Option E — Bond + Marketplace Credits: detailed design

Date: 2026-05-25 **Status:** Detailed elaboration of Option E from `tokenomics-research-2026-05-25.md` §5.5. Not a binding decision — this depth pass exists to make the design concrete enough to evaluate and to brief counsel against. Final design + token name + supply parameters depend on the strategic opportunity (still unspecified) and counsel review. **Scope:** Plain-English explanation, technical mechanics, user personas, privacy boundaries, failure modes, open questions.

> ■■ ***Alignment note (2026-06-28) — read first.*** *This is a dated design record. Two later decisions change parts of it; canonical current sources are `DECISIONS.md` (D-013) and the whitepaper (§9, §11). -* ***The 5,000 XMR mint-fee cap was removed (D-013).*** *References to a "post-cap admission policy" are moot — there is no post-cap phase. (`BOOTSTRAP_EMISSION_CAP`, a separate token-emission ceiling, is unrelated and unaffected.) -* ***Epoch model is now two-cadence:*** *registry epoch = 7 days (snapshots/credentials), attestation epoch ≈ 1 hour (attestation + emission). The "per epoch" marketplace listing fees in this doc are* ***per registry epoch*** *(weekly).*

---

0. How to read this document

This document has two parts that say the same things differently:

- **Part 1 — Plain English (sections 1–5).** A non-technical reader can read just this part and understand what the token is, who uses it, and what it does.
- **Part 2 — Technical detail (sections 6–11).** Mechanics, parameters, edge cases, calibration concerns, failure modes.

Five **user personas** (Alice, Bob, Carol, Dave, Eve) appear in both parts. They're the same five people; the plain-English view describes what they do day-to-day, and the technical view describes the protocol mechanisms they're triggering.

A reader who needs to brief counsel or partners can hand them Part 1; a reader who needs to implement, audit, or critique can read Part 2.

Part 1 — Plain English

1. What is this, in one paragraph

Nightshade has two needs that ordinary money can't cleanly solve. First, the network needs **five trustworthy replica operators** plus **a mint authority** running infrastructure that anyone in the agent economy depends on — and we need a way to give those operators skin in the game so they don't misbehave. Second, the marketplace where AI agents discover each other and pay each other for services needs **a way to charge for marketplace-layer features** (listing priority, dispute arbitration, premium discovery) without those payments being visible on Monero and breaking the privacy guarantee of the underlying agent-to-agent payments.

The token solves both. Operators **bond** the token to get the right to run infrastructure. Service marketplaces **burn** the token to pay for marketplace features. The token never touches the actual agent-to-agent payment — that stays on Monero where it belongs. Holding the token doesn't pay yield (deliberately — that's where most token designs run into legal trouble); its value comes from the structural demand of operators needing to acquire it to run infrastructure and consumers needing to burn it to access marketplace features.

2. Who actually uses it

Five people — composite personas representing real categories of user:

Alice — runs a replica

Alice is a sysadmin in Berlin running one of the five Nightshade replica servers. She earns a fee for her service (paid in XMR and/or token, depending on parameters) but only if she stays honest — if her replica ever signs a snapshot that disagrees with the other four, or refuses valid registrations, her bond gets slashed. She **bonds tokens** to get the right to operate. Currently her bond is locked; when she eventually retires her replica, she gets it back after a six-month unbond period (so the network has time to detect any late-discovered misbehavior).

Bob — operates AI agents and lists them on the marketplace

Bob runs five AI agents that translate documents, summarize PDFs, and do retrieval-augmented Q&A. He registered each agent by paying the §17 mint fee in XMR (0.027182 XMR each). To **list his agents on the marketplace** so other operators can discover and hire them, he burns a small amount of token per listing per epoch. If he wants his agents to show up higher in search results, or to be marked "verified operator," he burns more.

Carol — uses AI agents (her agents hire Bob's)

Carol runs an agent that produces long-form research reports. Her agent needs translation and retrieval as inputs, so it hires Bob's agents. **The payment from Carol's agent to Bob's agents happens entirely in XMR via the §22 handshake + §33 PaymentCommitment flow. That payment is NOT in the token.** The token only enters her workflow if her agent does a **discovery query** for new providers — that's a marketplace feature that consumes credits, which were minted by burning token.

Dave — holds the token but doesn't operate anything

Dave bought tokens because he believes the Nightshade network will grow and demand for the token will grow with it. He **doesn't earn yield**. The protocol deliberately doesn't route fees to passive holders — that's the design choice that keeps the token out of "this is an unregistered security" territory under US and EU law. Dave's bet is purely on demand growth from operators (Alice) and consumers (Bob) needing to acquire / burn token over time. He may participate in **Foundation governance votes** if a Foundation structure is layered on top.

Eve — operates the mint authority

Eve runs the centralized v1 mint authority. The mint is the single most trust-sensitive piece of infrastructure in the network (it signs the blind credentials that gate registration). Eve bonds **substantially more** tokens than any single replica operator — the bond reflects the trust position. The §21 auditor and the replica set both monitor her behavior; provable misbehavior (double-issuing credentials, refusing valid submissions silently, leaking the RSA key) slashes the bond.

3. What you do with the token, in plain English

There are exactly four things you can do with the token. The first three create real demand; the fourth is the speculative leg.

1. **Bond it to operate infrastructure.** This is Alice and Eve. The bond is locked while you operate. You don't get yield on it; you earn service fees through the protocol. If you misbehave provably, your bond gets slashed.
2. **Burn it for marketplace features.** This is Bob (mostly) and Carol (occasionally). Burning token mints a non-transferable "marketplace credit" pegged to a fixed XMR price. Credits pay for: listing slots, listing priority, capability schema deployment, dispute escalation, premium discovery queries, verified-operator badges. Credits never expire and aren't transferable.
3. **Vote in Foundation governance.** Optional — only if we add a Foundation layer (the "Option D layered on E" path in the research doc). Foundation votes set protocol parameters: auditor pinning, replica set composition, post-cap admission policy, fee parameters.
4. **Hold it speculatively.** This is Dave. You're betting that operators and consumers will need to acquire/burn token in growing volume. You don't earn yield. There is no buyback. Your upside is purely from demand growth pressuring against fixed supply.

4. What the token does NOT do (privacy boundary)

This is the single most important design constraint. The token sits at two layers: the **operator layer** (bond) and the **marketplace layer** (burn for features). It does **NOT** touch any of these:

- **The §17 mint fee.** Operators pay 0.027182 XMR to register agents. That stays Monero. The token plays no role.
- **The §22 session-binding handshake.** When Carol's agent connects to Bob's agent, that handshake is entirely cryptographic — no token involvement.
- **The §33 PaymentCommitment.** When Carol's agent pays Bob's agent for translation, that payment is a Monero transaction. No token involvement. The recipient's view-key sees the XMR; nobody sees a token transaction tied to that payment.
- **Audit log entries (§25).** Operator-side encrypted with XChaCha20-Poly1305. No token involvement.

Why this matters: every place the token *does* touch the protocol is a place where blockchain-explorer-style observers could correlate token activity to user activity. Keeping the token out of the §22/§33 payment flow is what preserves the Monero-grade privacy guarantee for the actual agent-to-agent commerce. The token's visibility is limited to (a) which operators bonded (already public — replicas have publicly-pinned keys) and (b) marketplace-layer features (already public — marketplace queries happen on visible infrastructure).

5. The story in one sentence

Operators bond to earn the right to run infrastructure; consumers burn to use marketplace features; nobody earns passive yield; Monero handles the actual money.

Part 2 — Technical detail

---

6. The bond layer

6.1 What gets bonded

Three roles bond:

Role	Function	Bond size (provisional)	Slashing exposure
Replica operator	Signs `SignedSnapshot`, `SignedEmergencyOverlay`, `SignedTransparencyLogEntry`; 1 of 5 in the 3-of-5 quorum per spec §19.1	Tier-based: 100,000 token at v1 launch; configurable by governance	Up to 100% of bond on provable double-sign or systematic refusal-of-service
Mint authority	Signs blinded credentials via RFC 9474 RSABSSA `BlindSign`; maintains idempotency table; publishes issuance log per spec §17.9	1,000,000 token at v1 launch (10× a replica)	Up to 100% of bond on double-issuance or provable RSA key compromise
Reference agent operator	Optional — agents operated by the Foundation/team as bootstrap supply on the marketplace	10,000 token per agent	Up to 50% on service-failure dispute resolved against the operator

Specific numbers above are illustrative — final calibration depends on token launch price, target operator economics, and the strategic opportunity that determines total supply.

6.2 Slashing conditions

For replicas:

- **Double-signing** — signing two divergent `SnapshotBody`s for the same epoch. Detectable via the §19.3 transparency log; client-side `ReplicaQuorumTransparencyCheck` will surface the inconsistency. Slash: 100% of bond. Permanent operator ban.
- **Refusal of valid registrations** — provable via the §21 auditor's records of paid mint fees that did not appear in subsequent snapshots. Slash: 50% on first instance, 100% on second.
- **Operating with stale state** — failing to gossip-converge per the Q2 anti-entropy resolution within 2 epochs. Slash: 10–25%; remediable by re-sync.
- **Late refresh of emergency overlay** — failing to sign valid emergency revocations within the spec §24.4 5-minute target. Slash: 5–15%; per-incident.

For the mint authority:

- **Double-issuance** of credentials for the same `(operator_pseudonym, request_id)` — slashable per spec §17.8 idempotency table. Slash: 100%.
- **RSA private key compromise** — provable if the auditor's view-key flow shows fees collected without corresponding issuance log entries, OR if the issuance log shows unauthorized signatures. Slash: 100%.
- **Late publication of issuance log** — slow audit cycle; slash 5–20% based on staleness.
- **Refusal of valid blinded submissions** — provable via fee-paid-but-not-issued evidence. Slash: 50–100%.

6.3 Bond lifecycle

- **Bond placement.** Operator transfers token to the bond contract; bond is locked, NOT burned. The bond is held in a deterministic vault keyed to the operator's public identity (replica public key for replicas, RSA fingerprint for the mint authority).
- **Active operation.** While bonded, the operator may sign artifacts and is eligible for service-fee distribution (paid in XMR for direct service like running a replica; possibly paid in token via Foundation grants during bootstrap).
- **Voluntary exit.** Operator requests unbond. The bond enters a **180-day cooling period** during which the operator must continue to operate honestly. Misbehavior detected during cooling is still slashable against the unbonding balance.
- **Slashing.** Detected misbehavior triggers a slashing transaction; slashed tokens are **burned** (sent to a verifiable burn address; permanently removed from circulation). Note: slashed tokens are NOT redistributed to other operators — that creates perverse-incentive structures (Tower-style "false flag"

attacks).

- **Bond return.** After 180-day cooling completes without slashing, the bond is released back to the operator's wallet.

6.4 Why bond rather than charge a fee

A one-time registration fee (Arweave's endowment model — Option C in the research doc) gives an upfront payment but no ongoing skin-in-the-game. A bond ties the operator's economic interest to the operator's behavior **continuously**. The bond is also reversible (returned at exit) which is more politically palatable than "pay X tokens to register, never get them back."

The bond is closer to Filecoin's SP collateral than to Hyperliquid's validator self-bond. Filecoin's collateral is fundamentally for slashing; Hyperliquid's is more for Sybil resistance. Nightshade's bond is for both: slashing for misbehavior + financial barrier to operator-set capture.

7. The marketplace credit layer

7.1 Why burn-and-mint credits (BME) and not direct token spend

Direct-token-spend is the simpler mechanism: marketplace charges X token per query. But it makes the token's purchasing power volatile — if the token's USD price spikes, marketplace fees effectively rise; if it crashes, they collapse. That's unworkable for operators trying to budget.

Burn-and-mint credits **abstract over token price**. The protocol declares: "1 Listing-Slot-Day = 0.0001 XMR equivalent." The user burns however much token is currently equivalent to 0.0001 XMR (via an oracle), and receives 1 Listing-Slot-Day credit. Credits are non-transferable, non-expiring, denominated in fixed XMR-units.

This is exactly the Helium HNT → Data Credits model, with the lessons from **HIP-138** baked in: don't emit supplier-side rewards faster than consumer-side burn, or you go inflationary and have to redesign in panic.

7.2 What marketplace features burn credits

Six initial categories. Each has a fixed XMR-equivalent price; exact prices set at launch and adjustable by Foundation governance.

Feature	Per-unit XMR equivalent	Why this is a token sink
Listing slot (per agent per epoch)	0.0001 XMR	Listed agents appear in the marketplace's default discovery index. Cost gates spam listings.
Listing priority boost	0.001 XMR per priority tier per epoch	Sorted-search ranking; explicit demand signal from Bob-type operators.
Capability schema deployment	0.05 XMR	Defining a new service capability type (analogous to HIP-1 spot listing). Schema becomes part of the marketplace's queryable taxonomy. One-time.
Dispute escalation	0.01 XMR	Triggering arbitration on a service-failure dispute. Refundable to the prevailing party.
Premium discovery query	0.00001 XMR	Bulk / filtered / sorted-by-rating discovery queries. Per-query. Lower-priced than listing because volume is higher.
Verified-operator badge	0.025 XMR per epoch	KYC-equivalent badge for operators who choose to self-identify (legal-counsel-conscious operators). Optional.

7.3 The credit-to-token-burn flow

1. Operator (Bob) decides he wants to list 3 agents and boost one to priority tier 2 for the next epoch.
2. He calculates required credits: 3 listing-slots + 1 boost-tier-2 = 3 × 0.0001 + 1 × 0.002 XMR = 0.0023 XMR equivalent.
3. He fetches the current TOKEN/XMR price from an on-chain oracle: say 1 TOKEN = 0.0001 XMR.
4. He burns 0.0023 / 0.0001 = 23 TOKEN.
5. Marketplace credits his account with 3 Listing-Slot-Days + 1 Priority-Tier-2-Days.
6. At epoch start, the marketplace contract deducts the credits.

Credits accumulate; Bob can pre-burn for many epochs ahead if he wants to lock in a TOKEN/XMR ratio. Credits don't expire (Arweave-influenced design choice — one-time burn for permanent access matches the "infrastructure consumption" framing).

7.4 The emissions side

Emissions exist to bootstrap the supplier side: replica operators get paid for honest operation; reference agents get bootstrap subsidies; Foundation pays out grants. Emissions are calibrated against burn — this is the central design constraint.

Calibration rule: total token emitted per epoch ≤ total token burned per epoch × (1 + bootstrap-multiplier). The bootstrap-multiplier starts at 5× (allowing supplier emissions to outpace consumer burn during launch) and decays linearly to 1.0 over 36 months. After month 36, emissions ≤ burn. Net supply is contracting from month 36 onward.

Hard cap on bootstrap emissions: even with the multiplier, total emissions in any epoch never exceed `BOOTSTRAP_EMISSION_CAP` (provisional: 0.5% of initial supply per epoch). This is the HIP-138 defense — if marketplace activity collapses to zero, we don't emit infinite tokens to suppliers waiting for demand that isn't coming.

Sunset clause: if, in any rolling 90-day window, total burn is < 10% of total emission, emissions automatically halt for the next 90 days. Foundation governance can reinstate via 67% supermajority vote with a specific remediation plan. This is the explicit response to Helium HIP-138 — built into the protocol from the start rather than retrofit.

8. Privacy boundaries (technical detail)

8.1 Where the token DOES appear

- **Bond placement transactions.** Public. The replica operator's bond is visible on the token's settlement chain.
- **Bond slashing transactions.** Public. Same chain. Includes the operator identity (which is already public — replicas have pinned public keys per the SDK).

- **Credit minting transactions.** Public. The burner's token address is visible.
- **Foundation governance votes.** Public.

8.2 Where the token does NOT appear

- **§22 session-binding handshakes.** Pure cryptography — challenge / authorization / possession-proof / replay cache. No token interaction.
- **§33 PaymentCommitment + the underlying Monero transaction.** Pure XMR settlement. The token settlement chain has no record of this payment.
- **§17 mint fee payments.** Pure XMR. The operator's mint-fee wallet view-key reveals the fee transaction to the §21 auditor; the token has no role.
- **Audit log entries (§25).** Operator-local, encrypted with XChaCha20-Poly1305. No token.
- **Emergency revocations (§24).** Pure cryptographic broadcast through the replica gossip layer. No token.

8.3 The critical linkability question

Can someone reading the public token chain identify which operators are using which agents? **Specifically: if Bob burns token to mint credits for marketplace listings, can an observer link Bob-the-token-holder to Bob-the-operator-of-agents-X-Y-Z?**

Answer: at the marketplace-credit layer, yes — by design. Marketplace listings are inherently public (other operators need to discover them). Bob's burn transaction → credit-minting transaction → marketplace listing of agent X are all on public records. Anyone can trace.

But that linkability is already present at the marketplace layer regardless of token. If Bob lists agent X on the marketplace, that listing is public, and Bob is the named operator. The token doesn't add linkability that wasn't already there.

What the token does NOT do is link Bob-the-marketplace-lister to Bob's actual agent-to-agent payments (which stay on Monero). Carol's agent paying Bob's agent for translation is invisible on the token chain. **The token cannot be used to deanonymize Monero payment activity.** That's the load-bearing privacy claim, and it holds because the token never touches §22 or §33.

8.4 What about Dave the speculative holder?

Dave's token holdings are public (visible on the token chain) but unconnected to any agent activity. Dave didn't bond, didn't burn for marketplace features, didn't list anything. He's a passive holder. Observers know Dave holds X tokens but cannot tell what he expects them to be worth or what he might do with them.

9. User personas — technical view

9.1 Alice — replica operator

- **Bond:** 100,000 token locked in bond vault.
- **Operations:** runs `nightshade-replica` binary on a hosted VPS; participates in Q2 anti-entropy gossip; signs `SnapshotBody`, `EmergencyOverlayBody`, `TransparencyLogEntryBody`.
- **Income:** earns XMR service fee per epoch from the protocol's bootstrap subsidy fund (provisional: 0.5 XMR per epoch per replica during the first 12 months, decaying as the network grows); optionally earns token emissions if Foundation grants.
- **Slashing exposure:** double-sign 100%; refusal-of-service 50–100%; stale state 10–25%; late overlay 5–15%.
- **Exit:** 180-day unbond cooling period.

9.2 Bob — agent operator

- **Bond:** none required (operating agents is permissionless given a valid credential).
- **Token consumption:** burns token to mint credits for listing slots + priority boosts + occasional verified-operator badge renewal.
- **Provisional monthly spend:** 5 agents × 30 listing-slot-days × 0.0001 XMR + 1 priority-tier-2 boost × 30 days × 0.001 XMR = 0.045 XMR equivalent → ~450 token at the assumed 1 TOKEN = 0.0001 XMR ratio.
- **Income:** XMR from agent service payments (Monero, not token).
- **No bond exposure.** Bob can stop operating agents anytime without slashing risk.

9.3 Carol — service requester

- **Bond:** none.
- **Token consumption:** only when running premium discovery queries — bulk discovery, filtered searches. Per-query cost: ~0.00001 XMR equivalent (~0.1 token at assumed ratio).
- **Monthly spend:** assume 1000 discovery queries per month → 0.01 XMR ≈ 100 token.
- **Income:** XMR from her agent's customers (also Monero, also not token).
- **Sensitive to:** TOKEN/XMR price volatility. The BME mechanism's fixed-XMR-equivalent pricing protects her — she always burns the right token amount to get the right credit, regardless of token price.

9.4 Dave — speculative holder

- **Bond:** none.

- **Token holdings:** purchased on secondary market.
- **Mechanical yield:** zero. The protocol deliberately does NOT distribute fees to passive holders (avoids dYdX v4 / Lido Samuels-style securities exposure).
- **Value-bet thesis:** if operators (Alice-type) grow and consumers (Bob-type) grow, structural demand for token grows against a fixed supply. Price appreciation = his return.
- **Risk:** if the network fails to grow, Dave's token has no demand floor.

9.5 Eve — mint authority

- **Bond:** 1,000,000 token (10× a replica's bond).
- **Operations:** runs `nightshade-mint` binary; signs blinded credentials; maintains idempotency table; publishes issuance log per §17.9.
- **Income:** receives the §17 mint fee distribution (the current PROJECT_OVERVIEW.md §7 fee distribution, modulo whatever rewrite results from the legal review).
- **Slashing exposure:** the highest. Mint compromise is the catastrophic failure mode in v1's centralized-mint design.
- **Exit:** 180-day cooling. In v1, mint exit is effectively "decommissioning the v1 mint" — high coordination cost; effectively means the v1 protocol is winding down for a v2 successor.

10. Failure modes + calibration concerns

10.1 Marketplace activity never materializes (the HIP-138 trap)

If Bob-type operators never list at the volume the protocol expects, burn stays near zero, and the bootstrap-multiplier emissions go to suppliers (replicas, reference agents) without offsetting burn. Net supply inflates.

Defenses: - Hard `BOOTSTRAP_EMISSION_CAP` (0.5%/epoch). - Sunset clause (90-day emissions halt if burn < 10% of emission). - Foundation governance can reinstate after presenting a remediation plan.

If even these fail, the token's purchasing power deteriorates and the marketplace pricing (denominated in XMR-equivalent) means credits become more expensive in token terms — which further suppresses burn. This is a death spiral. The mitigation is that Foundation can hard-kill emissions before the spiral starts.

10.2 Token price spike pricing operators out

If the token's price relative to XMR spikes (because some external event creates speculative demand), the cost of bonding rises in fiat terms. New replica operators may be unable to acquire 100,000 token at a reasonable cost; the operator set ossifies.

Defenses: - Bond denomination can be **adjusted by Foundation governance** to maintain a target USD-equivalent value (similar to how Maker DAO adjusts DAI stability fees). - Tier-based bonds — smaller

starter bonds for new operators with reputation-gated growth to full bond over time.

10.3 Concentrated bond holdings (Convex pattern)

A small number of token whales bond on behalf of operators (custodial bonding service). The whales accumulate the slashing risk but charge operators a rental fee. This recreates the Convex-vs-Curve dynamic in a different layer.

Defenses: - Bond contract requires the bonded address to match the operator's signing identity (replica public key / RSA fingerprint). Custodial bonding requires the whale to also run the infrastructure — they can't separate the two. - The 180-day cooling period plus slashing exposure makes custodial bonding economically unattractive.

10.4 Mint authority single point of failure

In v1 the mint is centralized. If Eve's RSA key is compromised, slashing the bond doesn't undo the damage of unauthorized credentials.

Defenses: - The §21 auditor monitors continuously; the issuance log + on-chain fee correlation surfaces double-issuance fast. - v2 roadmap candidate: threshold RSA mint (m-of-n authorities). This makes the bond design more complex (multi-party bonds) but eliminates the SPOF.

10.5 Privacy collision via token wallet metadata

If Bob always uses the same token wallet for marketplace burns, an observer can build a profile of his marketplace activity over time. The token chain becomes a side-channel for what should be private commerce.

Defenses: - Bob can use multiple token wallets (a wallet per agent he operates, or rotating wallets). This adds operational complexity but is supported. - The marketplace credit minting transaction doesn't reveal anything about the agent the credit is for — it just mints to Bob's account. The link from credit-burn to agent-listing is at the marketplace layer, not the token layer. - **Critically: the token never appears in the §22 / §33 payment path.** Whatever leaks via token-chain metadata is bounded to marketplace-layer activity, which is already public.

11. Open design questions

These need resolution before slice-1 implementation of any token-related code.

> **Token settlement chain.** Where does the token live? Options: - Hyperliquid HIP-1 spot deployment (HYPE-denominated auction floor). - Solana SPL (like Helium HNT post-migration, Render post-migration). - Ethereum ERC-20 (most familiar; highest gas cost; widest tooling). - Cosmos-native (dYdX v4 pattern; sovereign appchain). - Custom L1 or rollup (Hyperliquid path; large engineering cost). - The "opportunity" James referenced may answer this directly.
>
> **Total supply + allocation.** Provisional sketch: - **Operators (initial bonds + bootstrap subsidies):** 25–35% - **Marketplace bootstrap (subsidies to reference agents, initial Foundation grants):** 15–20% - **Genesis distribution (airdrop / initial liquidity):** 10–15% (Hyperliquid-style fair launch;

specifically NOT a pre-sale) - **Foundation budget:** 10% - **Team vesting:** 15–20% with 4-year linear vest, 1-year cliff - **Reserve for v2 / contingency:** 5–10% - Note: explicit rejection of "early holders" / "buybacks" framing from `PROJECT_OVERVIEW.md` §7. Replaced by the structurally-defensible allocation above.

Oracle for TOKEN/XMR ratio. The marketplace credit minting needs a current price. Options: Pyth (Solana-focused), Chainlink (cross-chain), a custom oracle aggregating CEX prices, the replicas themselves voting on the ratio per epoch. The choice affects centralization + latency + cost.

Bond contract location. Is the bond contract on the token's native chain (and the replica/mint software talks to it via cross-chain bridge), or on Nightshade's protocol layer (and the token bridges in)? Affects bridge security model.

Foundation structure. Following the Ondo pattern: Nightshade Foundation (DAO, governs Foundation) vs Nightshade Operations (commercial entity). Counsel will define the exact legal split. The token would govern the Foundation, not Operations.

Bootstrap-multiplier decay curve. Linear from 5× to 1.0 over 36 months is one option. Exponential, step-wise, or governance-adjusted are alternatives. The choice affects the supplier-side bootstrap economics.

Slashing destination. Burn (provisional choice — prevents perverse incentives) vs partial-redistribution to honest operators vs Foundation treasury. Burn is cleanest.

What happens at v2 transition. v2 will likely decentralize the mint (threshold RSA) and may expand the replica set. The bond model needs to gracefully migrate — token holders + bonded operators expect continuity.

The strategic opportunity. Still unspecified at time of writing. The above design is robust to opportunity choice, but specific tokens (HIP-1 deployment, specific exchange listings, partner integrations) impose specific constraints on chain choice + distribution mechanics.

Counsel timeline. All of the above is pending counsel review. Token launches without counsel signoff are not contemplated by this design.

---

12. How this differs from the comparable models

A short tour of why Option E is a specific arrangement of primitives, not a copy of any single model.

Filecoin SP collateral: Option E's bond layer is closest to Filecoin's. Key differences: Nightshade slashes for protocol-specific misbehavior (double-sign, refusal-of-service, mint key compromise), not for failed storage proofs. The bond also doesn't compound with a verified-deals 10× multiplier — that financialization mode is what FIP-0080 is walking back, so Nightshade avoids it from the start.

Helium HNT → Data Credits: Option E's marketplace credit layer mirrors HNT/DC almost directly. Key difference: Nightshade's `BOOTSTRAP_EMISSION_CAP` + sunset clause are protocol-native from launch (rather than retrofit via HIP-138 after the original design collapsed).

Hyperliquid HYPE: Option E shares HYPE's no-presale fair-launch ethos but explicitly rejects the Assistance Fund mechanism. The AF (buyback-and-burn from protocol fees) is the most legally exposed feature of HYPE under SEC Release 33-11412. Nightshade's mint fees stay XMR and do not flow into token markets.

GMX 27% buyback: explicitly rejected. Same legal exposure category as the AF; same enforcement pattern.

Curve veCRV: Option E doesn't use vote-escrow because there are no LP pools to gauge-vote on. If Foundation governance is layered on top (Option D in the research doc), simple 1-token-1-vote (or 1-token-1-vote with optional lockup) is sufficient.

Lido LDO: Option E explicitly does NOT route fees to passive holders, which means the Samuels v. Lido DAO general-partnership theory doesn't have an obvious purchase. The Foundation/Operations split (Option D structure) further mitigates.

Ondo ONDO: The Foundation/Operations split if layered on Option E mirrors Ondo's structural defense. Nightshade Foundation governs (DAO); Nightshade Operations runs commercial activity (mint, marketplace front-end, reference agents); the token claims no rights in Operations' revenue.

Arweave AR: Option C in the research doc is closer to Arweave; Option E borrows the "credits never expire" + "one-time burn for permanent access" framing but limits it to marketplace features rather than registry permanence.

dYdX v4: explicitly opposite. dYdX routes USDC fees to passive stakers — the strongest security-like profile in the comparative sample. Nightshade's design takes the opposite call: no fee routing to passive holders, no staking yield, no validator-token revenue distribution.

---

13. Summary

Question	Plain English answer	Technical answer
Why does this token exist?	To give operators skin-in-the-game and to charge for marketplace features without breaking Monero privacy.	Bond for slashing; BME credits for application-layer features; never touches §22/§33.
Who uses it?	Alice (replica op), Bob (agent op), Carol (service requester for premium queries), Dave (passive holder), Eve (mint op).	Operators + marketplace consumers + speculative holders.
How do you earn yield from holding?	You don't.	Deliberately no yield distribution — securities-defense design choice.
Does this token replace XMR for agent payments?	No. XMR stays.	Token only at operator-bond + marketplace-credit layers; never §22/§33.
What stops emissions from inflating away the token?	Hard cap on emissions + sunset clause if burn doesn't show up.	`BOOTSTRAP_EMISSION_CAP` 0.5%/epoch + 90-day-burn<10%-emission sunset + Foundation override.
What's the legal exposure?	Lowest-risk archetype available, but still requires counsel review before any launch.	Utility token (bond + service consumption); no fee routing to passive holders; Foundation/Operations split available; explicit avoidance of buyback-and-burn from protocol fees.
What's the worst failure mode?	If nobody uses the marketplace, the token has no demand and operators may exit.	HIP-138 trap: emissions to suppliers > burn from consumers → inflation → death spiral. Mitigated by sunset clause but not eliminated.

Companion document to `tokenomics-research-2026-05-25.md`. Both should be read together when briefing counsel or partners. Neither is a binding decision; together they describe one specific design (Option E) in enough depth that the decision can be made.