either way.

Definition 3.2 (KEM deployment). Mempool entries are sealed under the committee KEM public key at ingress and only decryptable inside block production:

$$\text{Decaps}(sk, \text{Encaps}(pk)) = \text{shared secret}, \quad (2)$$

feeding AES-256-GCM for the entry body. Threshold key support lets validator committees hold shares; note keys for shielded receiving are KEM-derived. Encryption covers the transaction bytes plus sender and nonce metadata used by the gap-buffered drain without decrypting.

3.3 BLAKE3 hashing

BLAKE3-256 is the single hash function: block hashes, stealth address derivation, schedule randomness, the Flock transcript, and compact-block short IDs. One primitive means one audit surface and one hardware-acceleration story (SIMD everywhere, $\approx$ GB/s per core).

3.4 Flock transparent zero-knowledge proofs

Shielded transfers prove in Flock, a hash-based (transparent) ZK system: no trusted setup, no verifier-key files, no ceremony.

Definition 3.3 (Flock deployment). Proofs are 224 bytes ($32 + 32 + 128 + 32$). Public inputs per transfer are (*nullifier*, *commitment*, *root*); the prover shows membership of the commitment under a recent root and ownership of the note without revealing sender, receiver or amount. Verification is a few milliseconds of BLAKE3 ($\approx 82,000$ hashes/s/core class throughput), so there is no per-block shielded cap.

Deposits mint commitments into the pool; transfers consume one note and mint new ones, revealing only nullifiers; unshielding drains a full note value to a transparent address in a fixed 348-byte envelope. Double-spends are refused at ingress by nullifier/key-image pins and again authoritatively at execution.

3.5 Stealth addresses

Stealth payments use dual-key meta-addresses with view tags: the sender derives a one-time address only the recipient's view key can recognize and only their spend key can move. The chain records sender and a sealed amount; receiver linkage stays off-ledger. Key images (one per spend, at fixed body offsets) make double-spends structurally impossible to seal twice.

3.6 Security posture

No elliptic curves, no pairings, no RSA, no Diffie–Hellman appear on any consensus-critical path. The quantum exposure of the system is the exposure of Falcon-512, ML-KEM-768, BLAKE3 and symmetric primitives (AES-256-GCM) — all NIST-approved or conservatively parameterized against quantum attack. Section 14 analyzes the remaining risks.

4 Assets and tokenomics

4.1 Units

The base unit is an integer; $1 \text{ WIZM} = 10^8$ base units (8 decimals). All balances, amounts, fees and rewards are integers; there is no floating point anywhere in consensus or execution.

4.2 Supply and emission

Total supply is fixed at genesis:

$$S_{\max} = 8,312,029,999,000,000 \text{ base} \approx 83,120,300 \text{ WIZM}. \quad (3)$$

Emission runs ten years in five two-year periods of 6,311,520 blocks each (≈ 10 s slots), then stops:

Period	Years	Emission (WIZM)	
1	1–2	22,248,474.30	
2	3–4	20,023,626.87	(−10%)
3	5–6	17,020,082.84	(−15%)
4	7–8	13,616,066.27	(−20%)
5	9–10	10,212,049.71	(−25%)

The per-validator block reward at height h with n validators is

$$R(h, n) = \frac{E[\lfloor h/B \rfloor / B]}{B \cdot n} \quad (h < 5B), \quad R = 0 \text{ thereafter}, \quad (4)$$

where E is the schedule above and B the blocks per period. Emission is deterministic and replayable: the same blocks always mint the same rewards.

4.3 Fees

One flat fee for everything:

$$F_{\text{tx}} = F_{\text{shielded}} = 10,000 \text{ base} = 0.0001 \text{ WIZM}. \quad (5)$$

There is deliberately no fee market and no shielded surcharge: Flock verification costs milliseconds, so spam resistance comes from uniform pricing plus admission backpressure, not auctions. During the ten emission years fees burn; after emission they distribute to validators (modulo dust stays burned). Measured sealed-fee averages sit at 10,000–10,005 across millions of transactions.

4.4 Validator economics and slashing

Validators earn the per-seat block reward plus (post-emission) fee shares. Misbehavior is punished by pattern burn, not ejection:

$$\text{3rd distinct double-seal offense within one epoch} \implies \text{burn 5\% of balance}. \quad (6)$$

One or two collisions are recorded but never punished — honest lagging nodes collide under gossip saturation — so only a pattern pays. Burn goes to zero (no reward, no ejection); ejection itself is a separate governance act. Operator recovery for shallow incidents is the `wizm_rewindTo` RPC (bounded by journal depth); deeper ones wipe-and-resync.

5 Transaction model

5.1 Families

Family	Visible on chain	Hidden	Ordering
Transparent	sender, recipient, amount	—	account nonce
Stealth	sender	recipient, amount	none (key image)
Shielded deposit	sender, amount	note contents	account nonce
Shielded transfer	nullifier	all value flow	nullifier set
Shielded unshield	recipient, amount	source note	note + nonce

5.2 Lifecycle

Every transaction walks the same path:

1. **Sign.** Falcon-512 (transparent/stealth/deposit) or Flock note construction (shielded). Nonce-bearing lanes sign strictly increasing nonces.
2. **Submit.** gRPC `SendRawTransaction` or JSON-RPC submit with Bearer API key. Admission checks: size class, signature (transparent, verified before entry), nullifier/key-image freshness (shielded/stealth, refused as duplicates), per-IP rate limit (500/s), pool capacity (honest reject when full).
3. **Encrypt + pool.** The entry is KEM-sealed and indexed by (sender, nonce). Duplicates of queued twins drop as covered no-ops; already-sealed nonces answer **EXISTS** (idempotent success).
4. **Drain.** Block production pops entries in sender/nonce order with gap buffering (Section 8).
5. **Execute.** Deterministic application with full re-verification (signatures, proofs, nullifiers, balances).
6. **Seal + vote.** The producer's block carries a Falcon seal; validators vote; finality advances (Section 7).

5.3 Nonces and replay protection

Transparent and deposit lanes are nonce-ordered per sender starting at zero; the chain applies strictly increasing nonces and execution skips anything else, so replays can neither apply twice nor brick a lane. Stealth and shielded transfers carry no account nonce; their replay protection is the nullifier/key-image set (each spends exactly once, checked at ingress and at execution). Resubmission is always safe: sealed work answers **EXISTS**, queued twins dedup, and only genuinely new work pools.

5.4 Client discipline

Reference senders batch-sign (one process per batch), submit sequentially in nonce order so the drain never gap-holds their own batch, rotate submit nodes on failure, and refill exactly the on-chain frontier window on stagnation (idempotent, uncounted). A watchdog escalates to alternate nodes after repeated useless refills and pauses (freezes) senders whose lanes are jump-frozen until the chain moves — feeding a frozen lane only parks unsealable nonces.

6 State model

6.1 World state

Definition 6.1 (World state). The state σ is the tuple

$$\sigma = (A, N, K, C, I, T, r, \rho), \tag{7}$$

where A maps addresses to account states (balance, nonce, keys), N is the spent-nullifier set, K the spent key-image set, C the commitment list with index I , T the depth-22 shielded Merkle tree, r the incremental state root and ρ the pool root. Mutation is journaled per block for rewinds.

6.2 Shielded pool

The pool is a binary Merkle tree of depth 22 (capacity $2^{22} = 4,194,304$ notes). Deposits append commitments; transfers prove membership under a recent root from a short ring of retained roots (so path-fetch races resolve instead of stalling spenders) and reveal nullifiers; unshielding drains whole notes. The tree is fail-closed: a full pool rejects new deposits rather than degrading.

6.3 Snapshots, journal, rewind

Execution snapshots export full rest-state every 2,000 heights (a legacy 250-height loader remains for old data); a boot that finds a snapshot verifies roots, loads it, and replays only the tail. Every commit seals an undo frame into a rewind journal capped at 32 depths, persisted per commit so rewinds survive restarts. Operator rewind (`wizm_rewindTo`) unwinds executor, storage mirrors, machine, notary and mempool to any journaled height and re-queues the losing fork's transactions; anything deeper needs wipe-and-resync.

6.4 Retention

Bodies are kept for 4,000 heights, headers for 200,000, checkpoints lightly every 100 (Section 7). Anything older is pruned; sync serves from what is kept; ancient history replays from genesis plus snapshots.

7 Consensus: five-node proof-of-authority

7.1 Slots and committee

Time is divided into 10-second slots with 4 seconds of grace. Heights advance one block per slot under chain-time pacing (a block needs parent time + slot before sealing, so catch-up bursts cannot compress history). A five-seat committee maps heights to producers by attempt rotation; epochs span 360 heights with ratified snapshots for seat changes. Fallback attempts fire on wall-clock delay (60 s), so a slow mesh never triggers competing versions by itself.

7.2 Proposal, vote, finalize

The entitled producer drains up to 1,200 transactions, executes them, seals with a Falcon signature over (header hash || height), and fans the proposal out by direct unicast to all four peers plus gossip backup. Validators vote per height; finality fast-forwards to tip-128 at boot and then advances the contiguous-majority watermark lockstep behind the tip. A node that already proposed a height dedups rather than re-minting; a lost own-seal is retransmitted byte-identical.

7.3 One branch: fork rules

Proposition 7.1 (Deterministic convergence). *Competing versions of one height resolve by earlier seal (lower-hash tiebreak); the loser unwinds exactly one height and adopts. The rule is strictly improving, so versions converge without oscillation.*

Remark 7.1 (Mechanisms). Same-height convergence reorgs the tip on a better competing version; fork evidence (a competing version of our tip in hand while the mesh sealed beyond) unwinds one height onto the mesh version — including on the sync-response path, so followers cannot extend a fork blindly; ghost buffers prune after a 180 s sync stall; rewinds re-queue the losing fork's transactions back into the pool.

7.4 Seal suppression gates

A node must not seal what the mesh already decided:

- **Mesh settle:** no own proposals for 150 s after boot while the mesh reforms (receiving, adopting and voting continue).
- **Blind boot:** with no mesh sample at all, no seals for 600 s (genesis+1 exempt — somebody must seal first).
- **Behind gate:** a fresh mesh sample above our next height suppresses our seals while we lag.
- **Flap gate:** a stale sample still binds while any mesh contact arrived within 900 s — a node that heard the chain minutes ago is flapping, not solo, and must keep following instead of minting stale competing versions. True-solo nodes still seal via the blind rule.
- **First-seen:** a fresh buffered proposal for our next height means the mesh claimed it; we wait up to 120 s.
- **Syncing:** a node with buffered blocks above its tip never seals (its fallback timers are all due; any seal would be a competing version on stale state).

7.5 Watchdog, checkpoints, pipelined sync

A stall watchdog fires when the tip freezes 10 minutes while a fresh mesh sample leads by 12+ (or the pool idles full 30 minutes): it clears sync spans, sheds the oldest half-pool and re-syncs, at most once per 30 minutes. Light checkpoints every 100 heights carry header ancestry only (no seal stalls); recovery never trusts more than one checkpoint of unverified history. Catch-up pipelines up to three 100-height forward windows in parallel whenever the mesh leads by more than 8 — and, since the catch-up fix, whenever it leads by more than 32 even without a stall (parity followers never trip the stall clock yet must still converge); adoption stays strictly in order.

7.6 Why PoA

Payments need finality in seconds, fees in fractions of a cent, and energy measured in servers rather than power plants. Five known validators with deterministic rotation give all three; the decentralization path (Section 16) grows the committee without changing the rules.

8 Mempool: gap-buffered encrypted admission

8.1 Shape and limits

The pool holds KEM-encrypted entries indexed by (sender, nonce) in drain order, capped at 500 MB / 200,000 entries with a 5,400 s (90-minute) TTL. Admission sheds honestly past 100,000 entries (lane heads always admit, so lanes stay recoverable), rejects at the hard caps, and answers duplicates as covered no-ops. Every entry that enters yet never seals surfaces in exactly one shed counter (valve, settled, TTL, RSS, tail, forfeited, prescreened, duplicates) — the shed invariant operators watch.

8.2 Floors, jumps, parking

Definition 8.1 (Drain floors). Each sender has a floor: the next applicable nonce, advanced by commits. A drain emits entries head-first per sender, advancing a per-pass expectation so full nonce chains flow in one block.

Entries ahead of their floor gap-hold: they park aside (indexed and tracked, out of the scan path) instead of burning scan budget, and unpark the moment the floor reaches them. When nothing is queued in [floor, head) and the head is older than 300s, the prefix is proven lost and the floor *jumps* over it, forfeiting only the absent gap and jump-marking the lane: the forfeited range never applied on-chain, so emitting the head would manufacture an `InvalidNonce` skip. Marked lanes hold until commit-grounded floors advance past the mark (refills filled the hole), with a live-truth overlay letting arriving refills emit head-first. Late arrivals below a jumped floor drop as settled dups, so jumps are irreversibly safe.

8.3 Reaping and re-grounding

Three mechanisms bound rot: the pressure-gated liveness valve (drops long-held trailless dead heads past 80% fill), the ungated time-based sweep (TTL-expired parked entries die on a 60s timer no matter the fill — frozen lanes cannot rot forever at low pressure), and lane re-grounding (an emptied jump-marked lane resets floor and mark on its next tracked arrival, so senders restarting from chain truth flow instead of settled-dropping forever). Refill windows from clients (up to 500 ahead of the frontier) close holes the jumps skipped; a watchdog pauses senders whose lanes stay frozen past repeated refills so no one feeds unsealable nonces.

8.4 Fairness and cost

Drains interleave shielded and transparent entries round-robin (no shielded cap under Flock) with sender-partitioned causal dedup, so no lane starves and no double seal wastes space. The drain holds one write lock with a bounded scan (at most 16× the take, capped), logging loudly past 250ms; a full pool that yields nothing warns as starved with parked/stale/settled/tombstone counts attached.

9 Networking

9.1 Planes

Validator links run TCP port 7777 with a UDP/QUIC relay plane; gossipsub carries four topics (transactions, vertices, votes, sync) with peer scoring and shed odometers, and every proposal additionally fans out by direct unicast to all four peers so doubles resolve by height dedup downstream. Consensus traffic goes first; bulk transactions shed first under pressure, with per-topic counters proving it.

9.2 Client APIs

Nodes expose a gRPC stub (port 50051) and JSON-RPC behind per-node TLS proxies (external 8546 to local 8545): tip, blocks, sealed headers, balances, nonces, mempool stats, tokenomics, shielded commitments and notes, plus submit and operator rewind paths. Writes require Bearer API-key auth on both stacks; reads are open. A per-IP token bucket (500/s) sheds floods honestly, and nginx plus host firewalls pin the RPC face to known operators.

9.3 Propagation discipline

Proposals propagate by unicast + gossip backup within seconds; re-broadcasts repeat the same bytes (never a new version) after a 60s delay and never while syncing behind. Sync serves spans on request with pipelined forward windows; sync responses also carry fork evidence so followers unwind instead of extending forks.

10 Storage

Each node persists in RocksDB (128 MB block cache, 64 MB write buffers): block bodies (4,000 kept), headers (200,000 kept), executor snapshots (every 2,000), the rewind journal (persisted per commit, 32 deep), slash marks, committee snapshots and tokenomics. Body/header retention splits prune cost from sync cost; ancient history replays from genesis plus snapshots. A two-mode allocator setup (bounded arenas, 1 s dirty/muzzy decay, background purge thread) plus a periodic five-minute arena purge keeps RSS flat across days of full-block sealing.

11 Wallet

11.1 Zero-configuration desktop

The wallet is a Tauri v1 desktop app (macOS universal, Windows portable, Linux deb/rpm/AppImage/Flatpak on x86_64 and arm64) whose UI is three files served two ways: inside the app and, byte-identical, by the signing sidecar over the LAN. Node URL, API key and TLS cert are embedded or server-side; there are no settings screens, no saved knobs, and therefore no misconfiguration states. Opening the app connects.

11.2 Spending flows

Transparent sends sign-and-submit with nonce management and **EXISTS-as-success**; shielded deposits mint notes, transfers prove-and-mint with recipient scan pages, and unshields drain whole notes in 348-byte envelopes with progress, retry and rescan paths that survive fat blocks and plain-HTTP LAN transports (clipboard fallbacks included). The sidecar signs locally; keys never leave the user's machine or LAN.

11.3 Backup and recovery

Vault export/import uses pure-implementation PBKDF2-SHA256 (600,000 rounds) + AES-256-GCM in the same envelope on every platform, verified against eight test vectors including cross-implementation round-trips. Note identity is per-app: recipients recover by scanning where they deposited, which the UI guides.

12 Operational methodology

12.1 Soak design

Load runs are scripted five-minute phases from seeded schedules: warmup ramps (20/50/80), then random levels with a cooldown rule — every phase at or above 100 TPS is followed by a 20/50 drain phase, so spikes backfill by construction instead of stacking. A recent envelope runs 2,880 phases (ten days, ≈ 69 TPS average) over 140 funded accounts with refills, watchdog escalation to alternate nodes, freeze-pauses for wedged lanes and nonce recovery that never walks backward. Funding is parallel and verified on-chain before phases start.

12.2 Chain-true accounting

Submitted (SR), committed (CR) and their instant variants (ISR/ICR) are reconciled every cycle; rejects count only chain verdicts (mempool-full, invalid, rate-limit) while client timeouts, unreachable and mid-restart cancellations count separately as timeouts. A run is green when SR $\approx$ CR into the nineties with zero rejects — achieved repeatedly, including multi-hour 150 TPS stretches.

12.3 Recovery rehearsals

Forks: verify same-height divergence, unwind or wipe-and-resync the minority, confirm hash lockstep. Stalls: watchdog evidence, restart into drains. Deep lags: snapshot restore plus pipelined catch-up, or state copy for the impatient. Fleet upgrades: one node at a time with convergence checks; full wipes re-seed genesis in minutes. The live dashboard tracks every block within seconds with fork pill, per-node gauges (pending, parked, frozen lanes) and reason tooltips.

13 Measured performance

All figures from the production five-validator network, October 2026 soak runs (10 s slots, 1,200-tx blocks, mixed transparent/stealth/ shielded load).

Metric	Observed
Sustained throughput (SR $\approx$ CR)	60–95 TPS for days
Burst throughput	120–200 TPS phases, backfilled in drains
Block seal cadence	$\approx$ 10 s, full 1,200-tx blocks
End-to-end block execution (1,200 txs)	0.5–4.0 s
Chain-true rejects (10-day envelope)	0 at ≤ 150 TPS levels
Client timeouts at 150+ TPS	counted separately, retried as dups
Single soak volume	4M+ transactions committed
Mempool under soak	bounded, parked lanes reaped by TTL sweep
Node RSS (empty pools, post-purge)	flat across days

Remark 13.1 (Reading the numbers). Throughput here means *committed* transactions (chain-true), not submitted attempts. Timeouts, retries and duplicates are reported separately and never folded into either rate, which is why SR/CR equality is a proof rather than a slogan.

14 Security analysis

14.1 Threat model

Adversaries may submit arbitrary transactions (rate-limited, fee-paying, authenticated), run lossy or malicious network links, operate lagging or flapping validators, and — eventually — wield quantum computers. They may not break Falcon-512, ML-KEM-768, BLAKE3 or AES-256-GCM, invert the shielded pool without keys, or forge validator seats outside the ratified committee.

14.2 Quantum readiness

All consensus-critical cryptography is NIST post-quantum (Falcon-512 signatures, ML-KEM-768 encapsulation) or symmetric/hash-based with conservative parameters (BLAKE3-256, AES-256-GCM, hash-based Flock proofs). There is no elliptic-curve, pairing or RSA dependency to migrate later, and no trusted-setup toxic waste to protect today.

14.3 Spam, sybil and money economics

Spam pays the flat fee per attempt and meets layered shedding (rate limit, early shed, hard caps, per-address burst windows); sustained spam at 150 TPS costs ≈ 1.3 M base/s with zero effect on committed honest flow (pools bound, seals continue). Sybil senders gain nothing: ordering is per-sender and fees are uniform. Double-seals are slash-burned only as patterns, so honest lagging collisions are never punished while persistent offenders pay 5% per epoch.

14.4 Network and operator risks

Lossy links cause flapping, not forks: behind/flap gates suppress stale seals, evidence unwinds the rest within seconds, and pipelined catch-up reconverges laggards without stalls. Operator keys live per box (no central store); RPC writes need Bearer keys; host firewalls pin admin faces. Documented residual risks: a validator host loss takes its seat dark until repair (the chain seals on four); deep network partitions pause minorities instead of forking them; ancient history trusts checkpointed ancestry plus snapshots.

14.5 Privacy boundaries

Shielded notes hide sender, receiver and amount on-ledger; stealth hides receiver and amount with visible sender. Transparent transfers are fully public. Network-layer metadata (who submits, when) is not obscured — operators needing that use their own submission hygiene. Deposit-to-transfer linkage is broken by design; transfer-to-unshield linkage is visible at the unshield edge by necessity.

15 Comparison

	WIZM	Bitcoin-like	Account L1s	Privacy coins
Post-quantum sigs	Falcon-512 everywhere	no	no	no
Shielded default	notes + stealth	no	no	yes
Trusted setup	none	n/a	n/a	often
Finality	seconds (PoA)	$\approx 1\text{h}$	seconds–min	minutes
Fees	flat 0.0001	auction	auction	mixed
Throughput	≈ 120 TPS	≈ 7 TPS	1k+ TPS	tens TPS
Energy	5 servers	PoW nations	PoS committees	PoW/PoS

WIZM does not compete on raw TPS with high-throughput account chains; it competes on the conjunction lesser systems split apart: quantum-safe signatures *and* private payments *and* five-second-grade finality *and* flat fees, in one running mainnet.

16 Roadmap

1. **Committee growth.** From 5 toward tens of validators with the same slot/attempt math; seat snapshots and rotation already support it.
2. **Light clients.** Header-ancestry proofs plus checkpoint verification for wallets that do not follow full blocks.
3. **Independent audit.** External review of the Flock circuit binding, the mempool shed invariant and the PoA gate matrix.
4. **Fee-market research.** Current flat pricing holds through observed loads; revisit only with evidence, never in anticipation.
5. **Shielded pool growth.** Depth-22 gives 4.19M notes of headroom; rotation/archival design before it binds.
6. **Governance.** Committee add/remove operations exercised on testnets before any mainnet seat change.

References

- [1] NIST, *Post-Quantum Cryptography Standardization*, FIPS 203/204/205 (ML-KEM, Dilithium, SPHINCS+), 2024. (WIZM uses Falcon-512/ML-KEM-768 class primitives.)
- [2] O. et al., *BLAKE3: One Function, Fast Everywhere*, 2020.
- [3] G. et al., *GossipSub: Attack-Resilient Message Propagation in the Filecoin and ETH2.0 Networks*, 2020.
- [4] J. Iyengar, M. Thomson, *QUIC: A UDP-Based Multiplexed and Secure Transport*, RFC 9000, 2021.
- [5] Tauri Programme, *Tauri v1 documentation*, 2024.

A Parameter table

Parameter	Value	Role
Validators	5 (indices 0–4)	PoA committee
Slot / grace	10 s / 4 s	seal cadence
Epoch	360 heights	seat snapshot granularity
Fallback delay	60 s	competing-version guard
Max txs / block	1,200	proposal cap
Fee (all types)	10,000 base	flat pricing
Supply	83,120,299.99 WIZM	10-yr, 5 periods
Slash	5% on 3rd offense/epoch	pattern burn
Mempool	500 MB / 200,000 / TTL 5,400 s	admission bounds
Early shed	100,000 entries	honest rejects
Staleness bar	300 s	jump/valve timing
Refill window	500 nonces	hole healing
Parked sweep	60 s cadence	untgated TTL reap
Settle blackout	150 s post-boot	mesh formation
Blind suppress	600 s	solo liveness bound
Sample stale	300 s	behind-gate reopen
Contact window	900 s	flap suppression
Rewind journal	32 depths	operator recovery
Checkpoints	every 100 (light)	ancestry granularity
Snapshots	every 2,000	fast restore
Finality lag	128 heights	watermark
Bodies / headers kept	4,000 / 200,000	retention
Sync pipeline	3 × 100-height windows	catch-up
Catch-up trigger	32-height lead	parity-treadmill fix
Stall watchdog	10 min frozen + 12 lead	ghost-busting
Memory purge	300 s	allocator hygiene
Merkle depth	22 (4.19M notes)	shielded pool
Unshield envelope	348 B	note drain
Falcon-512 pk / sig	897 B / ≈690 B	signatures
ML-KEM-512 pk / ct (pool/notes)	800 B / 768 B	encapsulation
ML-KEM-768 pk / ct (transport)	1,184 B / 1,088 B	link keys
Rate limit	500/s per IP	flood shedding
Ports	7777 TCP · 7778 UDP · 8546/50051 API	fixed planes

B RPC catalog (abridged)

Method	Returns
wizm_getTip	height + hash (executed frontier)
wizm_getBlock	tx count, timestamp, body
wizm_getSealedHeader	hash, validator index
wizm_getBalance / wizm_getNonce	account state
wizm_getMempoolStats	pending, bytes, shed odometers, gauges
wizm_getTokenomics	supply, emitted, fees burned/distributed
wizm_getShieldedCommitments/Notes	pool data for wallets
wizm_sendRawTransaction (gRPC)	admission verdict
wizm_rewindTo	operator unwind (journal-bounded)

C Glossary

Flap A node alternating between mesh contact and silence on lossy links; the flap gate keeps it following instead of sealing.

Floor Per-sender next-appliable nonce in the mempool drain.

Forfeit Skipped absent nonces when a floor jumps a proven gap.

Ghost A buffered sync span for blocks that never link; pruned after the stall window.

Jump mark A lane flag recording forfeit-advance (not commits).

Parked lane Gap-held entries held aside, indexed, out of scan.

Reground Resetting an emptied marked lane to a fresh arrival.

Seal A producer's signed block; **sealing** is minting it.

SR/CR/ISR/ICR Submitted/committed (and instant) rates.

TMO Client-side timeouts: never chain verdicts, never rejects.