V1 dynamic replica set, expansion triggers, shadow attestation, transport choices, and mobile participation

Date: 2026-05-25 **Status:** Fifth iteration on the replica + tokenomics design surface. Incorporates four James-driven design decisions and refinements 2026-05-25: (1) V1 must support dynamic replica-set growth from day one (no waiting for V2); (2) explicit rubric for when the set should grow; (3) shadow attestation as the security service inactive bonded replicas perform to earn rewards + promotion priority; (4) clarifications on Tor-vs-clearnet transport choices; (5) speculative-but-promising mobile shadow-attestor tier. **Scope:** Plain-English explanation of all five + technical detail on manifest mechanism, expansion governance, shadow-attestation security model, slashing bounty mechanics, transport options, mobile architecture, revised reward economics. **Companion to:** `tokenomics-research-2026-05-25.md`, `tokenomics-option-e-detail-2026-05-25.md`, `replicas-detail-2026-05-25.md`, `replicas-open-set-and-halving-2026-05-25.md`. Read those for the context this document refines.

> ■■ ***Alignment note (2026-06-28) — read first.*** *This is a dated iteration record (companion to the other replica/tokenomics papers), superseded in framing by* `replica-system-2026-05-25.md` *and the whitepaper. Two later decisions:* ***(1)*** *the 5,000 XMR mint-fee cap was removed entirely (D-013) — no post-cap phase;* ***(2)*** *the epoch model is now two-cadence — a* ***registry epoch*** *of 7 days (snapshots) and a separate* ***attestation epoch*** *of ≈1 hour (attestation + emission). Canonical sources:* `DECISIONS.md` *(D-013) and the whitepaper (§9, §11).*

---

0. What's new in this document

Five design refinements + clarifications relative to the prior four docs:

1. **The active set is dynamic from V1**, not just V2. Prior doc proposed SDK-pinned V1 + on-chain V2; this doc replaces that with a unified Foundation-signed manifest mechanism that's identical in V1 and V2 (only the signing authority changes).
2. **Concrete trigger rubric for when to expand the active set.** Seven quantitative signals, each independently triggering a Foundation expansion review.
3. **Shadow attestation as the security service for the bonded-but-inactive pool.** Inactive replicas run the protocol in parallel, sign their own non-quorum-counting snapshots, and serve as a defense-in-depth divergence check + slashing-evidence collector. This justifies a small token reward (~10–15% of active rate) for real work rendered — replaces my prior "no reward, bond-age priority only" recommendation.
4. **Tor is recommended, not required.** Operators can run replicas on clearnet, Tor onion, or both. Best practice is dual-stack so all operator transport choices are supported.
5. **Mobile shadow attestors are technically viable** for the inactive pool (not the active set). Opens participation to anyone with a phone + wallet bond. Speculative tier; concrete architecture sketched.

Part 1 — Plain English

1. The dynamic replica set, available from V1

The previous design proposal said V1 ships with 5 SDK-pinned replicas, and V2 introduces an on-chain registry contract that supports dynamic growth. That's lazy — it delays the right design and complicates the V1 → V2 transition.

The cleaner approach: **V1 ships with a Foundation-signed manifest mechanism that's identical in V1 and V2.** The only thing that changes between versions is who signs the manifest (Foundation key in V1; eventually an on-chain registry contract in V2). The protocol shape doesn't change.

In practice this means:

- The SDK ships with a **bootstrap manifest** containing the initial 5 replica pubkeys + Foundation's signing key.
- When the active set expands (5 → 7, then 7 → 9, etc.), Foundation publishes an updated manifest.
- Clients fetch the new manifest periodically (cached, signed, verified against Foundation's key).
- The protocol's quorum threshold is derived from the manifest's set size: `floor(N/2) + 1`. It scales automatically as N grows.
- No SDK release required for set expansion. Foundation signs a new manifest; clients pick it up at next refresh.

The result: V1 replicas can scale 5 → 7 → 9 → 13 → arbitrary N without anyone needing to update their SDK. Operators bond, Foundation expands the active set, clients adapt automatically.

2. When to expand the set — the trigger rubric

Foundation can't expand arbitrarily — that's centralization without accountability. So there has to be a rubric: published, transparent triggers that motivate expansion + the reasoning every time it happens.

Seven signals, each independently triggering a Foundation expansion review:

Signal	Threshold	Why it matters
Hosting-provider concentration	≥2 of N replicas on the same hosting provider	Correlated downtime risk; single-vendor compromise risk
Geographic concentration	≥3 of N replicas in the same legal jurisdiction	Coordinated regulatory action risk
Quorum margin	A single replica outage would put us within 1 vote of the quorum threshold	Operational margin too thin; no headroom for surprises
Bonded pool depth	Bonded pool ≥2× current active-set size, with qualified candidates	Wasted operator commitment; expansion absorbs committed standby capacity
Marketplace growth	Registration volume up >50% year-over-year	Network maturity warrants more independent attestors
Mandatory review	Every 12 months regardless of other signals	Prevents stagnation; ensures the rubric is exercised regularly
Security incident	Any detected attack attempt, near-miss, or attested compromise	Reactive expansion to raise attack cost

When any of these crosses its threshold, Foundation publishes an **expansion proposal**: the triggering signal, the proposed new set size, the candidate operators selected from the bonded pool (with selection rationale), and the new quorum threshold. Token holders vote on the proposal under Foundation governance. The reasoning is on-record and auditable.

This avoids both "Foundation expands when convenient" (no rubric) and "Foundation can never expand" (rigid threshold). The triggers are objective; the response is governed.

3. Shadow attestation — what inactive bonded replicas actually do

In the prior doc I struggled with the bonded-pool reward question. James's "what could inactive replicas do?" reframed it perfectly: **the inactive pool runs a parallel protocol that provides defense-in-depth security, and the reward is for the security service performed**, not for waiting.

Here's how it works in plain language:

A shadow attestor is a replica that does everything an active replica does — accepts submissions, runs gossip, builds snapshots at epoch boundaries, signs them with its own key — **except its signature doesn't count toward the 3-of-N quorum.** Its signature is a parallel attestation that clients can compare against the active replicas' signatures.

If a shadow attestor's snapshot signature matches what the active replicas signed, everyone's honest. If it diverges, something is wrong — either an active replica is misbehaving (forging snapshots) or the shadow is broken/malicious. The divergence is detectable, attributable, and slashable.

This adds **independent triangulation** beyond the 3-of-N quorum. The transparency log already detects divergence over time; shadow attestation detects divergence in the moment a snapshot is published.

Concretely, an inactive bonded replica earns rewards for two things:

1. **Continuous shadow attestation service.** They publish signed shadow snapshots that match the active replicas' snapshots. Peers + clients can verify they're online and producing valid attestations. Token reward provisionally 10–15% of active rate, paid only when shadow service is verifiably rendered.
2. **Slashing-evidence bounties.** If a shadow attestor detects an active replica's signature diverges from the consensus shadow attestation, they can submit provable evidence (the divergent signature + the consensus signature). If confirmed, they earn a portion of the slashed active replica's bond — provisionally 20% of slashed amount.

This aligns incentives strongly: the inactive pool has direct economic reason to actively watch active replicas. It's a watchdog with skin in the game.

4. Tor — recommended default, not protocol requirement

Throughout the prior docs I've been saying "operators submit over Tor" reflexively. **It's worth being explicit: the protocol doesn't require Tor.** The §22 session-binding handshake and the §33 PaymentCommitment flow are protocol-cryptographic — they work over any transport.

Tor is the recommended default for two reasons:

- **Operator privacy.** An operator registering an agent over clearnet leaks their IP address to the replica accepting the submission. Over Tor onion, no IP leak.
- **Replica privacy + DDoS resistance.** A replica on a fixed clearnet IP is a DDoS target and a jurisdictional pressure target. A Tor onion service is harder to attack.

But operators can choose: clearnet HTTP, Tor onion, or both. Best practice is dual-stack — every replica offers a Tor onion address AND a clearnet endpoint. That way:

- Privacy-conscious operators can submit over Tor.
- Non-privacy-sensitive operators (or those without Tor capability — a hosted-on-mobile use case) can submit over clearnet.
- Replicas reach each other over whichever transport works.

The 3-of-N quorum doesn't care about transport. A clearnet replica's signature is just as cryptographically valid as a Tor replica's. **Mixed-transport replica sets are fine.**

5. Mobile participation — viable for shadow attestors, not active replicas

The active-replica role has hard requirements that mobile devices struggle with: always-online HTTP listener, ability to accept incoming connections (mobile carriers usually NAT), persistent state across reboots, gossip with peers every 30 seconds, sub-minute response times. A mobile phone running an active replica would be unreliable.

But the shadow attestor role has much weaker requirements: pull state from active replicas periodically, sign occasionally, upload signature. **No incoming connections needed.** Intermittent connectivity is fine. Battery-friendly. Could be packaged as a mobile app.

A practical mobile shadow attestor:

- Downloads as a mobile app (iOS / Android).
- User connects a wallet, bonds tokens (smaller bond than full replicas — provisionally 10k tokens vs 100k for active).
- App runs in background when on wifi, charging. Pulls snapshots from active replicas, signs its own, uploads.
- Earns the shadow-attestation reward when service is verifiably rendered.
- If user goes offline for >24 hours, the app pauses shadow attestation and the reward stops.

This is **speculative** — James flagged it as a "just a thought, not a design request" — but it's worth taking seriously. If implemented, it dramatically lowers the participation bar and grows the bonded pool's geographic + jurisdictional + demographic diversity. A bonded pool with 10,000 mobile shadow attestors across 100 countries is qualitatively different from a pool of 50 server operators in 10 countries.

The honest caveats: mobile shadow attestors are harder to vet, may have lower up-time reliability, and need careful wallet-integration design. v2-or-later candidate; not v1 launch. But the protocol design above (manifest mechanism + shadow attestation) **supports it natively without further protocol changes.** Mobile is a deployment story, not a protocol redesign.

Part 2 — Technical detail

---

6. The signed manifest mechanism

6.1 The manifest data structure

```
pub struct ReplicaSetManifestBody {
   pub version: u32,                          // Monotonic; increments on each manifest update
   pub published_at_unix_seconds: u64,
   pub valid_through_unix_seconds: u64,       // Manifest expiry; clients refuse stale manifests
   pub active_replicas: Vec<ActiveReplicaEntry>,   // sorted by replica_id
   pub shadow_attestors: Vec<ShadowAttestorEntry>, // sorted by attestor_id
   pub quorum_threshold: u32,                 // = floor(active_replicas.len() / 2) + 1
   pub min_active_bond: u64,                  // in atomic token units
   pub min_shadow_bond: u64,                  // smaller for shadow tier
   pub current_halving_epoch: u32,            // Which halving regime we're in
}

pub struct ActiveReplicaEntry {
   pub replica_id: [u8; 32],
   pub pubkey: [u8; 32],
   pub onion_address: Option<String>,         // Tor v3 onion, if available
   pub clearnet_endpoint: Option<String>,     // HTTPS URL, if available
   pub jurisdiction_code: Option<String>,     // ISO 3166-1 alpha-2; informational
   pub bonded_at_unix_seconds: u64,           // For seniority tracking
}

pub struct ShadowAttestorEntry {
   pub attestor_id: [u8; 32],
   pub pubkey: [u8; 32],
   pub bonded_at_unix_seconds: u64,
}

pub struct SignedReplicaSetManifest {
   pub body_bytes: Vec<u8>,                   // canonical encoding of body, verbatim
   pub foundation_signature: [u8; 64],        // Ed25519 signature on manifest_hash
}

manifest_hash = SHA-256("nightshade-replica-manifest-v1" || 0x00 || body_bytes)
```

Each entry must validate `(onion_address.is_some() || clearnet_endpoint.is_some())` — at least one transport. Operators can choose either or both. The protocol enforces no transport requirement.

6.2 Manifest verification by clients

When a client receives a snapshot:

1. Client has a cached `SignedReplicaSetManifest` from a recent refresh.
2. Client verifies `foundation_signature` against the SDK-pinned Foundation pubkey.
3. Client checks `now < valid_through_unix_seconds` (manifest not expired).
4. Client uses `quorum_threshold` and `active_replicas` to verify the snapshot's `replica_signatures` — at least `quorum_threshold` distinct active replicas signed the snapshot's `snapshot_hash`.

Manifest is fetched from a well-known Foundation-published endpoint (HTTPS + Tor mirror). Caching policy: respect `valid_through_unix_seconds`, refresh on miss or expiry. New SDK installs ship with a bootstrap manifest signed at SDK release time.

6.3 Manifest update flow

When Foundation expands the active set:

1. Foundation publishes a proposal explaining the trigger (which §7 criterion fired), the selected new active replicas (drawn from the bonded pool with selection rationale), the new `quorum_threshold`, and the proposed `valid_through_unix_seconds`.
2. Token holders vote under Foundation governance.
3. On approval, Foundation signs the new manifest.
4. Foundation publishes the new manifest to its well-known endpoint.
5. Clients refresh on next polling cycle (or on cache expiry).
6. Snapshots signed by the new replica set are now valid; snapshots signed by the old set with version `< N+1` remain valid until their own expiry.

There's a clean migration window — old clients can verify until their cached manifest expires, then they refresh and get the new set. No flag day.

6.4 V2 transition: Foundation key → on-chain registry contract

V2 introduces an on-chain registry contract on the token's settlement chain. The contract holds the canonical manifest state. The Foundation publishes signed manifests by writing to the contract (instead of to a Foundation HTTPS endpoint). Clients query the contract for the current manifest.

The manifest data structure doesn't change. The verification flow doesn't change. Only the **signing authority + publication endpoint** change. This is a clean V1 → V2 transition that doesn't require redesigning the protocol — only swapping the manifest publishing mechanism.

7. The expansion trigger rubric

7.1 Quantitative thresholds for review

Foundation reviews expansion eligibility every quarter. Any of the following crossing its threshold automatically triggers a formal expansion proposal:

Signal	Computed	Threshold	Action
Hosting-provider concentration	`max(count by provider) / N`	≥ 0.40 (≥2 of 5, or ≥3 of 7)	Propose expansion to add operators on under-represented providers
Geographic concentration	`max(count by jurisdiction) / N`	≥ 0.60 (≥3 of 5, ≥5 of 7)	Propose expansion to add operators in under-represented jurisdictions
Quorum margin	`N - quorum_threshold` (slots that can fail)	< 2 (less than 2 slots can fail)	Propose expansion to increase headroom
Bonded pool depth	`pool_size / N`	≥ 2.0	Propose expansion to absorb committed standbys
Marketplace growth	YoY registration volume change	> 50%	Propose capacity-driven expansion
Mandatory review	Time since last expansion	> 12 months	Mandatory review regardless of other signals
Security incident	Any of: transparency-log inconsistency detected, attestation divergence detected, compromise alleged	Any single occurrence	Immediate emergency-review (governance timeline shortened)

7.2 The proposal + vote process

When a trigger fires:

1. Foundation publishes the **expansion proposal** within 30 days: - Triggering signal + supporting data - Proposed new set size (typically +2; can be +1 or +4 if justified) - Selected candidates from the bonded pool, ranked by: bond size, bond age, attestation history, jurisdictional diversity contribution, hosting diversity contribution - New `quorum_threshold` - Proposed `valid_through_unix_seconds` for the

new manifest

2. Token holders have **14 days** to vote.
3. Approval requires **>50% of votes cast** AND **>20% of circulating supply participating**.
4. If approved, Foundation signs and publishes the new manifest within 7 days.
5. If rejected, the trigger remains active and Foundation must propose again within 60 days with adjusted reasoning.

Quorum requirements prevent low-turnout approvals; the 60-day re-proposal requirement prevents indefinite stalling.

7.3 Emergency expansion path

Security incidents bypass the normal cadence:

- Trigger: attestation divergence, attempted compromise, single-replica failure when quorum margin was already at minimum.
- Foundation can publish an emergency manifest update **within 24 hours** with reasoning.
- Token holders have **7 days** to ratify retroactively.
- If retroactive ratification fails, the emergency update is rolled back to the pre-emergency manifest at next refresh cycle.

This is the equivalent of the spec §24.4 emergency overlay — fast response with retroactive accountability.

8. Shadow attestation — technical mechanics

8.1 What shadow attestors run

A shadow attestor runs the same `nightshade-replica` binary as active replicas. The only difference is **whether it's in the manifest's `active_replicas` list (full participation) or `shadow_attestors` list (parallel attestation only)**.

Both lists' members: - Accept submissions over their published transport (Tor onion / clearnet / both). - Run anti-entropy gossip with all peers in both lists (active + shadow attestors gossip together). - Build deterministic snapshots at epoch boundaries from the same input set. - Sign with their own Ed25519 key. - Publish signed snapshots to a well-known endpoint per attestor.

The differences: - **Active replicas' signatures count toward the 3-of-N quorum** clients use to trust snapshots. - **Shadow attestors' signatures don't count toward quorum**. They're verifiable independently — clients can fetch them, verify Foundation's manifest authorizes them, verify their Ed25519 signature, but the snapshot is "trusted" only when the active-replica quorum holds.

8.2 Divergence detection

For any epoch E and snapshot S:

```
active_attestations = [signatures from active_replicas on snapshot_hash(S)]
shadow_attestations = [signatures from shadow_attestors on snapshot_hash(S_shadow)]
```

If `snapshot_hash(S) != snapshot_hash(S_shadow)` for the same epoch + same input submissions, that's evidence of divergence. Reasons:

- An active replica forged or altered the snapshot. Provable misbehavior. Active replica slashable.
- The shadow attestor is broken or malicious. Provable. Shadow attestor slashable.
- A submission was accepted by one tier but not the other (gossip divergence). Both diagnoseable.

The divergence claim is filed with Foundation governance (or in V2, with the on-chain registry contract). Reviewer runs the determinism check (re-build the snapshot from the input submissions) and identifies the misbehaving party. Slashing follows.

8.3 Slashing bounty

If a shadow attestor's divergence claim is confirmed AND identifies an active replica as the misbehaving party, the shadow attestor earns a **bounty payout** equal to 20% of the slashed bond (provisional). The rest of the slashed bond burns.

This creates strong incentives for shadow attestors to actively watch active replicas. The bounty is a real economic reward for caught misbehavior; the shadow attestor's own bond is at risk if they file frivolous claims (Foundation can slash a shadow attestor's bond for repeated unfounded claims).

8.4 Shadow attestor reward rate

Provisional: **10–15% of the active replica's per-epoch token reward**, paid only when shadow attestation is verifiably rendered. Verification: peers attest that the shadow attestor's signature is present + valid + matches consensus at each epoch.

The XMR fee share does NOT extend to shadow attestors — they're not the ones processing the registration; the active replicas are. Shadow attestors earn only token rewards (and slashing bounties when applicable).

9. Transport choices in technical detail

9.1 Transport variants

Variant	Replica setup	Operator setup	Privacy properties
Tor onion only	Run Tor daemon, expose onion v3 address	Run Tor; submit via onion	Both parties' IPs hidden from each other and from observers
Clearnet only	Expose HTTPS endpoint on public IP	Standard HTTPS client	Operator IP visible to replica; replica IP visible to operator + observers; metadata correlation possible
Dual-stack (recommended)	Expose both onion v3 + HTTPS endpoint	Operator chooses transport based on their privacy posture	Best of both — privacy-conscious users can choose Tor; non-privacy-sensitive users can choose clearnet

The recommended best practice is **dual-stack**. The protocol doesn't enforce a choice; the manifest entries describe what's available per replica.

9.2 What's lost on clearnet-only operators

If a replica is clearnet-only: - Operator's IP is logged at the replica. - The replica's IP is visible to anyone watching network traffic. - Jurisdictional pressure on the replica's hosting provider works directly. - DDoS attacks are straightforward.

What's preserved (because the protocol is cryptographic, not transport-dependent): - All §22 handshake security properties. - All §33 PaymentCommitment flow. - All §17 mint credential validation. - The blind credential's unlinkability between payment and registration.

So **clearnet replicas don't break the protocol's cryptographic guarantees** — they just expose operator + replica IPs to network observers. That's a real privacy degradation but not a security degradation.

9.3 What the bonded pool should target for v1 launch

For v1 launch, the bonded pool should target a **diverse transport portfolio**:

- All 5 active replicas SHOULD offer Tor onion v3 + clearnet HTTPS (dual-stack).
- Shadow attestors MAY be clearnet-only if they prefer (mobile shadow attestors will be clearnet-only by necessity).
- The manifest documents per-replica transport availability so operators can choose.

10. Mobile shadow-attestor architecture (speculative)

10.1 Hard problems for a mobile shadow attestor app

- **Wallet integration.** App needs to interact with a wallet (the user's token wallet, on whatever chain the token settles to). Existing patterns: WalletConnect, Phantom mobile, MetaMask mobile.
- **Background execution.** iOS strict; Android more lenient. Best approach: run only when app is foreground OR when device is on wifi + charging (background fetch).
- **Network reachability.** Mobile NATs prevent incoming connections. Shadow attestor doesn't need them (pull-only). Wifi vs cellular: prefer wifi (no carrier traffic logging).
- **Storage.** Snapshot bodies are large (potentially MBs per epoch). Mobile storage is limited but adequate for short-window caching. Long-term archives stay with server-based shadows.
- **Battery.** Run on wifi + charging. Pull state periodically (every 5–10 minutes when active). Skip gossip during off-hours.
- **App store policies.** Both Apple and Google have crypto-app restrictions. Token wallet integration may require workarounds. Reference: how Helium's mobile-hotspot rewards app handles this.

10.2 Sketch architecture

```
[Mobile shadow attestor app]
   ↓ uses
[Wallet SDK] ←→ [User's wallet, bonded tokens]
   ↓ talks to
[Nightshade Foundation Manifest endpoint] (cached signed manifest)
   ↓ pulls from
[Active replicas, via clearnet HTTPS] (snapshot bodies + signatures)
   ↓ builds + signs locally
[Shadow snapshot signature, uploaded to Foundation endpoint]
   ↓ rewards distributed to
[User's bonded wallet]
```

Operationally:

1. User installs app, connects wallet, bonds tokens through wallet's signing flow.
2. App registers as a shadow attestor (Foundation receives the new attestor registration; next manifest update includes them).
3. App pulls the latest snapshot from 3 active replicas. Verifies their signatures + manifest authorization.
4. App builds its own deterministic snapshot from the same input submissions (which the app received via gossip with active replicas).
5. App signs the snapshot hash with the user's bonded-attestor key.
6. App uploads the shadow signature to a Foundation-published endpoint.
7. At epoch + N (provisional N = 12 hours), Foundation aggregates shadow signatures + distributes rewards.

10.3 Limitations

- Mobile attestors will have higher offline rates than server operators. Reward rate must accommodate periodic absence.
- Mobile attestors are easier to compromise than dedicated servers. Should not be in the active set for v1.
- Wallet integration is a significant engineering investment. v2-or-later candidate; not v1 launch.

10.4 Why this is worth the engineering investment eventually

The pool depth that mobile participation unlocks is qualitatively different from server-only participation. A bonded pool of 50 server operators is impressive but discoverable. A bonded pool of 10,000 mobile shadow attestors across 100 countries is genuinely hard for any single attacker to corrupt — the cost scales with operator count, and 10,000 attestors at low individual bond is much higher cost-to-corrupt than 50 attestors at high individual bond.

This is the Helium hotspot insight applied to a different role. Helium's value-proposition is exactly "thousands of independent hosts distributed globally." Nightshade's shadow-attestor tier can capture some of that distribution-as-security benefit.

11. Revised reward economics

Updates to `replicas-detail-2026-05-25.md` §11 + `replicas-open-set-and-halving-2026-05-25.md` §10:

11.1 Active replicas

- **Token reward:** R■ at v1 launch, halving every 12 months (Choice A: per-replica rate stable within halving period).
- **XMR fee share:** 1% of each registration mint fee → split per-registration among signing replicas.
- **Bond:** `min_active_bond` from manifest, provisionally 100,000 token at launch.
- **Slashing:** as per prior docs.
- **Unbond cooling:** 90 days.

11.2 Shadow attestors (NEW — replaces my prior "no reward" recommendation)

- **Token reward:** 10–15% of active replica's per-epoch reward, paid only when shadow attestation is verifiably rendered. Halves on the same schedule as active rewards (so the relative gap stays constant).
- **No XMR fee share.** Active replicas are doing the work of processing registrations; shadow attestors are doing parallel attestation.
- **Slashing bounty:** 20% of slashed active-replica bond if shadow attestor's divergence claim is confirmed.
- **Bond:** `min_shadow_bond` from manifest, provisionally 10,000 token (1/10 of active bond) for server-based shadow attestors. Mobile attestors v2+ candidate; bond TBD.
- **Slashing:** for offline-without-pause (failure to attest while claiming to be active) at 5–15%; for filing unfounded divergence claims repeatedly at 10–25%.
- **Unbond cooling:** 90 days, same as active.

11.3 Promotion path

- When an active-set slot opens, Foundation selects from the bonded pool ranked by: 1. Bond age (longer = higher priority) 2. Shadow attestation history (more consistent = higher priority) 3. Jurisdictional + hosting diversity contribution 4. Slashing-bounty history (one or more successful catches = positive signal)

- A shadow attestor promoted to active immediately transitions to the active reward rate + XMR share. Their bond automatically increases to `min_active_bond` (operator must top up; otherwise they decline promotion).
- An active replica that exits unbonds normally OR can transition to shadow-attestor role (much smaller bond release; still slashable on shadow terms).

12. Open design questions added in this iteration

1. **Manifest cache lifetime.** Provisional `valid_through_unix_seconds` = 7 days after publication. Too short = lots of refresh traffic; too long = expansion lag.
2. **Foundation governance vote quorum and threshold.** Provisional >50% of cast votes AND >20% of circulating supply participating. Calibrate based on actual token holder count.
3. **Shadow attestor reward rate.** Provisional 10–15% of active rate. Higher = larger pool, more emission; lower = smaller pool, less emission.
4. **Slashing bounty %.** Provisional 20% of slashed bond. Higher = stronger watchdog incentive; lower = more of slashed amount remains as burn.
5. **Mandatory mandatory-review cadence.** Provisional every 12 months. Different cadences possible.
6. **Foundation publication endpoint resilience.** Mirror to onion service + multiple geo-distributed clearnet endpoints. CDN with origin failover.
7. **Manifest version rollback.** If a new manifest is signed and turns out to be malicious or buggy, what's the rollback path? Pre-signed previous manifest, retroactively re-served? Emergency manifest published?
8. **Mobile shadow attestor: which chain.** The wallet integration depends on which chain the token settles to. The Hyperliquid-vs-Solana-vs-EVM choice from the strategic-opportunity discussion blocks the mobile implementation question.
9. **Shadow attestor onboarding KYC.** Active replicas have Foundation-vetted KYC-light review. Shadow attestors at lower bond should have lower KYC bar, but how much lower? Self-attestation? Wallet-history requirements?
10. **Transition from v1 Foundation-signed manifests to v2 on-chain registry.** Specific milestone: when the token's settlement chain has stabilized + a registry contract is deployed + audited.

---

13. Summary

Fifth document in the working tokenomics design surface as of 2026-05-25. Companion to `tokenomics-research-2026-05-25.md` (wide research), `tokenomics-option-e-detail-2026-05-25.md` (Option E first-pass), `replicas-detail-2026-05-25.md` (replica role + reward), and `replicas-open-set-and-halving-2026-05-25.md` (open set + halving). The five documents together describe the design in enough depth for counsel review + strategic-opportunity-driven final calibration. None is a binding decision.