NABD: A Heartbeat-Driven, Human-Gated Layer-1 with Deterministic Finality for Sovereign Digital Economies

khaled@nabdchain.org
NABD Foundation

Abstract. NABD is a layer-one blockchain for sovereign digital economies that need deterministic settlement, operational resilience, and accountable human authorization. Its consensus layer, NabdBFT, runs four deterministic phases (PROPOSE, VOTE, CERTIFY, COMMIT) under a strict $Q = \lfloor 2N/3 \rfloor + 1$ quorum, and introduces a *proactive heartbeat* liveness mechanism that monitors validator health continuously and independently of the consensus round. Where PBFT, Tendermint, and HotStuff detect a failed leader *reactively* (only after a full round timeout), NABD classifies a silent validator as suspect within 1.5 s and dead within 2.3 s, and pre-emptively triggers view change, completing leader failover in about 3.3 s: three to ten times faster than classical reactive protocols. Finality is deterministic and fast: a mean hash-confirmed public-ingress finality of 1.214 s (p95 1.436 s) on a live testnet running on modest two-vCPU, four-gigabyte shared instances. Above this base, NABD makes human authorization a first-class primitive: biometric evidence is matched outside public chain state by trusted capture hardware, and the chain records only commitments, attestations, public keys, and signed policy effects, gating treasury and asset movement, governance, and emergency recovery. The ledger adds native multi-asset state, deterministic autonomous-program execution, and a post-quantum-ready signature envelope (FN-DSA / FIPS 206 with Ed25519). The network is live at `testnet.nabdchain.org`.

Index Terms. blockchain, Byzantine fault tolerance, deterministic finality, liveness, self-healing networks, biometric authorization, autonomous programs, post-quantum cryptography, finality measurement, digital identity

2.3 s	**3.3 s**	**1.214 s**	$N{=}8$
leader-failure detection	leader failover	mean finality	live testnet

I. Introduction

Digital economies include people, companies, cities, AI agents, robots, sensors, and public services. These systems need a ledger that is final, auditable, inexpensive to operate, and resilient when ordinary computers fail. They also need a way to bind critical actions to human presence without turning human bodies into public data.

Classical BFT protocols meet the safety bar but recover slowly. PBFT-style view change runs ten to thirty seconds in practice, and Tendermint and HotStuff still detect a failed leader *reactively*: the system waits for a full round timeout to expire before it begins recovery. For a settlement layer that real economies depend on, minutes, or even tens of seconds, of stalled finality on a single crashed leader is the wrong failure mode.

NABD takes a different posture. A heartbeat subsystem monitors every validator's health continuously, decoupled from and running faster than the consensus round, so a stalled or dead leader is detected in seconds and view change fires pre-emptively rather than after a timeout. NABD treats finality as a shared economic pulse, liveness as a proactive protocol signal, and biometric identity as human authorization for recovery and high-value actions. The result combines machine-speed deterministic finality with human accountability, on hardware a community can actually afford to run.

II. Contributions

The paper makes five contributions:

- a *proactive* heartbeat liveness mechanism that detects a failed leader within a bounded T_{dead} and triggers pre-emptive view change, cutting leader failover to about 3.3 s versus the reactive, post-timeout recovery of PBFT, Tendermint, and HotStuff;
- a deterministic four-phase BFT settlement path with explicit quorum boundaries and view-scoped leader recovery;
- a ledger model with canonical transaction hashing, native assets, and autonomous program execution under deterministic state rules;
- a privacy boundary for biometric human authorization in which raw body data and matching logic stay outside public ledger state;
- a measurement model for public-ingress finality, health, and release assurance anyone can repeat from observable protocol artifacts.

III. Design Principles

- **Deterministic settlement**: a finalized block is the shared record of state rather than a probabilistic settlement estimate.
- **Self-healing liveness**: validators are tracked as alive, suspect, dead, and recovered so the network can move away from stalled views.
- **Human-scale time**: block rhythm is designed for wallets, devices, AI agents, robots, and human review.
- **Native assets**: fungible assets and unique units are part of ledger state and consensus validation.
- **Biometric circuit breakers**: critical policy can require live palm-vein device authorization without publishing biometric templates.
- **Cryptographic agility**: a scheme-aware envelope supports classical and post-quantum authentication policies.

IV. Related Work

NABD sits in the tradition of state-machine replication for Byzantine environments. PBFT established the practical three-phase pattern for permissioned replicas and the quorum-intersection argument used by many later systems [1]. BFT-SMaRt refined this line into a reusable Java replication library with reconfiguration and production-oriented engineering [2]. HotStuff reduced leader-change complexity through chained quorum certificates and linear view change [3], while Tendermint adapted BFT voting to blockchain operation with proposer rounds, validator sets, and immediate finality under partial synchrony [4]. NabdBFT does not claim lower asymptotic message complexity than these protocols. Its contribution is an explicit liveness repair plane around a conservative quorum path, together with a separate CERTIFY evidence boundary used by catchup and view-change logic.

The liveness model draws on the failure-detector view of distributed systems. Chandra and Toueg showed that unreliable failure detectors can be sufficient for consensus when their assumptions are stated precisely [5]. NABD follows the same discipline by keeping alive, suspect, dead, and recovered labels outside the commit predicate. A liveness label can start repair or view change, but it cannot lower quorum size or replace signatures. The formal model follows the TLA+ tradition of specifying safety invariants before implementation claims [6], [7]. Its current role is supporting evidence: it checks agreement, validity, vote and certify uniqueness, locks, and view-reset behavior within the bounded model recorded in the repository.

Permissioned blockchain architecture has also influenced NABD. Hyperledger Fabric separates endorsement, ordering, validation, and commit, and its execute-order-validate model treats deterministic validation and policy as first-class concerns [8]. Fabric private data collections keep raw private payloads off the public channel while committing hashes for audit and validation [9]. NABD uses a different consensus engine and a single native ledger model, but it adopts the same architectural lesson: public state should carry commitments and policy effects, while private material stays behind an explicit boundary.

Leader unpredictability and randomness are a related concern in public or semi-public validator settings. Algorand uses verifiable random functions for cryptographic sortition [10], and RFC 9381 standardizes ECVRF semantics for verifiable pseudo-random outputs [11]. NABD implements VRF leader selection, but it is hard-gated behind an explicit environment flag and a public-network predicate, so it is *not* active in the current deployment. The implementation currently uses libsodium's draft-03 ECVRF interface; final RFC 9381 byte compatibility remains a migration target before production VRF activation. Production operation uses stable round-robin leadership with strict signatures and certify locks in public hardening mode.

The authentication layer relates to post-quantum transition work. NABD implements Falcon-512, the concrete liboqs parameter set its code and tests run today, which is the parameter set being standardized by NIST under the forthcoming FIPS 206 as *FN-DSA* [12], [13], [15]. NIST FIPS 204 already standardizes the larger-signature ML-DSA scheme [14]; NABD chooses Falcon-512 (the FN-DSA parameter set) for its compact signatures. NABD is therefore *post-quantum ready*, not FIPS 206 validated: the running implementation will not claim FIPS 206 compliance until the final FIPS 206 byte formats and known-answer tests are adopted by the runtime. NABD's practical position is crypto-agility: Ed25519, Falcon-512, and hybrid signatures can authorize the same canonical transaction payload while policy decides which forms a deployment accepts.

Finally, the biometric authorization model follows prior warnings about biometric privacy and revocation. Biometric systems must separate templates, feature extraction, matching, and public audit data because compromised body signals cannot be rotated like ordinary passwords [16], [17]. NABD therefore treats palm-vein evidence as an off-chain device or oracle event and records only commitments, attestations, keys, and signed policy effects on chain. WASM-based autonomous programs use the same narrow boundary: deterministic code executes through a bounded host interface, following the WebAssembly design goal of a portable, sandboxed execution format [18].

V. System Model

Let N be the number of validators in the active validator set. The quorum size is:

$$Q = \left\lfloor \frac{2N}{3} \right\rfloor + 1$$

The quorum rule gives the chain a deterministic finality boundary: validators accept a block when the protocol observes the required support under the active validator set and validity rules, without relying on probabilistic confirmation depth.

NABD is designed for known-validator networks under the standard partial synchrony assumption used by BFT protocols: messages may be delayed during faults, but after the network stabilizes, correct validators can communicate within bounded time. Safety requires fewer than one third of active validators to be Byzantine for a round. For any two quorums Q_1 and Q_2:

$$|Q_1 \cap Q_2| \geq 2Q - N$$

Let $f = \lfloor (N-1)/3 \rfloor$ be the Byzantine fault bound. The quorum rule above is equivalent to $Q = N - f$. Two quorum certificates therefore intersect in at least $N - 2f$ validators, and since $N \geq 3f + 1$, that intersection contains at least $f + 1$ validators. At least one intersecting validator is correct, which is the basis for preventing two conflicting committed blocks at the same round and view when validators obey the locking and validation rules.

VI. Safety Argument

The safety argument is deliberately separated from the liveness classifier. Liveness signals may request repair or a new view, but they do not lower quorum size, change transaction validity, or authorize commit without a certificate.

Lemma 1 (quorum intersection). Let $f = \lfloor (N-1)/3 \rfloor$ and $Q = \lfloor 2N/3 \rfloor + 1 = N - f$. Any two quorums intersect in at least $f + 1$ validators.

Proof. Two quorum sets Q_1 and Q_2 have $|Q_1| = |Q_2| = Q = N - f$. Therefore $|Q_1 \cap Q_2| \geq |Q_1| + |Q_2| - N = N - 2f$. Since $N \geq 3f + 1$, $N - 2f \geq f + 1$. At most f validators are Byzantine, so the intersection contains at least one correct validator.

Lemma 2 (view-scoped locking). A correct validator does not certify two different block hashes for the same round, including across views. When a higher view is formed, the proposal path must preserve the highest valid lower-view lock known to quorum evidence at proposal time.

Argument. The implementation and TLA+ model represent correct validator vote and certify uniqueness, together with per-validator certify locks across views. A correct validator that has locked a hash only releases or advances that lock through higher-view evidence that is itself quorum-backed. Therefore two conflicting hashes cannot both collect valid certify evidence unless a correct validator violates its locking rule.

Lemma 3 (liveness-plane safety preservation). The alive/suspect/dead/recovered classifier cannot create a conflicting commit.

Argument. Liveness classification changes repair behavior and view selection only. It is not an input to the block validity predicate, it is not a replacement for signatures, and it does not enter the commit predicate as a quorum substitute. A suspect or dead signal can start recovery, but the recovered path must still present ordinary quorum evidence before commit.

Theorem (agreement for $N \geq 3f + 1$). For any active validator set with $N \geq 3f + 1$ and at most f Byzantine validators, two correct validators cannot commit different valid blocks for the same round.

Proof sketch. Assume two correct validators commit conflicting hashes h and h'. Each commit requires quorum-backed certify evidence. By Lemma 1, the two quorums intersect in at least one correct validator. By Lemma 2, that correct validator cannot certify both hashes in the same round/view, and cannot carry a conflicting lock into a higher view without quorum-backed evidence. By Lemma 3, liveness repair cannot bypass this certificate path. The assumption therefore contradicts correct-validator locking and quorum intersection.

The formal model under `formal/tla/nabdbft/` covers agreement, validity, correct-validator vote/certify uniqueness, per-validator certify locks, and proposal-time correct-leader reproposal of lower-view locks. The model is bounded supporting evidence for the proof obligations above, not a replacement for the quorum-intersection proof.

VII. Consensus, Liveness, and Time

NabdBFT orders transactions through four logical phases:

1) **PROPOSE**: a leader builds a candidate block for a round and view.
2) **VOTE**: validators validate the candidate and vote for its block hash.
3) **CERTIFY**: validators observe quorum support for the candidate.
4) **COMMIT**: validators finalize the block according to the active rule.

TABLE I
Validator Liveness States

State	Meaning	Protocol Role
Alive	Recent activity observed	Normal voting and peer health
Suspect	Silence exceeds suspect threshold	Early liveness signal
Dead	Silence exceeds failure threshold	View change and repair signal
Recovered	Activity resumes	Peer returns to normal operation

A. Message Complexity

For a validator set of size N, the leader sends one proposal to the other validators, giving $O(N)$ proposal messages plus block bytes. In the common broadcast form, each validator then sends vote, certify, and commit evidence to the peers, giving at most $3N(N-1)$ small consensus messages per round. The round therefore has $O(N^2)$ control-message complexity and $O(NB + N^2S)$ byte complexity, where B is block payload size and S is the size of signed control evidence. Heartbeats add a separate $O(N^2)$ health stream per heartbeat interval, but that stream does not enter the commit predicate. With the current $N = 8$ topology, the three all-peer phases account for at most 168 peer control messages per round, before retries, catchup, or client relay traffic.

B. Why CERTIFY Is Explicit

The CERTIFY phase is not a latency optimization. It is an evidence boundary. It materializes the set of validators that observed and accepted a candidate before commit, gives late or recovering nodes a compact certificate to verify during catchup, and makes the view-change lock visible to the repair plane. A three-step design can merge this evidence into neighboring phases, but then catchup and view-change logic must reconstruct which validators observed which candidate. NabdBFT pays one additional quorum-observation step to make that boundary explicit.

The protocol adds a liveness plane around this quorum path. Every peer is tracked by recent activity:

The four-state liveness model gives the chain a self-healing control loop around quorum safety. It helps the network maintain progress when a leader stalls, a peer stops responding, or a machine returns after a fault.

Crucially, this monitoring is *proactive* and decoupled from the consensus round. A heartbeat subsystem runs at a higher frequency than the round itself: each validator emits a heartbeat every $T_h = 0.5\,\mathrm{s}$, and any message from a peer (not only heartbeats) refreshes that peer's liveness, so active consensus participation doubles as a liveness signal. A peer silent for $T_{\mathrm{suspect}} = 1.5\,\mathrm{s}$ is marked suspect; silent for $T_{\mathrm{dead}} = 2.3\,\mathrm{s}$,

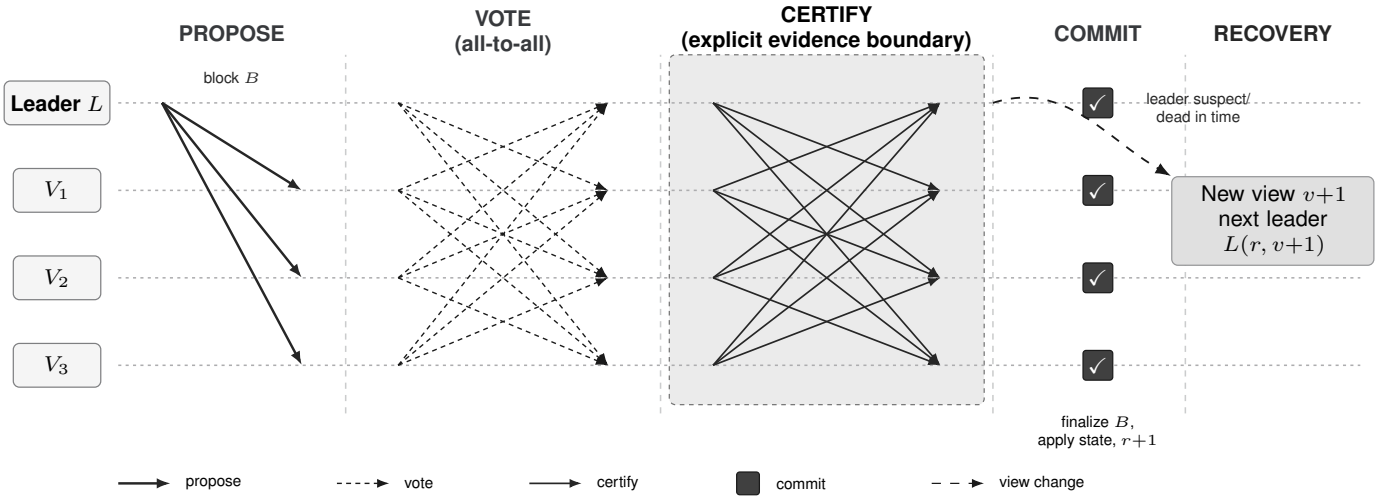

Fig. 1. One NabdBFT round over a leader and three replicas (representative of the active set). PROPOSE is a single $O(N)$ broadcast; VOTE and CERTIFY are all-to-all $O(N^2)$ phases. CERTIFY is a deliberate evidence boundary, not a latency optimization: it materializes the quorum that observed B so catchup and view-change can verify it directly. Liveness may trigger the recovery branch to a new view, but commit still requires quorum-backed certify evidence.

Round algorithm: NabdBFT round r, view v

1. Select leader $L(r, v)$ from the active validator set.
2. L proposes block B with height, round, view, parent hash, transaction root, state precondition, and signature.
3. Each validator checks chain identity, proposal signature, parent, state precondition, transaction validity, account nonces, balances, fee policy, and autonomous-program (AP) or asset rules.
4. A valid proposal produces a signed vote for $H(B)$ scoped to (r, v).
5. A validator that observes $\geq Q$ valid votes for $H(B)$ emits a signed certify message and records a view-scoped lock.
6. A validator that observes $\geq Q$ valid certifies commits B, applies the deterministic state transition, records the post-state hash, and advances to $r + 1$.
7. If liveness marks the leader suspect or dead before quorum evidence appears, validators enter view change; the next view must preserve the highest valid lock known to quorum evidence.

Fig. 2. Round algorithm corresponding to Fig. 1. Steps map to the PROPOSE, VOTE, CERTIFY, and COMMIT phases; step 7 is the recovery branch.

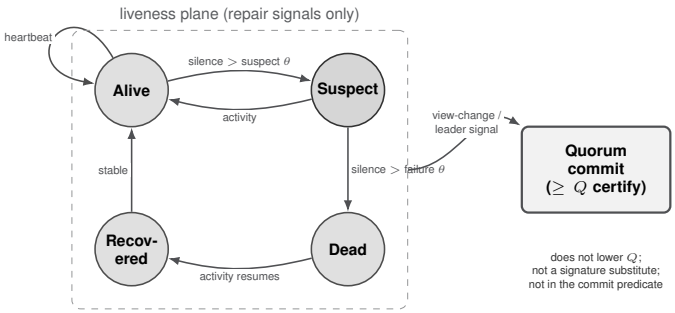

Fig. 3. The four-state liveness classifier (Table I) is a self-healing repair plane *around* the quorum path. It can request a view change or change leader selection, but by Lemma 3 it never lowers Q, never replaces signatures, and never enters the commit predicate.

it is marked dead. When the current leader is marked dead, view change fires immediately rather than waiting for the per-phase timeout $T_{\text{phase}} = 0.75\,\text{s}$ to expire across a full round. This is the difference from reactive protocols, which only begin recovery once a round timeout lapses. Leader failover is therefore bounded by

$$T_{\text{failover}} \;\leq\; T_{\text{dead}} + \Delta_{vc} + T_{\text{phase}},$$

where Δ_{vc} is view-change propagation time; with $T_{\text{dead}} = 2.3\,\text{s}$ and $\Delta_{vc} \approx 0.7$–$1.0\,\text{s}$, failover is about $3.3\,\text{s}$, as observed consistently across repeated lab leader-kill tests. None of these signals lowers Q or enters the commit predicate (Lemma 3): the heartbeat plane decides *when* to change leaders, never *whether* a block is safe to commit.

C. Human-Scale Time

Digital economies need time that machines can process and people can understand. A block cadence near human interaction time is useful for wallet payments, asset transfers, device checkpoints, AI-agent commitments, robotic handoffs, and identity verification events.

NABD block time records quorum acceptance of an ordered event at chain time. Physical-world assurance is supplied by device security, biometric capture quality, oracle policy, and governance. The chain supplies the final ordering layer for those external signals.

VIII. Measurement Methodology and Metrics

NABD measures finality from the perspective of a wallet or application, not from an internal validator timer. The primary latency metric is the elapsed time from signed transaction submission at public ingress to hash-confirmed status through `/api/v1/transactions/:hash`. The metric captures load balancing, client relay, validator mempool admission, quorum commit, and public read visibility.

In a bounded public-ingress operating sample, hash-confirmed transfer latency remained inside a human-interaction

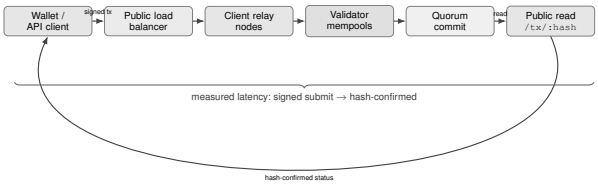

Fig. 4. Public-ingress measurement path. Latency is timed end to end from a client's signed submission to hash-confirmed visibility through `/api/v1/transactions/:hash`, so it captures load balancing, relay, mempool admission, quorum commit, and public read visibility rather than an internal validator timer.

TABLE II
PROTOCOL METRICS USED FOR EVALUATION

Metric	Definition
Hash finality	Signed submit to confirmed transaction hash
Block freshness	Age of latest committed block
Quorum availability	Healthy validators over active set size
Peer completeness	Validator peers over expected mesh size
Queue drain	Mempool and relay queues after smoke traffic
Chain identity	Network identifier, genesis hash, fingerprint

envelope. Read the sample as operating evidence for the measurement path, not as a long-horizon service-level objective.

IX. EVALUATION EVIDENCE

Evaluate NabdBFT against established deterministic BFT families. Its defining difference is *when* a failed leader is detected: classical protocols wait for a round timeout to expire (reactive), while NabdBFT's heartbeat subsystem detects a dead leader within $T_{\text{dead}} = 2.3\,\text{s}$ and triggers view change pre-emptively. Table V summarizes the comparison.

Failover figures for PBFT, Tendermint, and HotStuff are the ranges reported in their published evaluations [1], [4], [3], not same-environment reimplementations; the NabdBFT figure is from repeated lab leader-kill tests. NabdBFT shares the $O(N^2)$ messaging of PBFT and Tendermint, which is negligible for the target permissioned-to-sovereign scale (N in the single to low double digits) and to which threshold-signature aggregation could later be applied. The current evidence consists of a live public-ingress finality sample, target-VM signature microbenchmarks, a controlled $N = 8$ single-wallet ingress check, implementation-level tests, and historical throughput artifacts. The paper does not claim baseline throughput for PBFT, HotStuff, or Tendermint from this deployment.

The quorum rule $Q = \lfloor 2N/3 \rfloor + 1$ gives different headroom at different validator-set sizes (Table VI). As with any BFT system, finality tolerates up to f validator failures and pauses beyond that until the active set is restored. Larger validator sets and one-validator-per-host placement add spare capacity above the quorum threshold, and are natural growth paths as the network scales.

TABLE III
AGGREGATE PUBLIC-INGRESS FINALITY SAMPLE

Metric	Observed value
Samples	30 transactions
Mean latency	1.214 s (post-fix tx-gossip)
Median (p50)	1.197 s
95th percentile (p95)	1.436 s
Maximum latency	2.179 s
Confirmation	Transaction-hash lookup

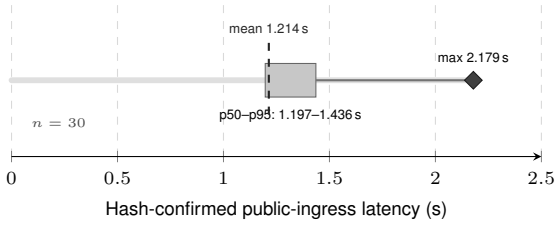

Fig. 5. Post-optimization profile of the hash-confirmed public-ingress finality sample (Table III): $n = 30$, with the p50–p95 inter-percentile band shaded, the mean dashed, and the maximum at the whisker. A transaction-gossip fix deployed on 2026-06-06 reduced mean finality from 3.091 s to 1.214 s; the residual latency is batch-validation overhead on under-provisioned validator containers, not protocol-inherent. Read this as operating evidence for the measurement path on the current testnet, not a long-horizon service-level objective.

A. Current Operating Topology

NABD runs a live public testnet at `testnet.nabdchain.org`. The current operating sample uses eight validators. Four Hetzner CPX22 virtual machines (2 shared vCPU, 4 GB RAM each, AMD EPYC) run two validators each, and one additional virtual machine hosts clients and explorer services outside the consensus quorum. The validator machines are distributed across three Hetzner datacenters in three distinct locations, Germany, Helsinki, and Amsterdam, so that host-, rack-, and site-level faults are geographically decorrelated. For $N = 8$, the quorum rule gives $Q = \lfloor 16/3 \rfloor + 1 = 6$, so the network continues through the loss of any single validator virtual machine (six validators remain available for quorum). Operating across multiple cloud providers is a planned extension.

This configuration is a deliberate floor, not a recommended ceiling. Each validator runs on a Hetzner CPX22 instance, two *shared* vCPUs and 4 GB of RAM, and two validators share a single VM, so the active set is squeezed onto modest, contended hardware with no headroom. The finality figures reported here are therefore a lower bound: they are what NabdBFT delivers at its most constrained, and dedicated CPUs, more memory, one-validator-per-host placement, and faster storage can only improve tail latency and throughput. Running a full deterministic Layer-1 (ten nodes, load balancers, and an explorer) on this class of hardware is itself the point: low-cost sovereignty means a community can own its settlement layer instead of renting trust.

TABLE IV
FINALITY LATENCY BY SCENARIO

Scenario	Setup	Mean	Min	Max	Note
User smoke test	2 TX, parallel	1.70 s	1.47 s	2.00 s	Manual rapid submit
30-TX spaced (lab VM)	30 TX, 2 s apart	1.21 s	0.96 s	2.18 s	Spaced submission

TABLE V
PROTOCOL COMPARISON SUMMARY

Protocol	Msg. compl.	Phases	Detection	Failover
PBFT [1]	$O(N^2)$	3	Reactive	10–30 s
Tendermint [4]	$O(N^2)$	3	Reactive	3–10 s
HotStuff [3]	$O(N)$	3	Reactive	5–15 s
NabdBFT	$O(N^2)$	4	**Proactive**	≈ 3.3 s

X. LEDGER, REPLAY PROTECTION, AND ASSETS

NABD uses account-based state with NABD1 account addresses. Transactions are signed over canonical payloads so validators verify the same meaning. Blocks are protocol-versioned, and asset state participates in deterministic ledger state.

Native asset operations include creation, minting, transfer, and burn. Fungible assets represent community money, credits, points, quotas, or regulated asset units. Unique units represent certificates, tickets, access passes, inventory receipts, machine identities, or claims on physical objects.

A. Transaction Validation and Replay Protection

Transaction validation binds the signed payload to account sequence and chain identity. Validators verify canonical transaction meaning, account nonce, sufficient balance, transaction type, fee policy, asset rules, and signature scheme. A previously accepted nonce cannot be replayed for the same account.

Cross-chain replay is blocked by signing the network identifier and genesis hash into the protocol payload. The transaction hash is derived from the signed payload, and validators reject mismatched chain identity. A valid signature from one NABD network has no standing on another network or on a fork with a different genesis identity.

XI. HUMAN AUTHORIZATION, AUTONOMOUS PROGRAMS, AND PRIVACY

A. Biometric Human Authorization

In many blockchain systems, biometrics unlock a local phone or laptop while the chain still sees an ordinary software key. If the device is lost, stolen, destroyed, or electrically damaged, the human identity behind the account strands.

NABD treats biometric proof as protocol-level authorization. The body remains off-chain. A high-quality multi-layer palm-vein scanner captures the biometric signal inside secure hardware or a trusted oracle, where template matching and liveness detection are performed. *Validators never see or match biometric data*: they verify only the transaction signature bound

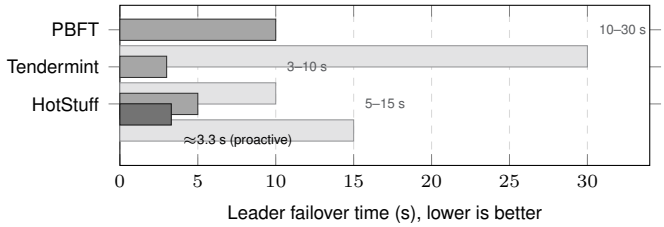

Fig. 6. Leader-failover time: NabdBFT's proactive heartbeat detection yields a consistent ≈3.3 s recovery, versus the wide timeout-dependent ranges of reactive protocols (published figures [1], [4], [3]). Light bars show the worst-case end of each reported range.

TABLE VI
QUORUM RESILIENCE ACROSS NETWORK SIZES

N	f	Q	After 1 failure	After 2 failures
5	1	4	4/4 exact quorum	3/4 halted
7	2	5	6/5 one spare	5/5 exact quorum
8	2	6	7/6 one spare	6/6 exact quorum
9	2	7	8/7 one spare	7/7 exact quorum
11	3	8	10/8 two spare	9/8 one spare

to the off-chain attestation, never a biometric template. The chain receives only commitments, hashes, attestations, public keys, and signed policy results.

Human authorization is not limited to a single emergency mode. The same mechanism lets a community require live biometric approval across a spectrum of high-value actions: authorizing treasury and asset movements, approving governance changes and program actions, gating physical access, and, at the high-risk extreme, acting as an emergency recovery circuit breaker for account recovery, suspension, or program shutdown. In every case the protocol keeps live human authorization at the boundary and raw biometric data outside public chain state.

B. Security Considerations

Biometric attestations must be transaction-bound. A valid attestation includes a fresh nonce or action hash, the target transaction or program action, a policy identifier, an expiry bound, and device or oracle attestation metadata. Replay fails when the nonce is consumed, the transaction hash changes, or the expiry passes.

A single trusted device is not a universal trust root. Higher-assurance deployments can require M-of-N biometric oracles, independent device attestations, governance approval, or a recovery delay before critical actions execute. If biometric infrastructure is unavailable, biometric-gated actions should fail closed or pause according to policy while ordinary non-gated transactions continue. Palm-vein liveness checking remains inside the secure capture boundary, and biometric templates require revocation and replacement policy outside public ledger state.

C. Biometric-Gated Autonomous Programs

A central application of biometric authorization is autonomous program policy. Biometric-only accounts are possible

TABLE VII
BENCHMARK EVIDENCE USED IN THIS REVISION

Benchmark	Evidence	Interpretation
Signature microbenchmarks	Four Hetzner CPX22 validator VMs (2 shared vCPU, 4 GB RAM, AMD EPYC), production image path rebuilt in disposable containers: Ed25519 sign mean 0.030 ms, verify 0.051 ms; Falcon-512 sign mean 0.198 ms, verify 0.039 ms; hybrid verify mean 0.091 ms, p95 0.109 ms	Hybrid verification is sub-millisecond on the target validator CPU class. The Ed25519 figure reflects a pure-Perl path and is not directly comparable to the native Falcon-512 timing.
Public-ingress finality	Live $N = 8$ run on validator 1 with two remote observer validators: 30 submitted, 30 confirmed; mean 1.214 s, p50 1.197 s, p95 1.436 s, max 2.179 s (after the 2026-06-06 tx-gossip fix; pre-fix mean was 3.091 s)	Confirms user-visible finality on the current Hetzner topology under non-saturation load.
Single-wallet ingress check	Live $N = 8$ controlled run, one funded wallet, fixed batch size 100, live fee floor 1000, direct validator ingress: 4700 submitted, 4700 confirmed, zero failed submissions, 130 confirmed TPS over 36.1 s, 84 committed blocks	Establishes clean acceptance and confirmation under ordered single-sender load. It is not a saturation ceiling.
Historical throughput artifact	Earlier $N = 5$ saturation rows: 597 to 1152 confirmed TPS, mean 936 TPS	Retained only as historical capacity evidence from a prior topology. Current claims use the $N = 8$ run above.

Method note: The Ed25519 and Falcon measurements are implementation-path timings. Ed25519 signing uses the wallet-compatible `Crypt::Ed25519` path, while Falcon-512 uses native XS/liboqs. The live leader-kill recovery run and CERTIFY A/B latency experiment are not reported in this revision, and no claim depends on them.

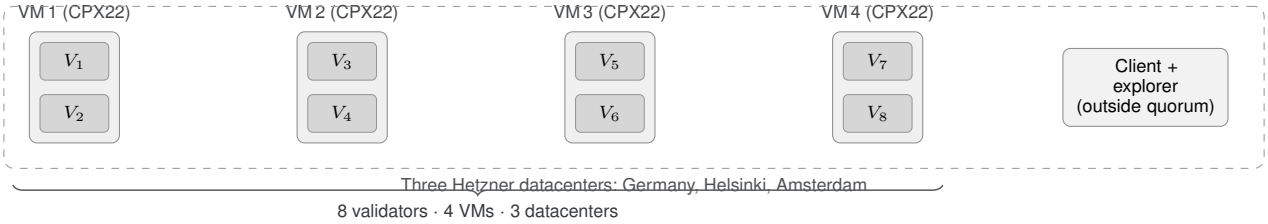

Fig. 7. Current $N = 8$ operating topology on the public testnet (`testnet.nabdchain.org`). Two validators share each VM, so $Q = 6$ tolerates the loss of any single validator VM (six validators remain). The validator VMs are distributed across three Hetzner datacenters in three distinct locations (Germany, Helsinki, and Amsterdam), geographically decorrelating site-level faults; operating across multiple providers is a planned extension.

TABLE VIII
VALIDATOR DEPLOYMENT MAPPING

VM	Datacenter	Spec	Validators / role
VM 1	Germany	CPX22 (2 vCPU, 4 GB)	V_1, V_2
VM 2	Helsinki	CPX22 (2 vCPU, 4 GB)	V_3, V_4
VM 3	Amsterdam	CPX22 (2 vCPU, 4 GB)	V_5, V_6
VM 4	Germany	CPX22 (2 vCPU, 4 GB)	V_7, V_8
VM 5	Germany	4 vCPU (client tier)	Client + explorer (outside quorum)

Public addresses are omitted here and retained in an internal audit appendix.

in controlled deployments, while autonomous programs provide the broader programmable pattern.

In the current stack, autonomous programs execute as deterministic WASM3 modules through the XS interpreter path rather than native-code JIT. The Perl host layer enforces the manifest, owner, code hash, ABI hash, capabilities, storage limits, fee limits, event limits, return-size limits, timeout, and upgrade/freeze policy. Deployments and calls are signed transactions, so normal nonce, replay, chain-identity, and fee checks apply before execution.

The execution model is intentionally narrow. Programs interact with ledger state through a bounded host API, including balance lookup, transfer, storage read/write, and state recording. High-assurance deployments can require a policy grade in which machine signatures and a human biometric signature are both present before deployment or before a critical call crosses its configured boundary.

An autonomous program can run normal rules automatically while reserving high-risk actions for biometric authorization. Examples include:

- a community treasury that releases emergency funds only after approved human biometric signatures;
- a robot controller that requires human presence before changing safety limits;
- a public-service workflow that requires a live person before issuing a credential;
- an asset autonomous program that pauses minting, burning, or transfer during a risk event until biometric approval is provided;
- a recovery program that restores access when ordinary devices or keys are lost.

Human consent becomes part of the execution policy itself. The program verifies whether the protocol has received the required biometric authorization evidence before crossing a configured critical boundary.

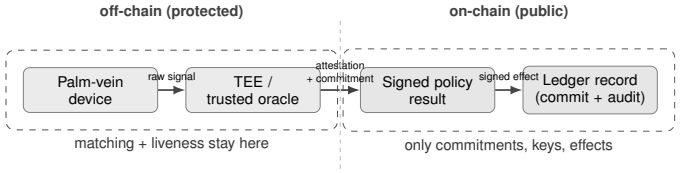

Fig. 8. Biometric privacy boundary. Palm-vein capture, matching, and liveness detection stay inside the off-chain secure boundary; only commitments, attestations, public keys, and signed policy effects cross to the public ledger. The raw biometric template never appears in chain state.

D. Privacy

Biometric systems require strict data minimization. NABD follows a simple rule: raw body data belongs outside chain state. Public surfaces show what users and communities need to verify state, while sensitive payloads, diagnostics, secrets, raw state diffs, and unnecessary proof material stay hidden by default.

Privacy in this model comes from separation: biometric capture and matching stay in protected environments, while the chain records commitments and signed results. Public accountability is preserved without turning the ledger into a biometric database.

XII. Post-Quantum-Ready Authentication

NABD uses a scheme-aware signature envelope so transaction authorization can evolve across cryptographic algorithms. The same transaction meaning can be authenticated under Ed25519, Falcon-512, or a hybrid policy. The envelope keeps the ledger model stable while allowing stronger signature requirements where a community or institution chooses them.

Falcon-512 support is active in the implementation through the `signatures[]` envelope. Hybrid transactions can carry a Falcon-512 signature and an Ed25519 signature over the same canonical payload, preserving legacy compatibility while allowing post-quantum-ready policy. Repository tests verify Falcon-512 key generation, Falcon-only transaction signing, and hybrid Ed25519 plus Falcon transaction signing.

Naming and status are kept precise. *FN-DSA* is the standard family name under the forthcoming FIPS 206; *Falcon-512* is the concrete liboqs parameter set the runtime actually executes. Accordingly, NABD claims to be *post-quantum ready*, not FIPS 206 validated: it will not assert FIPS 206 compliance until the final FIPS 206 byte formats and known-answer tests (KATs) are adopted by the runtime.

Target-VM Falcon-512 and Ed25519 sign and verify timings on the CPX22 validator class are reported in Table VII; batch-verification throughput is a separate measurement tracked for hybrid-by-default operation.

XIII. Assurance Evidence and Scope

NABD separates claims into auditable evidence classes: formal safety models, deterministic validation tests, public ingress operating samples, release integrity gates, and durability checks. These evidence classes support the protocol properties stated in this paper and make clear which claims are architectural, which are measured, and which require deployment-specific governance.

TABLE IX
Signature Envelope Characteristics

Mode	Size evidence	Role
Ed25519	64 B signature	Compatibility and fast verification path
Falcon-512 (FN-DSA)	897 B public key; 648–662 B signature (mean 655.2 B, 1000 samples)	Post-quantum-ready authorization
Hybrid	Additive entries in `signatures[]`	Transition and high-assurance policy

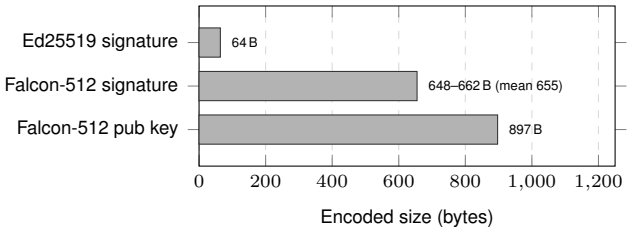

Fig. 9. Signature-envelope size evidence (Table IX). Falcon-512 (the FN-DSA parameter set being standardized under the forthcoming FIPS 206) is chosen for its compact post-quantum signatures relative to other lattice schemes, at the cost of a larger public key than Ed25519. A hybrid transaction carries both an Ed25519 and a Falcon-512 entry in the `signatures[]` array.

A. Formal Model Summary

The TLA+ model inventory covers the main safety model and a smaller view-reset regression model. On 2026-06-04, TLC 2.19 completed the $N = 4, F = 1$ baseline with $Q = 3$, `MaxRound=0`, `MaxView=1`, and two modeled block hashes. The run checked `TypeOK`, `Agreement`, `Validity`, `VotingUniqueness`, and `CorrectLeaderReproposal`. It completed without invariant violations after 2,611,170,154 generated states, 230,597,253 distinct states, depth 60, and zero states left on the queue.

The `CorrectLeaderReproposal` invariant is proposal-time based: a correct leader that observes lower-view lock evidence when it proposes must re-propose that locked hash. This avoids a false failure in the pool-free model where a lower-view certificate appears after a higher-view proposal and would otherwise retroactively invalidate an earlier correct proposal. The view-reset liveness model also completed with 42 generated states, 21 distinct states, and depth 9. The expected-fail reset diagnostic remains useful because it violates `NoSameViewReset` when the unsafe reset behavior is enabled.

These TLC runs are bounded model-checking evidence. They support the general proof sketch above but do not replace a theorem-prover result for all validator set sizes. The completed $N = 4, F = 1$ baseline was independently rerun (fp=64, seed 123456789), reproducing the same completion signature with

TABLE X
ASSURANCE EVIDENCE USED TO SUPPORT PROTOCOL CLAIMS

Evidence class	Claim supported
Formal model	Quorum intersection, view-scoped voting, and agreement safety
Consensus regression	No-reorg behavior, leader recovery, and fork-resilience
Transaction tests	Nonce, replay, overspend, batch, asset, and AP validation
Public ingress sample	End-to-end hash-confirmed finality path from user entry
Release gates	Chain identity, image content, dependency security, and config lock
Durability drill	Backup, restore, quick-check, and ledger-count consistency

TABLE XI
TLA+ EVIDENCE SUMMARY

Model	Result	Evidence
$N = 4, F = 1$ safety baseline	Complete pass	2.611B generated, 230.6M distinct, depth 60
$N = 4, F = 1$ rerun (fp=64)	Complete pass	collision risk 3.5×10^{-4}
View reset liveness	Complete pass	42 generated, 21 distinct, depth 9
View reset regression	Expected fail	`NoSameViewReset` diagnostic violation
$N = 7, F = 2$ sweep	Bounded partial	12.28B generated, 1.18B distinct, depth 13, disk exhausted

a reduced fingerprint collision-risk estimate of 3.5×10^{-4} (down from 1.4×10^{-2} in the original fp=37 run). A larger $N = 7, F = 2$ production-geometry sweep ran for ≈ 10.2 hours on a lab VM and explored 12.28 billion generated / 1.18 billion distinct states to depth 13 with zero invariant violations before terminating on *disk exhaustion*, with the state queue still growing (≈ 752 million states). This bounded partial run supports the claim that the production geometry admits no small-counterexample safety violation within the explored frontier; it is explicitly *not* exhaustive verification.

B. Scope Boundaries

The present implementation does not claim deployed cryptographic confidential transfers, validator-side biometric matching, ISO 20022 payment-network compliance, or root-to-tenant checkpoint anchoring. The privacy model in this paper is data minimization, disclosure control, and off-chain biometric matching with on-chain commitments and policy effects. Confidential transfer circuits and cross-network anchoring are compatible extension paths, but they are not needed for the deterministic finality and human-authorization model described here.

XIV. OPERATING MODEL AND GOVERNANCE

A. Low-Cost Sovereignty

A useful layer one can operate without enormous infrastructure. NABD is intended for modest, reproducible operation: explicit genesis state, deterministic serialization, conservative services, clear validator roles, and plain operational surfaces. A community, company, school, cooperative, or public-good network can own its settlement layer instead of renting trust from a single platform.

B. Long-Lived Operation

The protocol favors components that are understandable and replayable: canonical payloads, deterministic state, explicit configuration, small trusted surfaces, and reproducible release artifacts. Long-lived systems survive by being rebuildable and understandable rather than relying on opaque infrastructure dependencies.

C. Governance

Validator membership, biometric policy, asset authority, and emergency actions are governance questions. NABD makes those questions explicit. A network can define who proposes changes, who approves them, when they activate, and which authorization proofs are accepted for critical actions.

XV. CONCLUSION

NABD's central idea is to make liveness proactive. By decoupling validator health monitoring from the consensus round, a dead leader is detected within $T_{\text{dead}} = 2.3\,\text{s}$ and replaced in about $3.3\,\text{s}$, three to ten times faster than the reactive, post-timeout recovery of PBFT, Tendermint, and HotStuff, while a strict four-phase quorum keeps finality deterministic at a mean of $1.214\,\text{s}$ on modest two-vCPU shared hardware. On that base, NABD adds what a sovereign settlement layer actually needs: off-chain biometric human authorization for treasury, governance, and recovery; native multi-asset state and deterministic autonomous programs; and a post-quantum-ready signature envelope. The result is a ledger that recovers like a live system, finalizes like a settlement system, and answers to people where the stakes are high, running today at `testnet.nabdchain.org`.

REFERENCES

[1] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, 1999.

[2] A. Bessani, J. Sousa, and E. E. P. Alchieri, "State Machine Replication for the Masses with BFT-SMaRt," in *Proceedings of DSN*, 2014.

[3] M. Yin, D. Malkhi, M. Reiter, G. G. Gueta, and I. Abraham, "HotStuff: BFT Consensus with Linearity and Responsiveness," in *PODC*, 2019.

[4] E. Buchman, J. Kwon, and Z. Milosevic, "The latest gossip on BFT consensus," arXiv:1807.04938, 2018.

[5] T. D. Chandra and S. Toueg, "Unreliable Failure Detectors for Reliable Distributed Systems," *Journal of the ACM*, vol. 43, no. 2, 1996.

[6] L. Lamport, *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley, 2002.

[7] Y. Yu, P. Manolios, and L. Lamport, "Model Checking TLA+ Specifications," in *CHARME*, 1999.

[8] E. Androulaki et al., "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *EuroSys*, 2018.

[9] Hyperledger Fabric Documentation, “Private Data,” Linux Foundation, accessed 2026.
[10] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, “Algorand: Scaling Byzantine Agreements for Cryptocurrencies,” in *SOSP*, 2017.
[11] S. Goldberg, L. Reyzin, D. Papadopoulos, and J. Vcelak, “Verifiable Random Functions (VRFs),” RFC 9381, 2023.
[12] P.-A. Fouque et al., “Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU,” NIST Post-Quantum Cryptography Standardization Project.
[13] National Institute of Standards and Technology, “Post-Quantum Cryptography Standardization,” NIST Computer Security Resource Center.
[14] National Institute of Standards and Technology, “Module-Lattice-Based Digital Signature Standard,” FIPS 204, 2024.
[15] R. Perlner et al., “Status Update on the FN-DSA Standard (Draft FIPS 206),” National Institute of Standards and Technology, PQC status update, 2025.
[16] N. K. Ratha, J. H. Connell, and R. M. Bolle, “Enhancing Security and Privacy in Biometrics-Based Authentication Systems,” *IBM Systems Journal*, vol. 40, no. 3, 2001.
[17] ISO/IEC 24745:2022, “Information Security, Cybersecurity and Privacy Protection: Biometric Information Protection,” International Organization for Standardization, 2022.
[18] A. Haas et al., “Bringing the Web up to Speed with WebAssembly,” in *PLDI*, 2017.