Hearth: A Verified-Inference Layer-1 Secured by Proof of Inference

July 3, 2026
DRAFT Version

Abstract

Hearth is a Layer-1 blockchain running an open market for verifiable AI inference, built on a mature Fair Proof-of-Stake (FPoS) base and secured by a new consensus mechanism, *Proof of Inference* (PoI): useful, verified inference amplifies a staker's block-production probability, and with no demand the chain falls back to ordinary Fair Proof-of-Stake. The network token, HRTH, secures the chain and subsidizes early supply; a single USD-denominated, capacity-backed, burn-on-use Cred meters inference in Phase 1 and, as verified self-hosted capacity comes online, graduates into per-model Creds that trade on open markets, so each model's Cred price becomes the live, discovered price of its proven output. Verification is what makes inference tradeable: model-pinning and hardware-attested execution turn each model's output into a standardized, fungible good, backstopped by an attested-verifier committee and distributional statistical tests, with zero-knowledge and interactive fraud proofs reserved as a trustless floor. The network ships in three phases of rising assurance, Relay (attested gateways to hosted models), Compute (self-hosted confidential GPUs under the full verification stack), and Coordinate (verifiable multi-model orchestration), a capital-efficient rollout that stands up the market before securing high-value settlement. This paper specifies the market, the consensus and token mechanisms, the verification framework, and each phase.

1 Introduction

Large language models are economic infrastructure, yet running one is rarely *verifiable*. In hosted APIs a buyer's only assurance that the advertised

model produced the output, unsubstituted and at full precision, is provider reputation; silently serving a cheaper or quantized model is tempting and hard to detect from outputs alone [1]. A hosted model is also a revocable chokepoint, since providers routinely deprecate and retire models [2, 3]. As inference becomes a commodity traded between agents with no prior trust, the absence of a checkable receipt becomes the binding constraint.

Decentralized inference networks remove the intermediary but inherit a harder problem: anonymous, incentivized nodes can run a cheap model, fabricate a reply, or charge for work never done. Re-running and comparing fails because generation is nondeterministic, and zero-knowledge proofs of a forward pass remain far too slow for frontier-scale models [4].

1.1 Contributions

Hearth composes the mechanisms that work today, hardware attestation, statistical model-identity testing, and committee re-execution, into a layered guarantee anchored in a proof-of-stake chain whose block production is amplified by verified useful work. This paper defines *Proof of Inference*, a Fair Proof-of-Stake modification in which verified inference is a capped multiplier on forging weight, falling back to pure stake without demand (§4); specifies a two-asset economy of a network token (HRTH) and a USD-denominated inference credit (Cred), single in Phase 1 and graduating into per-model Creds in Phase 2, with the liquidity-mining subsidy that bootstraps it (§5); presents a layered verification framework that establishes correctness without trustless proofs of frontier inference (§6); and details a three-phase deployment, Relay, Compute, and Coordinate, each a strict superset of the last (§7–§9).

1.2 Why a dedicated L1

Every mechanism Hearth requires is consensus-level, the forging-weight rule, generation-suspension as the penalty for cheating, and native verification opcodes (secp256r1, zkTLS), none expressible as an application-layer contract. Building on a mature Fair Proof-of-Stake L1 whose block-generation delay function Hearth inherits [5] supplies a contract VM, native asset issuance, and node software while granting full control over consensus, penalties, and monetary policy from genesis. The cost is bootstrapping security from zero: a proof-of-stake chain's security is bounded by the market value of its staked coin, and HRTH launches with none, leaving the chain cheaper to attack

than an established L1, so the first phase exists to attract stake and usage before the network secures anything of value.

2 Background and related work

Trusted Execution Environments (TEEs), such as Intel TDX/SGX and confidential-computing modes on NVIDIA H100 and Blackwell GPUs, let a workload prove, via a vendor-signed quote, what code it ran. Confidential GPU inference costs single-digit-percent overhead and lets a verifier confirm the advertised model is loaded, shifting trust from operator honesty to the vendor's attestation root [6, 7]. It is central to Hearth's guarantee, backstopped by independent statistical tests (§11.2).

Verifying the computation itself has two lineages: optimistic schemes such as opML resolve disputes with fraud proofs over a re-executable trace [8], following Arbitrum's bisection [9], and zero-knowledge approaches such as zkLLM prove a forward pass succinctly but stay too slow for large models [4]. Locality-sensitive commitments, TOPLOC [10] and DiFR [11], make logit comparison robust to GPU floating-point nondeterminism. Hearth uses attestation and committee re-execution live, reserving ZK and bisection as a future trustless floor.

Independently, Model Equality Testing (MET) checks whether an API serves the claimed model via a Maximum Mean Discrepancy two-sample test with a string kernel [12, 13], and watermarking embeds a detectable signal in outputs [14]; both anchor Hearth's Relay-phase defenses when re-execution is unavailable. For consensus, Hearth builds on Fair Proof-of-Stake, where block-generation delay depends on generating balance and an unpredictable VRF hit [5], modifying only the balance term.

3 Design principles

Three principles run through the design. Stake is the security floor: verified work amplifies a staker's weight up to a capped multiplier but never substitutes for it, so compute without stake cannot influence consensus and, from Phase 2, cannot even serve inference (§5.3); security never falls below pure proof-of-stake, and with no demand Hearth is ordinary FPoS. Verification is layered and stated, not obscured: attestation, statistical identity testing, watermarking, and committee re-execution each catch a different failure; the residual assumption is named at each phase; and impractical trustless

guarantees such as frontier-scale ZK are deferred rather than overclaimed. Assurance is staged to match tooling: borrowed hosted compute first, self-hosted verified compute next, verified orchestration last, each phase reusing the prior's primitives so nothing is rebuilt.

4 Consensus: Proof of Inference

4.1 The forging-weight modification

Fair Proof-of-Stake selects a block generator by a delay function: a node generates sooner the larger its generating balance b_i and the more favorable its Verifiable Random Function (VRF) hit, approximately

$$T_i = T_{\min} + C_1 \ln\left(1 - \frac{C_2 \ln(X_n / X_{\max})}{b_i \cdot A_n}\right),$$

where A_n is the network baseTarget (roughly 60 s blocks) and X_n is the VRF hit, unpredictable until the generation signature is published, which makes FPoS grinding-resistant.

Proof of Inference modifies exactly one term, the balance:

$$b_{\text{eff}}(i) = b_i \cdot \big(1 + \min(\text{workBoost}_i,\, B_{\max})\big), \qquad b_i \geq b_{\min}.$$

Here workBoost_i is node i's normalized, finalized verified-work over a trailing window, and $B_{\max}$ (e.g. 2–3×) caps the multiplier, bounding centralization pressure from accumulable compute. With no work, $\text{workBoost}_i = 0$ and the rule is vanilla FPoS. The VRF hit is untouched, so work cannot bias the randomness. And because $b_i \geq b_{\min}$, a stake-less operator earns no weight however much it computes.

4.2 Specifying workBoost

The forging rule leaves workBoost_i abstract; this subsection fixes it.

Definition 1 (workBoost)**.** *Let $r_i(E)$ be the USD value of Creds retired on node i's finalized, payer-gated, verified jobs in epoch E (§4.3). Smooth it over a trailing window with an exponential moving average,*

$$w_i(E) = (1 - \alpha)\, w_i(E-1) + \alpha\, r_i(E), \qquad \alpha \in (0, 1],$$

whose effective window is about $1/\alpha$ epochs. Fix a governance-set dust floor $\tau_w > 0$ and let $S(E) = \{\, j : w_j(E) \geq \tau_w,\ b_j \geq b_{\min} \,\}$ be the eligible set.

Normalize against the stake-weighted median work level

$$m(E) = \arg\max_{w \geq 0} \Big\{ \sum_{j \in S(E)} b_j \, \mathbf{1}[\, w_j(E) \geq w \,] \;\geq\; q \sum_{j \in S(E)} b_j \Big\}, \qquad q = \tfrac{1}{2},$$

the retired-work level reached by the median unit of stake rather than the median identity, with the stake quantile q (default $1/2$) governance-tunable and distinct from the boost half-saturation κ. Write $g_i(E) = w_i(E)/m(E)$ and set

$$\mathrm{workBoost}_i = B_{\max} \frac{g_i(E)}{g_i(E) + \kappa}, \qquad \kappa > 0.$$

The map $f(g) = B_{\max}\, g/(g + \kappa)$ is monotone, concave, with $f(0) = 0$ and $f(g) < B_{\max}$, so the $\min(\cdot, B_{\max})$ clamp is redundant; a node reaches half the cap at $g_i = \kappa$, and zero demand recovers vanilla FPoS. The map is scale-free, depending only on a node's work relative to its peers, and its concavity caps a large operator's advantage and the marginal payoff of self-dealing near saturation (§4.5). Bounding each node's contribution before it sets consensus influence follows Yuma normalization in Bittensor [15] and share-based proof-of-useful-work accounting [16, 17].

A plain count median over positive-work nodes would be cheap to move: an adversary registering many tiny dust servers with w_j just above zero drags it down, raising every node's g_i so a genuine large-work node reaches the concave cap with far less real work. The dust floor τ_w and stake-weighting close this. Because each eligible node enters $m(E)$ with weight $b_j \geq b_{\min}$ rather than one identity, the statistic tracks the median unit of stake, and shifting it past a challenger requires controlling roughly half the eligible stake mass, as costly as a stake-51% attack on the base chain [18]. Sybil identity count is irrelevant without a scarce-resource anchor [19]; only stake moves $m(E)$, so dust registration no longer perturbs the concave boost.

Lemma 2 (Bounded amplification)**.** *For every node i, $b_i \leq b_{\mathrm{eff}}(i) \leq b_i(1 + B_{\max})$. Hence a party holding stake share $\sigma = b_i / \sum_j b_j$ has forging-weight share $s_i = b_{\mathrm{eff}}(i) / \sum_j b_{\mathrm{eff}}(j) \leq (1 + B_{\max})\, \sigma$, and controlling a forging majority ($s_i > 1/2$) requires $\sigma > 1/(2 + B_{\max})$.*

The bound is immediate from $0 \leq \mathrm{workBoost}_i < B_{\max}$: non-negativity gives $\sum_j b_{\mathrm{eff}}(j) \geq \sum_j b_j$, hence $s_i \leq (1 + B_{\max})\sigma$, and the threshold follows by pitting the adversary's maximally boosted weight against the unboosted honest remainder. Forging weight is amplified by up to $(1 + B_{\max})$, but the stake needed for a forging majority falls only from $1/2$ to $1/(2 + B_{\max})$, a

factor of $1 + B_{\max}/2$; equivalently PoI retains a fraction $2/(2 + B_{\max})$ of the pure-stake attack cost, so at $B_{\max} = 2$ an attacker still needs more than a quarter of stake against the half required for pure FPoS [20]. Against a griefing or censorship adversary willing to burn capital rather than profit, the boost is purchasable up to burn-cost, so the threshold really is $\sigma > 1/(2 + B_{\max})$, a deliberate, bounded concession accepted for the mechanism's utility. Choosing $(\alpha, \kappa, B_{\max})$ and measuring the realized concentration of b_{eff} on live work remains empirical (§C.1).

4.3 Pre-finalization

Work that buys block-production probability must be fully verified first. Work performed in epoch E, with attestation checked and its verification and challenge window closed, is written to a canonical on-chain verified-work balance that boosts forging only in epoch $E+1$. Validators re-derive b_{eff} from finalized state at validation time, with no heavy verification at consensus speed, and a node's work is fixed before the next epoch's VRF hit is knowable, so it cannot be ground against the draw.

4.4 Penalty: generation-suspension, not confiscation

The base chain's leased proof-of-stake is deliberately non-slashable, and Hearth preserves that contract. A node proven to have cheated does not lose principal; its generating balance is suspended for a penalty period and its accumulated workBoost reset to zero. During suspension it cannot forge, earn the boost, or collect verification income, and the forfeited rewards fund the verifiers that caught it. Reversible and proportionate, this suits statistical verification with a non-zero false-positive rate: a wrongly suspended honest node recovers, whereas burned principal could not. Suspension escalates with repetition and scales with throughput, keeping the deterrent meaningful across stake sizes without confiscating delegators' funds.

4.5 Self-dealing the boost

Because verified inference raises forging share, a node can run real inference for itself to inflate its weight, and verification cannot catch genuine work with fabricated demand, so the defense is economic. The attack is structurally bounded: block issuance is fixed, so the boost only redistributes a fixed pie, $B_{\max}$ and a per-device cap bound the share work can buy, and the stake floor keeps it below pure-PoS security. The primary defense is the observable

payer-gate: each Cred carries an on-chain provenance tag of the key that last acquired it, and a retired Cred is boost-eligible only if that acquisition was an open-market purchase (an AMM or DEX swap) by a key distinct from the serving operator, so a wallet-to-wallet transfer between a Sybil's own keys does not reset eligibility. Creds an operator minted or holds itself pay for inference but never drive workBoost. The residual Sybil, routing payment through distinct keys it secretly controls, stays bounded by burn-cost: it must spend Creds burned on use, forgoing their USD value, and incur real compute cost, all to chase a capped boost tracking $\text{HRTH\_price} \cdot B_{\max}/c$: expensive, on-chain, and self-defeating.

Because verified work both pays rewards and raises forging weight, any verification break is also a consensus-tilting vector, but the same levers contain it: forging weight accrues only on burned, externally-paid Cred demand, capped at $B_{\max}$ over a trailing window, so any break must first be converted into sustained, capital-costly, externally-corroborated fake demand. The residual $B_{\max}$ amplification is the deliberate concession of §4.2: fully decoupling work from forging would negate Proof of Inference itself.

5 Token model and the inference market

5.1 HRTH and Creds

Hearth separates the asset that secures the network from the assets that price its output [21]. HRTH is the single network token: staked, forged, spent as gas, used in governance, and issued as the reward for verified work, suspended but never confiscated on proven misbehavior.

Creds price the network's output, phased to match what the network can back and discover at each stage. Phase 1 uses a single Cred: one USD-denominated, capacity-backed, burn-on-use inference credit, quoted and settled through an external stablecoin (USDC, USDT) and retired when spent. Relaying to a hosted aggregator (§7) means no per-model self-hosted capacity backs distinct claims and no network-discovered per-model price exists, since relative prices are administered by the aggregator's schedule. A single metered credit is the right primitive, a joint fungible unit of account with externally fixed relative prices, exactly the regime Diamandis et al. [22] describe as trading granular price discovery for simplicity. It also preserves the singleness of money [23]: one par-fungible unit and one deep pool, no liquidity fragmentation across thin per-model markets during bootstrap.

Per-model Creds emerge in Phase 2. Once nodes self-host confidential

GPUs and full logit-committee verification turns on (§8), each model's verified output becomes a distinct, attestable good with its own capacity backing and a price the network can discover. The single Cred graduates: it persists as the settlement numeraire, a base Cred, while each model gains a Cred-M that redeems for one standardized unit of output from pinned model M, carries no peg, and floats on internal AMMs rather than tracking the aggregator. This is the move to multidimensional, per-resource price discovery [22], worth trading singleness [23] for, but only now that there is a genuinely discovered price to represent. Per-model Creds phase in model-by-model as each gains self-hosted verified capacity, with no flag-day cutover (§8.4). Neither corner is chosen: one Cred forever cannot discover per-model prices and collapses toward a reseller echoing the aggregator, while many Creds from day one back and discover nothing in Phase 1 and fragment liquidity prematurely.

5.2 A market for verified inference

In Phase 1 the single Cred trades in one deep, par-fungible USD market; the per-model market is the Phase-2 picture. Because every Cred-M is then a claim on the same standardized, attested good (model M at a pinned hash and precision), per-model Creds trade as ordinary fungible tokens on decentralized exchanges and AMM pools, quoted in USD. Model-pinning and attestation turn "some inference" into a commodity with a precise specification, so each model's market price becomes the discovered price of real inference, the network's core product (§12).

Consumption is not an order book in the request path: a buyer acquires a Cred upstream and later spends it on a job, so the inference call is instant, any eligible node serving a holder by protocol rule with no per-request negotiation. On spend the Cred is retired: a governance-set majority fraction is burned [24] to tie each model's Cred supply to its throughput, the remainder routed to the verifier pool, and the serving node rewarded in newly issued HRTH. Serving is market-elective (§5.4), so a Cred is a redeemable claim, not a guarantee: a holder is entitled to inference whenever the price sustains willing supply.

5.3 Staking, leasing, and serving

In Phase 1 stakers earn the single Cred as a USD-denominated yield, minted against the network's realized verified capacity; in Phase 2 this becomes a per-model, capacity-weighted basket, each Cred-M minted in proportion to model M's realized verified capacity, so no Cred is issued beyond the

compute that backs it. Creds follow the stake's owner, yielding three roles. A passive staker serves nothing and earns the Cred basket alone. A lessor delegates HRTH to a serving node under a non-custodial lease, keeping the basket and additionally earning a share of that node's HRTH rewards. A serving node runs attested inference and must hold stake, owned or leased-in, of at least $b_{\min}$; it earns HRTH (issuance plus the forging boost) and shares part back to its lessors.

This closes the economic loop with no explicit Cred-to-HRTH peg. External demand paid in dollars for Creds becomes, since Creds are the yield on staked HRTH, the USD return on staking. Nodes must stake to serve, so HRTH's value sets the serving floor: if HRTH is too cheap, nodes withdraw, supply falls, Cred scarcity lifts the price and hence the yield and HRTH's value, and serving resumes, implicit and self-correcting.

5.4 The subsidy, the floor, and self-dealing

HRTH issuance lets honest providers serve below cash cost: a node earns HRTH (issuance plus the boost) for verified work, so it can accept a low Cred price and still profit. This liquidity mining makes Hearth inference cheaper than incumbents and attracts the operator who runs GPUs cheaply but has no wish to build a brand, for whom passing attestation is the entire go-to-market (§12). The one entry cost is stake: serving requires $b_{\min}$, so $b_{\min}$ is an economic dial set low enough not to choke that supply, not only a security parameter.

The serving floor is set by the market, not the protocol: a node serves only when its HRTH reward covers its cost and withdraws otherwise, so entry and exit discover the floor rather than the protocol measuring an unobservable marginal cost. What it enforces instead is the payer-gate of §4.5, an observable external-USD signal. The health metric governance watches is the ratio of organic USD revenue to HRTH subsidy value: rising, the market is real; flat, the network is renting volume.

The link degrades gracefully rather than collapsing: even with the subsidy gone, every Cred stays redeemable for real inference at its true cost, because the backing is servable GPU-time, not a promise about HRTH's value. The worst case is that Hearth inference costs what inference actually costs.

5.5 Governance and value capture

Within protocol-enforced bounds, governance sets the pinned models, emission, subsidy, the retirement split, $B_{\max}$, the work-to-cap scaling (§5.6), and

the verification parameters (§13); no vote can mint a Cred faster than its model's verified capacity, a constraint, not a policy. HRTH accrues value through gas and settlement, governed fee burns [25], and most durably the staking demand that real USD Cred-yield creates: as dollar-paid inference volume grows, so does the value of the HRTH that must be staked to earn the Creds that serve it.

5.6 Economic security (worked figures)

The forging subsidy is bounded and small, capping what manipulating it can be worth. At an HRTH price of \$0.20, 20% of a roughly 120M supply staked, a 4-coin block reward at roughly 60 s blocks, and $B_{\max} = 2$, the annual forging pool is about 2.10M HRTH, about \$420k (independent of supply), the staked base is about \$4.8M, and the base staking yield is about 8.8% before any boost or Cred yield. That \$420k pie bounds boost self-dealing: at a marginal serving cost of order $\$10^{-4}$ per job, manufacturing fake demand to farm the boost is break-even only for a roughly 5%-stake whale (about 2.5 GPU-years of real compute for about \$42k of boost) and a loss for everyone smaller, before the retirement-burn and payer-gate make a self-dealt job strictly loss-making. The deterrent scales with the pool and hence with HRTH price, so the work to reach the boost cap must track $\text{HRTH\_price} \cdot B_{\max}/c$, a governance-pegged value and the single most important parameter to encode. The observable that protects price discovery protects consensus: forging weight is bought only with burned, externally-paid Cred demand (§4.5), so tilting consensus through fabricated work is as uneconomic as self-dealing the fee.

6 Verification framework

This section specifies how correctness of inference is established. It applies in full in Phases 2 and 3; Phase 1 uses only the subset that needs no re-execution (§7.4).

6.1 What is verified, and why logits

The threats are substitution (serving a cheaper or different model than claimed) and fabrication. Comparing generated text is useless, since sampling makes honest runs differ while a substitute still produces plausible text. Verification targets logits instead: at each position a model emits a logit

vector that is a deterministic function of (model, input), all randomness living in the sampling that follows, so it fingerprints the exact model even when the sampled text looks identical.

6.2 The core check: teacher-forced, single-step logit comparison

The verifier never generates freely. During the original run the enclave commits, per output position, to a sketch of that position's logit vector, assembled into a Merkle root in the receipt. To check a VRF-selected position j, the node reveals the committed vector and its Merkle path; the verifier confirms the path opens against the root, then runs a single forward pass of the pinned model on the claimed prefix (prompt plus the node's tokens up to j), with no autoregressive decoding, comparing its logits against the committed ones within tolerance. One forward pass is cheap relative to the generation it verifies.

6.3 Defeating nondeterminism

Even one forward pass is not bit-identical across machines. Sampling nondeterminism is eliminated by construction, since the check never samples. Batch and sequence nondeterminism, the dominant source, is the dependence of a sequence's logits on its batch neighbours; batch-invariant kernels [26] remove it but forfeit the batching that makes inference cheap and tax every job. To keep overhead below the conservatively-estimated honest cost spread (§12), a node by default serves on optimized kernels and commits those logits, and the batch-order gap from a verifier's single-sequence re-run is absorbed, with hardware and precision noise, into a tolerance τ: attestation pins kernel version and precision, and comparison uses tolerant matching (top-k agreement or LSH distance under τ) rather than equality.

A wider τ weakens each token as a signal, but the verdict is never per-token. Hearth anchors τ to two published operating points: He et al. show batch-invariant kernels make a forward pass bitwise reproducible, giving $\tau = 0$ for the regulated tier that opts in. TOPLOC reports a top-k locality-sensitive commitment detecting changes to model, prompt, or precision with 100% accuracy and no false positives on its benchmark, tolerating benign cross-batch and cross-hardware reordering, and DiFR characterizes the benign logit-difference distribution across hardware [26, 10, 11]. Inside that band a substitute must match the committed top-k ranks at every sampled position, excluding model swaps, quantization, and precision downgrades;

what survives, diverging only in the tail, is the target of the τ-independent distributional layer (§6.5) [12].

6.4 The verdict: attested-verifier committee

For each sampled job a VRF selects a committee of attested verifiers, each running the check inside its own TEE in the pinned environment and signing in-enclave. A supermajority decides; a split expands the committee until one forms or, at a cap, defaults to no penalty (fail-safe). An adversary holding a Byzantine share of verifier stake could withhold agreement to drive a job it wishes to shield to the cap, turning the liveness fallback into an escape hatch. The default is therefore value-conditioned: high-value jobs fail closed, withholding settlement or retrying rather than defaulting to no penalty. A verifier in the losing minority of a clear supermajority is suspended, making rubber-stamping and minority collusion negative-EV, and verifiers commit to their re-derived logit sketch before any are revealed, so none can echo a peer while the catch-bounty exceeds one re-execution's cost.

The VRF samples n verifiers, and a wrong verdict requires the adversary to hold $\lceil 2n/3 \rceil$ seats, which under an adversarial fraction f of bonded stake has probability $\Pr[\mathrm{Bin}(n, f) \geq \lceil 2n/3 \rceil]$ [27]. At the Byzantine threshold $f = 1/3$ this per-job capture probability, bounding both false suspension and missed cheating, is 8.5×10^{-3} at $n = 15$, 1.8×10^{-3} at $n = 21$, 4.2×10^{-4} at $n = 25$, and 9.4×10^{-5} at $n = 31$. So $n_{\mathrm{default}} = 21$, with a floor $n_{\min} = 15$ for low-value classes and escalation toward $n = 31$ for high-value or regulated tiers. Even odds needs f near $2/3$, comparable to attacking consensus itself. The binomial bound assumes independent seat selection; VRF sampling over the bonded-verifier set delivers this unless a single entity secretly controls many seats. Governance's TEE-vendor cap (§13) bounds correlated attestation-root failure but not operator collusion, so bounding one entity's effective seat share against a Sybil operator is a residual open item (§C.2).

The committee checks a deterministic property, whether the committed logits match the pinned model on the prefix, not answer quality, so shared preferences cannot manufacture false consensus as in quality-scored networks: honest verifiers converge on arithmetic, not taste. The layers fail independently (§11.2): a TEE break degrades correctness to the honest-majority guarantee rather than defeating it, and only a correlated attestation-root compromise, addressed by heterogeneous roots (§13) and the deferred floor (§6.7), escapes both.

6.5 The defense-in-depth stack

Around the committee sit complementary layers, each catching a distinct cheater: TEE attestation catches the lazy operator, watermarking the per-output substitute, cross-node consistency in Phase 1 (§7.4), honeypots the rubber-stamping verifier, and Model Equality Testing runs a continuous text-only distributional audit against the systematic substitute. All require verifiable jobs to run under declared decoding parameters and a committed seed.

The distributional layer rests on a power argument. The only substitute surviving the top-k check reproduces the committed ranks yet diverges in the tail of the next-token distribution; Model Equality Testing targets that residue with a Maximum Mean Discrepancy two-sample test over sampled completions, independent of τ, and its authors report detecting quantization, substitution, and fine-tuning across production APIs from hundreds of completions [12]. Hence a dichotomy: a tail divergence either shifts the sampled-output distribution enough to flag it, or it is so slight that outputs are statistically indistinguishable from the claimed model's, leaving no cheaper model and no gain. Latency follows the sampling rate: the committee accrues about $p\,m$ completions per epoch, for p of 1–5% (§6.6) and per-epoch job count m, so reaching MET's sample N takes about $N/(p\,m)$ epochs, and a node serving thousands of jobs per epoch is flagged within $O(1)$ epochs; exact N is the open item of §C.2.

6.6 What touches the chain, and sampling

Individual inference calls are never posted on chain. Each node commits one Merkle root of its epoch's receipts plus aggregate work and a data-availability pointer, an $O(1)$ per-node per-epoch footprint. Receipts (hashes, the TEE quote, the logit commitment) live off-chain under that root; prompts and outputs stay private, revealed only if sampled. Only a random fraction p of jobs is checked: a node faking a fraction φ evades detection in an epoch with probability about $(1-p)^{\varphi \cdot m}$, so even $p \approx$ 1–5% catches systematic cheating with near-certainty, while suspension on a single catch makes it negative-EV. Receipt availability is enforced through settlement: a node is paid only for jobs whose receipts it can produce when sampled (§7.2), so withholding data forfeits payment. Both this sampling bound and the Model Equality Testing dichotomy (§6.5) assume substitution spread across traffic; they weaken against an adversary that serves the honest model on nearly all prompts and substitutes only on a narrow high-value class, since a

small φ slips between per-job sampling and the aggregate audit. Sampling is therefore value-weighted: high-fee jobs are drawn at a raised rate, so the expected caught value stays bounded even as φ shrinks. Hearth thus defends systematic substitution unconditionally, while targeted high-value substitution is bounded by value-weighted sampling and, in the top tier, by the deferred trustless floor (§6.7).

6.7 Deferred trustless floor

Two mechanisms would survive even a compromised attestation root and are recorded as the upgrade path, out of scope initially. A single-shot ZK proof on dispute (groth16 on bn254 [28], verified cheaply on-chain) is sound and needs no determinism-matching because the circuit defines the arithmetic, but proving a frontier forward pass remains far too slow [4]. An interactive bisection fraud proof [8, 9] narrows a dispute to one primitive operation the chain executes itself, but requires canonical bit-exact arithmetic that GPU floating point lacks, so it becomes viable only under fully-specified integer or fixed-point inference. Both become practical now for the sub-1B router of Phase 3 (§9), inverting the usual assurance-by-cost.

7 Phase 1: Hearth Relay

The first phase ships the full economic and consensus machinery against real demand while deferring the hardest cryptography. Nodes do not run models; they relay requests to open-weights models hosted on an aggregator (OpenRouter) and forward the replies. Phase 1 is a decentralized incentivized gateway in front of a trusted API, not yet a verified-inference network.

7.1 Rationale

The token economics, marketplace, routing and reputation layer, FPoS forging-boost integration, and statistical-audit pipeline need no node to own a GPU. Borrowing hosted compute de-risks everything but the inference cryptography and bootstraps a staked HRTH base and validator set before the network secures high value: the compute is borrowed, the chain, token, and incentives real.

7.2 Architecture and request flow

A node registers, stakes HRTH, and runs its relay logic inside a CPU enclave (Intel TDX or SGX). The enclave terminates the TLS connection to the aggregator and holds the API key sealed, so the operator can neither read the traffic nor bypass the enclave.

A request proceeds in three steps. The user picks a model and node (or the protocol routes by price, latency, and reputation), signs the request, and escrows the Cred; signing the request rather than the reply proves a distinct payer paid real USD, the basis of the self-dealing defense. The enclave calls the aggregator with the declared model and fixed decoding parameters and returns $(output, receipt)$ with its attestation over the transcript. Settlement releases on attested delivery, not a user signature: the enclave's proof of a valid reply claims the escrowed credit, auto-finalizing after a short window into a Cred retirement (burn plus the verifier pool, §5.2), payment to the node in newly issued HRTH, and a next-epoch workBoost credit. A dissatisfied user may instead file a challenge within the window against the attested receipt, carrying a small anti-spam bond.

The receipt is the base receipt without a local forward-pass quote, since none ran:

```
relay_receipt = { H(request), H(output), model_id, decode_params,
                  timestamp, tls_attestation, node_sig, payer_sig }
```

The `tls_attestation` is an authenticated-data-feed proof in the Town Crier and zkTLS lineage: the enclave quote binds the request it sent (including the model id), the aggregator's TLS certificate, and the response. The `payer_sig` proves a distinct party paid real USD (§4.5).

This closes a settlement attack: were payment to depend on the user counter-signing after delivery, a user could take a valid answer and withhold the signature, leaving the node out-of-pocket. Anchoring release on attestation removes the asymmetry, the node proving delivery with its enclave quote and the user proving non-delivery with a signed timeout receipt (§7.4).

7.3 Node and enclave: two processes, one trust boundary

The relay TEE does not run inside the L1 node's enclave, and should not: they are two cooperating processes with different guarantees, joined by the signed receipt. Only the relay logic belongs in the TEE, whose single job is to make one claim trustworthy, that this request was sent to OpenRouter

over authenticated TLS and this response came back; it is a small stable codebase fit to be attested. The L1 node runs outside as an ordinary process, its block production, mempool, consensus gossip, registry, and token ledger secured by stake and the PoI rules. Placing a frequently-updated node inside an enclave would bloat the measurement and force re-attestation on every release, buying nothing, since consensus needs neither confidentiality nor a hardware quote.

The enclave hands the receipt to the node, which anchors it in the per-epoch Merkle root on chain; the node cannot forge it, lacking the enclave key. The two roles need not share a machine, since the receipt format is transport-independent, though v1 co-locates them. At registration the enclave generates its signing keypair sealed and the operator links it to their stake, so a receipt proves that this staked operator's enclave performed the relay, attributing the work to the correct node's workBoost.

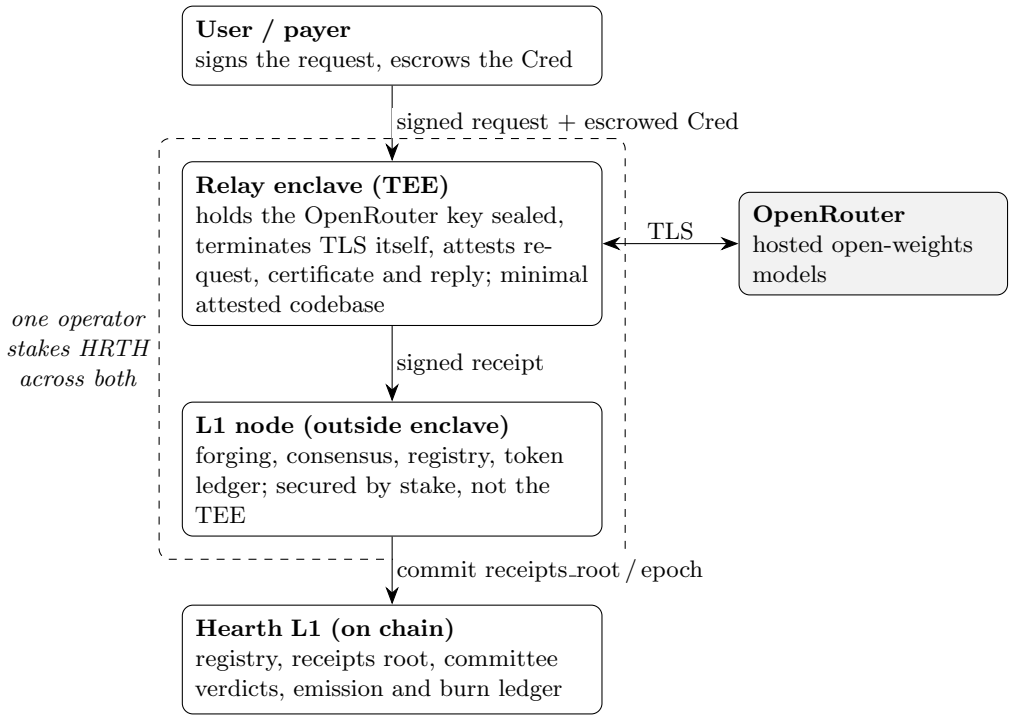

Figure 1: Phase 1 topology. The relay enclave attests the conversation with the aggregator and signs a receipt; the L1 node, secured by stake rather than attestation, commits per-epoch receipt roots on chain.

7.4 Verification in Phase 1

The computation happens behind the aggregator's TLS endpoint, which Hearth cannot attest, so verification splits into what the TEE makes trustless and what verifiers must audit.

The TEE makes the node trustless: since settlement requires the enclave's `tls_attestation`, a relay node cannot fabricate a reply, alter the response, claim model X while requesting model Y, or skip the paid call, as each leaves no valid attestation. The most worrying relay attacks are closed by construction.

What verifiers must audit is the aggregator, itself a router to upstream providers that may quantize or substitute; the enclave witnessed the conversation, not the computation. A light committee audits this statistically without re-execution: Model Equality Testing compares a node's output distribution against a reference of the claimed model [12, 13], cross-node consistency flags the one node whose distribution drifts from peers calling the same model, and honeypot probes carry pre-characterized correct-model behavior. A market-wide divergence signals an aggregator model change and triggers a delisting flag rather than mass suspension.

Liveness is enforced economically: a node that never serves produces no attestation and earns nothing, reputation-weighted routing starves it of traffic, and a client's signed timeout receipt makes non-service provable. Verifier probes double as undetectable liveness tests, so a node cannot selectively drop expensive jobs without sometimes failing an audit; sustained non-service escalates to generation-suspension, while transient failures only decay reputation.

7.5 Phase-1 economics

Because work is an API call rather than a local forward pass, the compute-cost floor that deters boost self-dealing is lower and the node's cost is fiat paid to the aggregator. Two adjustments follow. workBoost is driven by the Cred value retired on a node's jobs, not raw job count, so the boost cap pegs to the observable aggregator fee, and with the payer-gate (§4.5) a self-dealer must retire real externally-paid value to move its boost. And a node paid in HRTH must sell most of it to cover the fiat bill, creating sell pressure offset by the usage-scaling retirement-burn, so a healthy Phase 1 tilts toward genuine paid demand.

7.6 Trust model and limitations

Phase 1 rests on three assumptions (§11.1): sound CPU TEE attestation, an aggregator honest about which model it serves (audited, not assumed), and an honest bonded-verifier majority. Relative to a plain relay network this removes anonymous nodes from the trust set, leaving a single named party under continuous audit. Its limitations, all removed by Phase 2: no prompt privacy (the aggregator sees plaintext), no catching a one-off substitution (only systematic deviation), and the aggregator as a central point of failure.

8 Phase 2: Hearth Compute

The second phase moves inference onto the network's own hardware and turns on the full verification framework of §6, where Hearth becomes a verified-inference network in the strong sense.

8.1 Self-hosted confidential inference

Nodes bring their own accelerators and run open-weights models inside confidential-computing TEEs (NVIDIA H100 or Blackwell), weights pinned by `model_hash` in the on-chain registry and the signing keypair generated inside the enclave. Every response is signed in-enclave over the full transcript, closing the gap between "a model is loaded in an enclave" and "the endpoint actually routed this request to it." Attesting the loaded weight hash gives model-pinning against bait-and-switch for free, and one attestable chip equals one identity, so Sybil attacks cost real GPUs.

8.2 Full verification turns on

The teacher-forced logit committee (§6.2–§6.4) now applies per job: verifiers re-run sampled forward passes in the pinned precision class and reach a supermajority verdict on the tolerant logit match (§6.3), backed by the defense-in-depth stack of TEE attestation, watermarking, Model Equality Testing, and honeypots (§6.5). The aggregator leaves the trust set: only attested nodes the committee can re-execute remain. Phase 1's cross-node-consistency check stays useful but is no longer load-bearing once per-job re-execution is available.

8.3 Privacy

Because inference runs inside the enclave on hardware the network controls, prompt and output confidentiality, absent in Phase 1, becomes available: the enclave decrypts, computes, and returns without exposing plaintext to the operator. For audit, attestation plus output-only statistical checks apply, with requester-consented or validator-in-TEE re-execution where stronger assurance is needed (§6).

8.4 Migration and the mixed network

Phase 2 needs no flag-day cutover. The network runs mixed: relay and self-hosted attested nodes coexist, and the protocol prices the difference. Attested nodes earn a higher boost and are eligible for premium and privacy-sensitive jobs relay nodes cannot serve, a continuous incentive to migrate. As self-hosted capacity grows, dependence on the aggregator declines until it can be retired, and the per-model Creds (§5.1) turn on model-by-model as each model gains self-hosted verified capacity.

8.5 Trust model

Phase 2 rests on two independent assumptions, sound hardware attestation and an honest bonded-verifier majority (§11.1), with no third-party API in the trust set, so correctness survives either failing alone (§11.2). The sole residual, a correlated attestation-root compromise, is closed by governance-optional heterogeneous roots (§13) and the deferred floor (§6.7), bridging to Phase 3.

9 Phase 3: Hearth Coordinate

The third phase makes orchestration a first-class, verifiable citizen of the same economy. A governance-pinned multi-model router, a small coordinator in the family of learned multi-model routers, decides which model serves each step of a composed task and in what role, verified by the same machinery. Appendix B gives the full specification; the essentials follow.

9.1 The composed job

A composed job generalizes a single job to a short directed acyclic graph. Under a turn budget K, the router node holds the pinned router θ in its

TEE; at each turn it forms the transcript prefix, runs one forward pass at temperature zero, and emits a deterministic decision $(model\_class, role)$. Each chosen class is served by an ordinary Phase-2 verified job with its own attestation, receipt, and logit commitment, until a verifier turn accepts or the budget is exhausted. The job emits one route receipt chaining the decisions and referencing the worker receipts, sitting as a sub-tree under the node's per-epoch Merkle root, so consensus state is unchanged.

9.2 Routing is inference, so PoI already verifies it

A routing decision is a forward pass on a pinned model, so the teacher-forced logit check applies directly and more cleanly: there is no decoded sequence, only a hidden state mapped to head logits. A VRF-sampled turn is checked as any job: the router reveals the head-logit vector and Merkle path, and a verifier loads θ by hash and confirms both that the argmax reproduces the decision and that the logits match within τ.

9.3 Operator-independence and the verification asymmetry

A pinned router cannot be tilted by its operator: the decision is a deterministic function of $(\theta, transcript)$, and perturbing a single weight produces a `router_hash` mismatch or a τ-failing logit. The self-preferential-routing attack that plagues secret orchestrators is structurally absent, since the worker for a chosen class is assigned by VRF or protocol rule, not the router operator. The router is the most value-critical computation in a composed job yet the cheapest to verify (a sub-1B forward pass): the trustless floor Phase 2 defers for frontier workers, a ZK or bisection fraud proof over fixed-point arithmetic, is practical now for the router, so the routing layer can carry a guarantee that survives even a compromised attestation root while its workers carry the committee guarantee.

9.4 Reward and the coordination fee

Workers earn as in Phase 2, recycled value and work-units feeding workBoost. The router earns a coordination fee, a governance-capped slice (about 5–10%) of the job's total fee, paid for a verifiable route rather than the trivial compute of its forward pass; it earns fee, not work-units, so it cannot become a backdoor to forging weight. Self-dealing the fee is deterred as boost self-dealing is: with the router pinned, fake demand yields only the fee paid to oneself less the burn, and still requires a distinct co-signing payer who paid

external USD. A router proven to have emitted a decision inconsistent with θ is suspended like any cheating node.

9.5 What is sold, and to whom

The router is open, pinned, and replicable, so there is no margin in routing intelligence and none is sought. What is sold is a provably honest route: a machine-checkable receipt that the governance-approved coordinator, and only it, decided where a job went and that each worker served the model it claimed, something hosted orchestrators cannot offer without revealing the routing they monetize. It is load-bearing for regulated inference, where opacity is a procurement blocker, and for agent-to-agent compute commerce, where one autonomous agent paying another has no human to extend trust, so the route receipt and payer co-signature are the trust rather than a premium feature.

9.6 Trust model

Phase 3 inherits Phase 2's trust model and strengthens it on the most critical layer: the coordination decision can carry an optionally trustless guarantee (ZK or bisection on the small router), even as the workers rest on the attested-committee guarantee.

10 Beyond inference: the generalized market

Everything above reduces to one primitive and one economy. The primitive is attested computation: a workload runs in an attested environment against pinned artifacts, emits a signed receipt, and settles under the committee and statistical backstops of §6. The economy is the HRTH/Cred pair of §5: capacity-backed credits retired on use, staking converting external paid demand into the yield that secures the chain. Nothing in either is specific to text generation, so a further workload is admitted when it passes the three tests inference passes: verifiable by the existing primitives; carrying external paid demand from a distinct payer, so the payer-gate (§4.5) and the organic-revenue health metric (§5.4) extend unchanged; and standardizable, pinnable to a specification precise enough that its credit is fungible. This section is a map, not a schedule: each workload ships when its verification cost fits under the value it protects. The Cred doubles as a credibly neutral access instrument: a pinned model cannot be deprecated by a product decision (§1),

access requires only a Cred bought on an open venue, and any distributive program can grant it at cost without building serving infrastructure; person-level distribution requires sybil-resistant identity, outside the protocol's scope.

10.1 Near workloads

Nearest are workloads that are the same forward pass in different clothing. An embedding is a single deterministic forward pass, the router-verification case of §9.2 at scale, checked by the same teacher-forced comparison (§6.3); a Merkle-committed vector index extends it to verified retrieval, a receipt that an answer was retrieved from a committed corpus and generated by the pinned model. Because verification's fixed cost does not shrink with an embedding's price, sub-cent jobs settle at batch granularity (one commitment and sampling unit per committed batch), keeping overhead under the spread of §12 at the price of coarser sampling. Verified evaluation runs committed test suites under attestation against pinned models; a committed suite is discoverable through sampled verification and eventually trained on, so suites are epoch-fresh, produced by a pinned generator from a committed recipe and retired after one use, reducing evaluation neutrality to a governance-auditable artifact (§13). Verified synthetic data is bulk inference plus a per-dataset provenance receipt attesting the generating model, parameters, and deduplication against a committed manifest.

10.2 Adjacent workloads

Adjacent extensions sell the receipt rather than the forward pass. Weight availability erasure-codes pinned weights across the mesh under availability sampling, turning the deprecation risk of §1 into a guarantee; admission is license-gated to redistributable weights, and the guarantee is bounded: governance can unpin a model, halting emission and replication payments, but cannot make a permissionless mesh forget bytes it distributed. Creator royalties add an optional third leg to the retirement split (§5.2); attestation proves the served weights are the pinned weights but cannot establish authorship across fine-tunes and merges, so the royalty is a governance policy with disclosed attribution limits rather than a protocol-verified property. Exported over light clients, an attested receipt makes Hearth a verified-AI oracle for other chains, with one built-in distinction: the domestic regime is statistical, priced for flows of jobs (§6.6), whereas an oracle call gating a single settlement is an event, so high-value exports purchase mandatory

full-committee verification or a zero-knowledge proof. Agent execution runs the loop itself attested, with receipted tool calls and escrowed settlement reusing §7.2; the receipt proves the agent's code ran as pinned, not that its judgment was good or its external inputs honest, so agent quality rides reputation.

10.3 Frontier workloads

Fine-tuning is admitted as a commission market one assurance rung down: a buyer escrows against a committed data manifest and evaluation suite, a staked node executes under attestation, and settlement releases on attested completion plus a passing eval, since there is no training analog of the cheap teacher-forced check and verifiable training remains its own research program [29]. Completed training jobs earn their fee but never feed workBoost: a lumpy, self-payable, expensive-to-verify signal is what the self-dealing defense (§4.5) exists to keep out of forging weight, so a trained model influences consensus only downstream, if governance pins it and real payer-gated demand for it accrues boost to its servers. Data provision is priced on provable properties (integrity, provenance, licensing, deduplication) and deliberately not on contribution quality, for which no committee-checkable predicate separates valuable data from poison; a dataset's worth emerges from the eval-gated performance of models trained on it. Distributed frontier inference, sharding one model across the wide-area mesh, is admitted only as asynchronous, throughput-priced batch work, where interconnect latency amortizes; interactive sharded inference over a WAN is uncompetitive and not offered.

10.4 The assurance ladder and the invariants

Every workload climbs the same three rungs and states which one it stands on: hardware attestation with committee and statistical backstops (§11.2); optimistic bisection over deterministic arithmetic (§6.7), practical first for small artifacts (router, embeddings, evaluation suites); and zero-knowledge proofs, adopted per workload as proving cost falls under the spread its market tolerates [4]. The climb is per workload and per value tier, and does not fragment the market: the Cred prices the computation and assurance prices the checking, so each good keeps one Cred defined at the base guarantee, with higher assurance purchased as a fee at request time and fungibility (§5.2) preserved; upgrades are unilateral, while downgrading a deployed assurance level requires a governance supermajority with public disclosure. What makes

the expansion admissible is that the economics are invariant: every priced good gets a capacity-backed, burn-on-use credit, grouped by governance where per-good markets would fragment liquidity (§C.3); every boost-eligible unit of work passes the payer-gate; serving any workload requires stake (§5.3); suspension, never confiscation, remains the penalty (§4.4); and one health metric, organic USD revenue against subsidy value, governs the whole basket. The network remains one market, for attested computation, quoted in as many credits as it has goods worth pricing.

11 Trust progression and security summary

The three phases form a monotonic ladder of assurance, each reusing the prior phase's primitives (enclave, committee, VRF assignment, generation-suspension, the HRTH and Cred rails), so nothing is rebuilt.

Phase	**Compute runs on**	**Node trust**	**Residual trust**	**Privacy**	**New capability**
1 (Relay)	hosted aggregator	TEE-attested	aggregator (audited) plus verifier majority	none	ship economy and consensus on real demand
2 (Compute)	self-hosted GPU plus TEE	TEE-attested	verifier majority only	yes	verified inference, no third party
3 (Coordinate)	self-hosted plus pinned router	TEE-attested	verifier majority; router optionally trustless	yes	verified orchestration, agent-to-agent commerce

Table 1: The three phases as a monotonic ladder of assurance.

Across all phases, security is anchored in HRTH stake, Sybil resistance is hardware- and capital-bound, and a verification false positive costs at worst a reversible suspension, not confiscation. Residual risks are stated in §15.

11.1 The trust assumptions, stated

The live network (Phases 1 and 2) is not trustless. Its correctness rests on a small set of explicit assumptions, each with a bounded blast radius and a backstop.

Assumption	If it breaks	Blast radius	Backstop
Attestation soundness	forged quote or side-channel extraction	a node fabricates verified work, or serves a substituted model under a valid-looking quote	the statistical layers (MET, cross-node consistency) are TEE-independent and catch any systematic substitution; the workBoost trailing-window, burn, and payer-gate bound the economic and consensus gain; governance-optional heterogeneous roots break a correlated compromise
Honest verifier majority	Byzantine majority of a committee	false verdicts on sampled jobs	commit-reveal forces independent recomputation; the catch-bounty exceeds recompute cost; minority-suspension; VRF assignment; bounded early by a minimum committee size and lower-value settlement
Receipt availability	node withholds receipt data	a sampled job cannot be re-checked	settlement is gated on producible receipts (§6.6): withholding forfeits payment, indistinguishable from never serving
Aggregator honesty (Phase 1 only)	aggregator serves the wrong model behind TLS	substitution upstream of the node	MET and cross-node consistency flag it; the aggregator leaves the trust set entirely in Phase 2
Honest governance	captured or vote-bought quorum	every safety knob (emission, $B_{\max}$, the retirement/verifier-pool split, the pinned models and routers, the optional TEE-vendor cap), hence every security bound above	the knobs sit inside protocol-enforced bounds behind quorum and time-locks, so a captured vote still cannot exceed the hard constraints: it cannot mint a Cred faster than verified capacity or raise $B_{\max}$ past its ceiling

Table 2: Explicit trust assumptions of the live network, each with a bounded blast radius and a backstop.

11.2 Independence of the layers

The security argument does not depend on the TEE being unbreakable but on two layers that fail independently. Profitably exploiting a broken attestation root requires systematic substitution, which moves the output distribution and is caught by Model Equality Testing and cross-node consistency, tests that run outside the enclave. A TEE break still permits the single-shot tampering kept within distributional tolerance, but by the sampling argument it earns negligible reward and cannot move forging weight at scale. The profitable attack triggers the layer the break cannot touch; the evasive attack is not profitable. A statistical catch escalates to generation-suspension and a workBoost reset, never confiscation, because the verdict is probabilistic (§4.4).

The one case the live network does not cover until the trustless floor (§6.7) is a correlated compromise of the attestation root paired with a substitution subtle enough to evade the distributional tests, a strictly harder attack than either alone. Two measures close it: governance-optional heterogeneous attestation roots (§13), so no single vendor's compromise covers a supermajority, and the ZK or bisection floor, which lands first on the small Phase-3 router. Correctness holds if the attestation is sound or an honest majority statistically detects deviation; only the simultaneous failure of both defeats it, the standard and irreducible assumption.

12 Market design and competitive positioning

Verified-inference-as-consensus is no longer an unoccupied idea, and Hearth's defensibility rests not on the idea but on what it is built into: an open market.

Several well-funded efforts share adjacent ground. Ambient is a Solana-fork L1 whose Proof-of-Logits consensus turns the inference of a single network-wide model into the mining puzzle, fingerprinting logits to verify that inference [30]. Gonka, an adjacent decentralized-inference L1, validates inference by honest-majority re-execution plus randomized spot-checks and weights consensus by a transformer proof-of-work benchmark, so voting power is compute- rather than stake-weighted [31]; Hearth instead keeps stake as the floor with a capped work multiplier and verifies by teacher-forced logit comparison, committee, and MET. Bittensor scores and rewards model quality across subnets [15], Gensyn targets verifiable training [29], restaking platforms add economic guarantees over hosted inference, and confidential-

compute L1s provide TEE-backed privacy. Each verifies or prices something in AI; none runs an open market that discovers the price of verified inference across many models.

These rivals converge on proving an output and diverge on what that proof buys. EigenAI [32] and Ritual [33] make inference deterministic and re-checkable, Inference Labs proves it in zero knowledge [34], Allora scores it [35], and Aizel [36] and 0G [37] fold verification into a broader AI stack. A separate proof-of-useful-work lineage, Pearl [17], makes the inference itself the consensus puzzle, though an empirical audit finds most of that work is not externally useful [38]. Hearth differs on the economic primitive rather than the proof: a passed attestation is a capped multiplier on FPoS forging weight (§4.1), not the block puzzle, and value accrues in USD-denominated Creds (§5.1). Two-token burn-on-use is not new (Venice [39] predates Hearth), but coupling it to verified-inference-weighted forging is.

Hearth is not a cheaper verified-inference provider: a single-model design like Ambient's amortizes one model's cost better than a multi-model fleet, and competing on cost-per-token for one model is a losing fight this paper does not claim. Hearth is the market for verified inference: in the Phase-2 endstate any model that can be pinned and attested becomes a per-model Cred whose open-venue price is the public, on-chain price of its verified output. Competition runs through supply: a cheaper provider expands a model's capacity, capacity-weighted emission mints more of its Cred, and the added supply pushes the price down.

Incumbents structurally cannot occupy this position. A single-model chain cannot host a multi-model market, and Hearth is honestly a metered gateway in Phase 1, reselling at the aggregator's administered prices, discovering prices only in Phase 2 when the network itself is the verified source and each model's Cred floats on an internal venue. A reseller that never makes that move cannot prove the model it served [1], so it cannot make inference fungible enough to trade; compute marketplaces price the input, GPU-hours, leaving the buyer to bear substitution risk on the output. Verification is the enabling layer: it turns an unverifiable service into a fungible good, and a market in a good one cannot verify discovers only the price of the most convincing lie.

The durable advantage is the removal of the distribution tax. An independent operator running cheap GPUs must today build a commercial surface (API, billing, brand, support) or list on a margin-taking aggregator and still market themselves. On Hearth, passing attestation is market access: demand routes on price, though serving requires staking HRTH (§5.3), so the wedge

is a calibrated stake, not frictionless entry. This admits pure-infrastructure supply that cannot appear on a posted-price platform, and the HRTH subsidy (§5.4) accelerates it.

The market is worth running only if honest providers differ in cost, and they do: same-model endpoint prices on public aggregators already span several-fold across honest providers [40], with the underlying GPU-hour rates dispersing comparably [41], driven by power and region, owned-versus-rented hardware, and idle capacity dumped at marginal cost. We take a spread on the order of 30% as a deliberately conservative lower bound on this dispersion, not a measured constant. That spread is load-bearing twice: it is the efficiency an exchange routes to buyers, and it is the ceiling verification must fit under, since proving a model's identity for more per job than the spread would make the market unable to afford its integrity. Hence batch-invariance is optional (§6.3): a kernel tax of 10–30% would consume the spread, whereas the chosen design costs a sub-one-percent commit per job plus a sampled re-execution. The overhead argument requires only that the spread exceed this commit-plus-re-execution cost, so it holds comfortably as long as the real spread is anywhere near or above the conservative 30% figure.

Phase 3's verified router (§9) is not a separate product but a derivative market on the spot markets: once each model's verified output has a price, routing across models and settling with a bonded receipt is the natural next venue, serving segments where the proof is the precondition of the transaction, regulated inference and agent-to-agent commerce.

Building on a mature Fair Proof-of-Stake base [5] is an engineering-velocity choice, not positioning: the market thesis would hold on any capable base.

13 Governance

HRTH holders govern, within protocol-enforced bounds, the parameters shaping the economy and security surface: Cred emission (per-model from Phase 2), the forging-boost cap $B_{\max}$, the retirement split (burn versus verifier pool), the work-to-cap scaling that pegs the self-dealing deterrent to HRTH price, the pinned models and (Phase 3) routers, and an optional cap on any single TEE vendor's share, off at launch and available to tighten as staked value grows. That these safety-critical parameters sit inside protocol-enforced bounds is a constraint, not a policy: quorum and time-locks defend the vote against vote-buying or flash-loan-style capture, and the most dangerous knobs

move only within hard limits and behind a delay, so a captured quorum still cannot mint a Cred faster than verified capacity or push $B_{\max}$ past its ceiling. Pinned models and routers are public, replicable artifacts evaluated on published task and reward suites, so a biased coordinator is detectable before it is pinned; the residual question Phase 3 raises is not whether the operator is honest but whether the pinned objective is well-specified.

14 Roadmap

The phases are sequential in their hard dependencies but overlapping in delivery. Phase 1 (Relay) ships the forked chain, HRTH and the single Cred, the FPoS forging-boost integration, the marketplace and reputation layer, the statistical-audit committee, and CPU-enclave relay, establishing usage and a staked base. Phase 2 (Compute) adds confidential-GPU inference, the teacher-forced logit committee with tolerant matching (§6.3), the full defense-in-depth stack, and the per-model Creds it can now back, running mixed with Phase-1 relay nodes until self-hosted capacity retires the aggregator. Phase 3 (Coordinate) adds the pinned router, composed jobs and route receipts, the coordination fee, and, uniquely at sub-1B scale, a trustless floor on the routing layer that later extends to workers as ZK proving for larger models matures. Past the three phases, further workloads join the same market under the admission rule and assurance ladder of §10.

15 Limitations and open problems

Several risks are intrinsic and stated rather than hidden. A freshly forked chain's proof-of-stake security is bounded by staked-HRTH value, small at launch, so high-value settlement is gated by a governance-set minimum-staked-value threshold, below which the protocol disables it; the intended trajectory has staked value lead settled value, with Phase 1 growing the staked base and usage first. The logit-matching threshold τ must separate honest rounding from real substitution, an empirical, fleet-dependent separation whose mis-calibration risks false positives or negatives, mitigated by statistical aggregation and a pinned hardware and precision class. Correctness behind the aggregator's endpoint in Phase 1 is audited, not proven, until Phase 2 removes it.

The deepest caveat is that trust is relocated, not removed. Hardware attestation has a live history of side-channel and microarchitectural attacks,

so correctness rests on the two independently-failing layers of §11.2: it holds if attestation is sound or an honest verifier majority statistically detects deviation. The residual, a correlated root compromise paired with a distribution-evading substitution, is unmitigated until the trustless floor lands and partly closed by governance-optional heterogeneous roots.

The economic loop through staking (§5.3) is genuinely reflexive: it self-corrects with lag, so a sharp demand drop can overshoot HRTH downward before Cred scarcity pulls it back, degrading gracefully because Creds stay redeemable at true cost (§5.4). Serving requires locked HRTH, a capital cost to the cheap-infrastructure supply the market courts, so $b_{\min}$ must stay low enough not to choke it. Verification overhead must stay well below the honest cost spread it protects, itself a conservative assumption backstopped by observed provider price dispersion and still to be measured on Hearth's own fleet; keeping overhead beneath it as models scale is an open constraint, not a settled result.

A few risks are external or structural. The verified-inference-as-consensus thesis is contested by well-funded efforts, notably Ambient, so Hearth's defensibility is the market and the supply-side distribution wedge (§12), not the verification primitive. Compute, like stake, concentrates; the $B_{\max}$ cap bounds this (a disclosed 2–3× amplification of the most-compute party's weight) without eliminating it, since fully decoupling work from forging would negate PoI, and the coordination-fee cap bounds the analogous Phase-3 pressure. A K-turn composed job is K sequential verified sub-jobs, trading latency for assurance.

16 Conclusion

Hearth turns useful AI inference into block-production weight on a dedicated L1, anchored in stake and verified by a layered guarantee with trust assumptions stated at every step. Staging the rollout (hosted compute made trustless at the node by attestation, then self-hosted inference verified end-to-end, then orchestration held to the network's strongest cheap guarantee) ships value at each step and climbs from a gateway to a settlement layer for verified agent-to-agent AI commerce; the same primitive and economy then generalize past inference into a market for attested computation (§10). The economic bet is the one proof-of-work also makes: that demand to hold the native asset for verifiable useful work exceeds the inflation that funds it, strengthening as the network ascends the assurance ladder.

References

[1] Will Cai, Tianneng Shi, Xuandong Zhao, and Dawn Song. Are you getting what you pay for? auditing model substitution in LLM APIs, 2025.

[2] Anthropic. Commitments on model deprecation and preservation. `https://www.anthropic.com/research/deprecation-commitments`, 2025. Accessed June 2026.

[3] OpenAI. Deprecations. `https://developers.openai.com/api/docs/deprecations`, 2025. Accessed June 2026.

[4] Haochen Sun, Jason Li, and Hongyang Zhang. zkllm: Zero knowledge proofs for large language models, 2024.

[5] A. Begicheva et al. Fair proof of stake. ResearchGate, publication 335147247, 2019.

[6] Intel Corporation. Intel trust domain extensions (intel TDX). Whitepaper, 2023.

[7] NVIDIA. NVIDIA confidential computing (H100 tensor core GPUs). Technical documentation, 2023.

[8] KD Conway, Cathie So, Xiaohang Yu, and Kartin Wong. opml: Optimistic machine learning on blockchain, 2024.

[9] Harry Kalodner, Steven Goldfeder, Xiaoqi Chen, S. Matthew Weinberg, and Edward W. Felten. Arbitrum: Scalable, private smart contracts. In *USENIX Security Symposium*, 2018.

[10] Jack Min Ong, Matthew Di Ferrante, Aaron Pazdera, Ryan Garner, Sami Jaghouar, Manveer Basra, Max Ryabinin, and Johannes Hagemann. Toploc: A locality sensitive hashing scheme for trustless verifiable inference, 2025.

[11] Adam Karvonen, Dario Reuter, Roy Rinberg, et al. Difr: Inference verification despite nondeterminism, 2025.

[12] Irena Gao, Percy Liang, and Carlos Guestrin. Model equality testing: Which model is this API serving?, 2024. ICLR 2025.

[13] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

[14] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning (ICML)*, 2023.

[15] Yuma Rao and Opentensor Foundation. Bittensor: A peer-to-peer intelligence market. White paper, `https://bittensor.com/whitepaper`, 2021.

[16] Ilan Komargodski and Omri Weinstein. Proofs of useful work from arbitrary matrix multiplication, 2025.

[17] Pearl Research Labs. Pearl: A proof-of-useful-work layer-1 via matrix multiplication. White paper, `https://pearlresearch.ai`, 2026.

[18] Elizabeth Lui and Jiahao Sun. Bittensor protocol: Critical and empirical analysis, 2025.

[19] John R. Douceur. The Sybil attack. In *Peer-to-Peer Systems (IPTPS 2002)*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 2002.

[20] Vivek Bagaria, Amir Dembo, Sreeram Kannan, Sewoong Oh, David Tse, Pramod Viswanath, Xuechao Wang, and Ofer Zeitouni. Proof-of-stake longest chain protocols: Security vs predictability, 2020.

[21] Aggelos Kiayias, Philip Lazos, and Paolo Penna. Single-token vs two-token blockchain tokenomics. In *6th Conference on Advances in Financial Technologies (AFT 2025)*, 2025.

[22] Theo Diamandis, Alex Evans, Tarun Chitra, and Guillermo Angeris. Dynamic pricing for non-fungible resources: Designing multidimensional blockchain fee markets, 2022.

[23] Rodney Garratt and Hyun Song Shin. Stablecoins versus tokenised deposits: Implications for the singleness of money. BIS Bulletin No. 73, 2023.

[24] Aggelos Kiayias, Philip Lazos, and Jan Christoph Schlegel. Would Friedman burn your tokens? In *Financial Cryptography and Data Security (FC 2024)*, 2024.

[25] Urban Jermann and Haotian Xiang. Tokenomics: Optimal monetary and fee policies. Working paper, The Wharton School, 2022.

[26] Horace He and Thinking Machines Lab. Defeating nondeterminism in LLM inference. Thinking Machines Lab: Connectionism, 2025.

[27] Yue Zhang, Shouqiao Wang, Sijun Tan, Xiaoyuan Liu, Ciamac C. Moallemi, and Raluca Ada Popa. Proof of sampling: A nash equilibrium-based verification protocol for decentralized systems, 2024.

[28] Jens Groth. On the size of pairing-based non-interactive arguments. In *Advances in Cryptology – EUROCRYPT 2016*, 2016.

[29] Oguzhan Ersoy et al. Verde: A verification system for machine learning over untrusted nodes. Gensyn, `https://blog.gensyn.ai/verde-a-verification-system-for-machine-learning-over-untrusted-nodes`, 2024.

[30] Ambient. Ambient: Verified AI inference via proof of logits on an SVM layer-1. `https://ambient.xyz`, 2025.

[31] David Liberman and Gonka (Product Science Inc.). Decentralized AI: Meaningful utilization of computational power for real-world application. Gonka white paper, `https://gonka.ai/whitepaper.pdf`, 2025.

[32] David Ribeiro Alves, Vishnu Patankar, Matheus Pereira, Jamie Stephens, Nima Vaziri, and Sreeram Kannan. Eigenai: Deterministic inference, verifiable results, 2026.

[33] Ritual Foundation. Ritual chain: A decentralized inference execution layer with deterministic verification. Documentation, `https://docs.ritualfoundation.org`, 2025.

[34] Inference Labs. Proof of inference: The zero-knowledge verified inference network (zk-vin) and sertn. Documentation, `https://docs.inferencelabs.com`, 2025.

[35] Allora Foundation. Allora: A self-improving, decentralized machine intelligence network. White paper, `https://research.assets.allora.network/allora.0x10001.pdf`, 2024.

[36] Aizel Labs. Aizel network: The verifiable intelligence protocol (white paper v1.0). `https://aizelnetworka.com/whitepaper.pdf`, 2025.

[37] Zero Gravity Labs. 0g: Towards a fully decentralized AI operating system. White paper, https://0g.ai, 2025.

[38] Abhinaba Basu. The usefulness gap in proof-of-useful-work: An empirical study of pearl's cupow protocol, 2026.

[39] Venice AI. VVV: The privacy coin for AI. Venice AI, 2026.

[40] OpenRouter. Models: Provider endpoints and per-token pricing. https://openrouter.ai/models, 2026. Same-model endpoints (e.g. Llama 3.3 70B) list per-token prices spanning several-fold across honest providers. Accessed 2026-07-01.

[41] Thunder Compute. NVIDIA H100 pricing: On-demand GPU rate comparison across providers. https://www.thundercompute.com/blog/nvidia-h100-pricing, 2025. Accessed 2026-07-01.

Appendix A Parameter reference

The protocol and economic quantities used across the paper are collected here. Governance-set values are marked as such; the worked-economics block is illustrative, not fixed by the protocol.

Symbol	Meaning	Value	Section
A_n	block time (baseTarget block-delay target)	about 60 s	§4.1
$b_{\min}$	minimum generating balance to forge	governance-set	§4.1
$B_{\max}$	work-boost cap	2 (range 2 to 3)	§4.1
α	workBoost EMA rate (window about $1/\alpha$ epochs)	governance-set	§4.2
κ	workBoost half-saturation	governance-set	§4.2
τ_w	workBoost dust floor for the normalization set $S(E)$	governance-set	§4.2
q	stake-weighted-median quantile for $m(E)$	governance-set, default 1/2	§4.2
$E \to E{+}1$	epoch lag, the verification window	one epoch	§4.3
epoch length	duration of one epoch (the workBoost and verification window unit)	governance-set, illustratively about 1 day ($\approx$1440 blocks)	§4.3
τ	logit-match tolerance	0 under batch-invariant kernels, else governance-set per model class	§6.3
n_{default}	committee size	21 ($n_{\min} = 15$, up to 31 for high-value tiers; 2/3 supermajority)	§6.4
p	sampling rate	1 to 5%	§6.6
penalty	response to proven cheating	generation-suspension (escalating)	§4.4
Illustrative economics (§5.6), not protocol constants:			
HRTH price	network-token price	about \$0.20	§5.6
supply	HRTH supply	about 120M	§5.6
staked	fraction staked	about 20%	§5.6
block reward	emission per block	4 HRTH per 60 s	§5.6

Table 3: Consolidated parameter reference. Governance-set entries are tuned by on-chain governance; the economic block is one illustrative parameterization used in the worked example.

Appendix B Composed-job route receipts

A composed job of turn budget K (§9.1) emits one route receipt recording the pinned router's per-turn decisions and chaining the worker sub-receipts they produced. Each hop is itself a Phase-2 receipt, so the composed receipt is a sequence of independently checkable artifacts, not one opaque summary.

```
route_receipt = {
  job_id,
  route = [ (model_id_1, sub_receipt_1),
            (model_id_2, sub_receipt_2),
            ...
            (model_id_K, sub_receipt_K) ],
  router_model_id, router_commitment,
  H(input), H(final_output),
  workBoost_attribution
}
```

Each `sub_receipt_k` is the Phase-2 receipt for hop k, carrying its attestation, logit commitment, and work-units. The hops chain because the router's turn-k decision is a deterministic function of the transcript prefix fixed by prior hops' committed outputs, so `H(final_output)` is reproducible from `H(input)` and the ordered route. A verifier checks any hop in isolation, replaying the pinned router on the committed prefix to confirm the recorded `(model_id_k, role)` decision (§9.2) and separately re-checking `sub_receipt_k` as an ordinary Phase-2 job. The `router_commitment` is the router's committed per-turn head-logit sketch, checked by the same logit comparison, and `workBoost_attribution` splits retired-Cred value across hops, so credit follows the worker that did the compute, not the router that placed it.

Appendix C Open questions and future work

The caveats in §15 are accepted properties. The items here are gaps between the protocol as described and as implementable, each paired with the measurement, proof, or specification that would close it.

C.1 Specifying and bounding Proof of Inference

The forging multiplier b_{eff} is now fully specified (§4.2), and Lemma 2 bounds a party's forging-weight share at $(1 + B_{\max})$ times its stake share. Two things remain open: calibrating $(\alpha, \kappa, B_{\max}, \tau_w, q)$ against live demand, which no data yet sets, and measuring the bound's tightness by tracking the realized Gini or Herfindahl concentration of b_{eff} to show where the cap must lie to keep amplification inside the disclosed 2–3× band.

C.2 Calibrating the statistical verification layer

Three parameters of the audit pipeline are specified but not measured. The logit-match threshold τ is anchored to the published TOPLOC and DiFR operating point [10, 11]; open is re-measuring it on Hearth's pinned hardware and precision classes and reporting fleet-level false-positive and false-negative rates. The power of Model Equality Testing near τ is settled as a security argument, since the dichotomy of §6.5 flags any divergence large enough to yield a cheaper model [12, 1]; open is only the query count N the near-neighbour case needs, which sets detection latency, not whether the backstop holds. The committee verdict (§6.4) needs an attack-cost model of how much stake corrupts a verdict at a given committee size and Byzantine fraction, including operator or entity seat concentration, since the binomial bound assumes independent, non-colluding seats.

C.3 Economic dynamics and market structure

The reflexive coupling between Cred price, staking yield, HRTH value, and serving decisions (§5.3) is argued to self-correct with lag but only asserted. An economic-security simulation under demand shocks, reporting time-to-recovery and where recovery fails, would replace that assertion with a measured stability region.

The phased Cred structure (§5.1) resolves most of the earlier liquidity worry: the single Phase-1 Cred avoids fragmenting thin markets during bootstrap, and per-model Creds phase in only as each model gains capacity to back them. The residual is the base-to-per-model transition mechanics and which long-tail models stay grouped under the base numeraire Cred rather than graduating, settled once per-model liquidity data exists.

The self-dealing defense (§4.5) leaves a Sybil-payer residual, an operator routing payment through distinct keys it secretly controls, bounded by burn-cost economics rather than a check and folded into the simulation

above.
Genesis is undecided: the initial HRTH distribution and emission schedule are unspecified; one candidate under discussion, recorded as an open design item not a choice, is a "fallen angels" genesis crediting balances tied to deprecated tokens of other networks.

C.4 Node selection, market clearing, and aggregator diversity

How the node for a job is selected is not yet pinned: a VRF lottery and a price/latency/reputation market (§7.2) are both consistent with the text but carry different fairness and self-preferential-serving properties. The empty-market case also needs defined behavior when no operator will serve a model at its Cred price, rather than silent non-service.
Phase 1 attests a single aggregator (OpenRouter). Supporting several concurrently would remove that single point of failure, and nothing in the receipt format binds to one, so the gap is specification, how a job selects among aggregators and how cross-aggregator consistency is judged, not redesign.

C.5 Toward a trustless floor and a pinned specification

The trustless floor remains future work. Frontier-scale zero-knowledge proofs of inference [4] and deterministic-arithmetic bisection are today a fallback; making them the default needs explicit milestones: proving cost per token, added latency, and the model scale at which a ZK or bisection path becomes cheaper than the honest-provider spread it protects, a spread taken as a conservative assumption backstopped by observed provider price dispersion and still to be measured on Hearth's own fleet.