Elephant Pool: A Distributed Inference Marketplace for Open-Weight Models on Verified Heterogeneous Hardware

De Buck Technologies

Belgium — contact@elephantpool.ai

Version 0.2 (public edition) — 17 August 2026

Abstract

Elephant Pool is a distributed inference marketplace: an OpenAI-compatible, token-metered API for open-weight language models, served by a verified pool of heterogeneous hardware — from consumer gaming rigs to datacenter nodes, NVIDIA and AMD alike — coordinated by a lightweight European control plane. Hardware owners run a single open-source agent and receive 80% of the revenue of every token their machines serve; work is verified through activation fingerprints (TOPLOC), sampled replay, and publicly anchored cryptographic receipts. Commercially, the network exploits two documented gaps: EU-sovereign inference providers charge a 2–5× premium over the US price floor on every model both sides list, and their catalogs omit several of 2026's strongest open models entirely (Qwen3.8, MiniMax M3, Gemma 4). By combining community hardware economics with an attested EU-only routing tier, Elephant Pool undercuts every sovereign cloud while paying hosts market-credible rates and carrying the newest catalog first — DeepSeek V4-Flash, GLM-5.2, Qwen3.5-397B and peers. This paper presents sourced market data, the network architecture, the host executable, verification and payment design, and the unit economics of the spread.

Contents

Public edition v0.2 — 2026-08-17 — De Buck Technologies, Belgium

> *Elephant Pool is a distributed inference network: anyone installs an executable (`mahout`), contributes GPU compute — NVIDIA or AMD, from a gaming rig to a datacenter node — and serves open-weight AI models; clients buy token-metered inference through an OpenAI-compatible API. Hardware owners receive **80% of the revenue of every token their machines serve**; all work is cryptographically verified and publicly receipted. Every load-bearing number below is sourced; unverified items are flagged.*

---

1. Positioning: the gap we fill

Three adjacent markets exist today, and none combines them:

Player type	Examples	What they have	What they lack
Inference routers	OpenRouter (Stripe acquisition reported 2026-08-16 at >$7B — Bloomberg/TechCrunch; unconfirmed by the parties), Together, Fireworks, DeepInfra	Token-metered API, model catalog	No community hardware; they rent datacenters
GPU marketplaces	Vast.ai, Salad, io.net, Akash, RunPod community	Community hardware, hourly rental	No token API — clients must be devops-capable
EU-sovereign inference	Scaleway, OVHcloud, IONOS, StackIT	GDPR positioning	Charge **2–5× the US price floor** (see §2) — and list only a fraction of the current open-model catalog

Elephant Pool = the three combined: a token-metered, OpenAI-compatible API for the latest open models, served by a verified swarm of community and operator hardware, with an EU-sovereign tier priced *below* Scaleway/OVH but *above* marginal cost — the spread is the business.

The **anchor fleet** — datacenter-class nodes operated by De Buck Technologies — serves the biggest models and guarantees baseline capacity and SLAs; the community swarm adds elastic capacity at zero capex.

---

2. Market data (sourced, accessed 2026-08-17)

2.1 What inference sells for

The classic benchmark — Llama 3.3 70B, per Mtok (in/out):

Provider	Price in/out	Sovereignty
DeepInfra (US)	$0.10 / $0.32	none — US price floor
Nebius	$0.13 / $0.40	EU data-residency only (US-listed)
StackIT (DE)	€0.45 / €0.65	German DCs
IONOS (DE)	€0.65 / €0.65	German DCs, BSI-certified
OVHcloud (FR)	€0.67 / €0.67	EU sovereign, zero-retention
Scaleway (FR)	€0.90 / €0.90	EU sovereign, Paris
Together (US)	$1.04 / $1.04	none

The same premium holds on the **2026 flagship open models** (per Mtok in/out, cheapest US vs cheapest strict-EU-sovereign):

Model	Cheapest US	Cheapest EU sovereign	EU premium
DeepSeek V4-Flash (304B MoE, 1M ctx, MIT)	$0.08 / $0.18 (DeepInfra)	€0.40 / €0.80 (Scaleway)	~4–5×
GLM-5.2 (753B MoE, MIT)	$0.48 / $1.49 (Novita)	€1.80 / €5.50 (Scaleway)	~3–4×
Qwen3.5 397B-A17B (Apache-2.0)	$0.30 / $1.93 (DigitalOcean)	€0.60 / €3.60 (Scaleway/OVH)	~2×
Qwen3.8 27B (Apache-2.0, 262K ctx)	$0.40 / $3.00 (Chutes)	**no EU sovereign listing**	∞
MiniMax M3 (427B MoE, 1M ctx)	$0.28 / $1.10 (DeepInfra)	**no EU sovereign listing**	∞
Gemma 4 26B-A4B (Apache-2.0)	$0.07 / $0.34 (DeepInfra)	**no EU sovereign listing**	∞

Sources: deepinfra.com/pricing, novita.ai, scaleway.com Generative APIs, OVHcloud AI Endpoints catalog, IONOS AI Model Hub, together.ai/pricing, OpenRouter per-provider endpoints API.

Two conclusions. First, **the EU-sovereignty premium is real: 2–5× on every model both sides list.** Second — just as valuable — **the EU sovereign catalog is stale**: several of 2026's best open models (Qwen3.8, MiniMax M3, Gemma 4) have *no* strict-EU listing at all. A price point of

€0.35/€0.55 for 70B-class and €0.25/€0.55 for DeepSeek-V4-Flash-class in an EU-sovereign pool undercuts every sovereign provider by ~30–50% while staying well above marginal cost on community hardware (§8) — and being *first* in the EU with the newest catalog is a wedge no incumbent contests today.

2.2 What hosts earn today (our competition for supply)

- Salad, RTX 4090: "up to $130–180/month" (top-25% hosts, 24/7) — salad.com; community 30-day tests: ~$115/GPU/mo on RTX 5090 rigs (YouTube, May 2026).
- Vast.ai (official article, 2026-05-18): consumer RTX 5090 earns $0.30–0.60/GPU-hr; 4×5090 rig at 80% utilization → $700–1,400/mo; renter price ≈ host take +~25%.
- Akash posted prices: H100 $1.45/hr, A100 $0.77/hr, RTX 3060 from $0.10/hr.
- Electricity reality check: US residential 18.44¢/kWh (EIA, May 2026); EU household average €0.29/kWh, Belgium €0.35 (Eurostat H2-2025). A 24/7 RTX 4090 at 300–450W average costs $40–60/mo in the US, ~€75–95/mo in the EU → **hourly-rental hosting is marginal-to-negative for EU consumer hosts. Per-useful-token payment (our model) beats per-hour idling.**

Verified platform take rates: TensorDock **25%** commission (official supplier agreement); Vast.ai ~**20%** effective ("prices ~25% above host earnings", own article); Render **5%** network-operator fee; Akash **4%** (AKT) / **20%** (USDC); io.net 0–2% + reservation fees (subsidized by token emissions); Salad undisclosed with a large implied spread.

Market size calibration: RunPod hit **$120M ARR by Jan 2026** (TechCrunch) selling exactly this hybrid datacenter+community model; Salad ≈ $10.4M ARR (2024, est.); io.net ≈ $12M annualized (self-reported); Akash $3.15M network spend in 2025. OpenRouter routed **25T tokens per week** before its reported acquisition (official Series B post, May 2026). The revenue is real and concentrated in platforms that abstract the hardware away — which is what a token-metered API does maximally.

2.3 Throughput facts that make consumer hardware viable

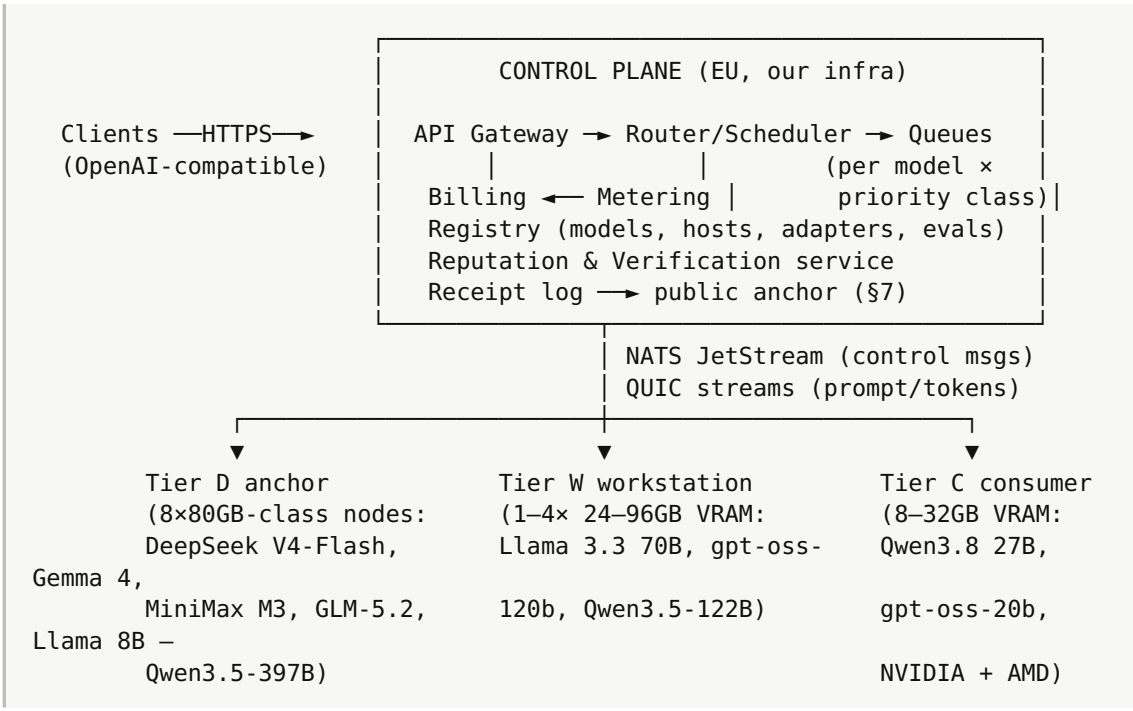
- 2× RTX 4090, Llama 3.3 70B AWQ-INT4, vLLM, batched: **467 tok/s aggregate** (CloudRift benchmark, 2025-10-09) ≈ **1.68M tok/hr**.
- 2× RTX 5090, same setup: **1,230 tok/s** (same source).
- Single RTX 5090: ~4,570 tok/s batched on a 30B-class MoE coder model (CloudRift); single 24GB card serves 32B-dense at ~61 tok/s single-stream (Hardware Corner, 2025-11).

- **The 2026 catalog is kind to consumer cards**: Qwen3.8 27B (dense, 262K ctx) needs ~15 GB at 4-bit; Gemma 4 26B-A4B (MoE, 4B active) ~14 GB; gpt-oss-20b ~13 GB — all fit a single 16–24 GB GPU, NVIDIA or AMD.
- Datacenter comparison: H100 fleet serving 70B at INT4 ≈ $0.77/Mtok on-demand-rental cost (Spheron, 2026-03); B200 spot $0.88/Mtok.

Key consequence: **batching is everything.** A consumer rig doing single-stream chat produces ~20–60 tok/s; the same rig continuously batched produces 15–50× more. The scheduler's job (§3.4) is to keep hosts saturated.

---

3. Network architecture



3.1 Design rule #1: a model never spans hosts over WAN

Petals (arXiv 2209.01188) proved WAN pipeline-sharding of a single model works — and topped out at **1–6 tok/s** on 70B/180B models because per-token WAN round-trips dominate; the project is effectively dormant since ~2024. Tensor parallelism over WAN is outright non-viable (all-reduce every layer). Everything that serves LLMs in production (llm-d, Ray Serve, SGLang router, exo) assumes LAN/datacenter interconnect, and every commercial GPU network (Vast, Salad, io.net, Akash) schedules whole models onto single nodes and competes on verification + payments instead. We follow: **each model instance fits entirely on one host** (multi-GPU tensor-parallel *inside* a host over PCIe/NVLink is fine; llama.cpp MoE expert-offload to system RAM extends reach). Distribution happens at the *request* level, not the *tensor* level.

3.2 Control plane (our EU servers, boring on purpose)

- **API Gateway**: OpenAI-compatible (`/v1/chat/completions`, `/v1/embeddings`, streaming SSE). Auth: API keys; per-key spend limits; EU-pool flag per key or per request (`"data_residency": "eu-sovereign"`).
- **Message bus**: NATS JetStream for host heartbeats, leases, job offers. gRPC for host RPC. Postgres for registry/billing; ClickHouse for telemetry.
- **Data path**: prompts and tokens do NOT transit the bus. The gateway opens a direct QUIC stream to the assigned host (hole-punched; relay fallback on our infra when NAT-blocked). TLS 1.3 end-to-end gateway↔host; prompts never written to disk on either side (zero-retention default, §6).
- **Registry**: model catalog (weights hash, quantization, engine image digest, license), adapter catalog, host inventory (probed hardware, benchmark scores, region, attestation status), eval results per model build.

3.3 Per-model queues & priority classes

Each catalog model has a virtual queue with three classes:

Class	SLA	Price	Mechanics
`interactive`	TTFT < 2s target	1.0×	Routed only to warm hosts (model in VRAM)
`standard`	TTFT < 30s	0.8×	May trigger a model load on a warm-disk host
`batch`	< 24h, async API	0.4×	Fills host idle troughs; mirrors industry −50% batch discounts (Scaleway/Together both do −50%)

Queue-depth per model is the demand signal: the **prewarmer** watches depth/forecast and instructs idle hosts to pull+load models before the queue saturates. Cold-start anatomy is well documented: weight load of a 140GB 70B FP16 ≈ 40–45s at 3–4 GB/s NVMe (a 40GB Q4 proportionally less); engine init adds 10–30s; GPU-memory snapshot techniques cut 7–13B cold starts to 2–5s (Spheron cold-start analysis 2026; FlashServe ACM 2026: 70B in 5.8s via DRAM-tier caching). Weight pulls are P2P-seeded between hosts, BitTorrent-style, to spare our egress.

3.4 Router / scheduler

Score-based dispatch. For each job, candidate hosts (model loaded or loadable, class-compatible, region-compatible) are scored:

```
score = w1·(measured tok/s for this model)      // from continuous telemetry
      + w2·(1 / TTFT_ewma)
      + w3·reputation                            // §5
      + w4·session_affinity                      // same host for same
conversation → KV/prefix-cache reuse
      + w5·(host_price_bid inverse)              // hosts may bid below
default rate
      − w6·(current_load / capacity)
```

Session affinity matters more than it looks: cache-aware routing is worth ~2× throughput on SGLang's router (158k vs 83k tok/s vs round-robin, LMSYS) and ~3× output throughput / 2× lower TTFT on llm-d (Red Hat, 2026); prefix-heavy workloads (RAG, multi-turn) gain up to 6.4× with RadixAttention (SGLang paper). The router pins conversations to hosts with a TTL to capture the same effect across the swarm.

Failure handling: host heartbeat lapse or mid-stream stall → job re-queued at front, partial tokens discarded, client stream resumes from last flushed chunk (gateway buffers); host reputation debited. Every job carries a lease with a deadline.

3.5 Model routing & delegation (the "big model uses small models" part)

Two mechanisms, both request-level:

1. **Auto-router** (`"model": "elephant/auto"`): a RouteLLM-style classifier (cheap, runs on control plane) routes each request to the cheapest model predicted to satisfy it, falling back upward on low confidence. The evidence is strong: RouteLLM retains ~95% of frontier quality while sending only ~26% of calls to the big model (~85% cost cut, LMSYS); FrugalGPT cascades report up to 98% cost reduction at matched quality (Chen/Zaharia/Zou 2023). Sold as a discount tier; the margin improvement is ours.
2. **Subagent delegation**: orchestrator models (e.g. GLM-5.2 or MiniMax M3 on a Tier D node) run with tool-calling enabled; a `delegate` tool lets the big model dispatch self-contained subtasks (summarize this file, extract entities, draft boilerplate) as new jobs on *smaller* model queues (Qwen3.8 27B, Gemma 4 on Tier C), where consumer capacity is abundant and nearly free. The orchestrator's context stays on its host; only subtask prompts/results move. Expensive tokens for reasoning, cheap tokens for grunt work — this is what makes the consumer swarm valuable even when demand concentrates on big models.
3. **Speculative decoding** stays host-local (draft model 1–3B colocated with target model on the same box) — it needs µs-latency to the target; never cross-host.

3.6 Model ↔ hardware tier map (launch catalog, verified on Hugging Face 2026-08-17; VRAM ≈ 4-bit serving)

Model	Params (active)	Ctx	License	~VRAM	Tier
Llama 3.1 8B	8B dense	128K	Llama	~5 GB	C
gpt-oss-20b	21B (3.6B)	131K	Apache-2.0	~13 GB	C
Gemma 4 26B-A4B	26B (4B)	256K	Apache-2.0	~14 GB	C
Qwen3.8 27B	27B dense	262K	Apache-2.0	~15 GB	C
Llama 3.3 70B	70B dense	128K	Llama	~40–43 GB	W (2×24GB / 48GB)
gpt-oss-120b	117B (5.1B)	131K	Apache-2.0	~63 GB (MXFP4)	W (96GB) / D
Qwen3.5 122B-A10B	122B (10B)	262K	Apache-2.0	~61 GB	W / D
DeepSeek V4-Flash	304B (13B)	1M	MIT	~155 GB	D
MiniMax M3	427B (~23B)	1M	community	~215 GB	D
Qwen3.5 397B-A17B	397B (17B)	262K	Apache-2.0	~200 GB	D
GLM-5.2	753B (~40B)	1M	MIT	~380 GB	D (8×80GB)

Network-scale giants — Qwen3.8-2.4T-A95B (~1.2 TB), Kimi K3 (2.78T, ~1.56 TB), DeepSeek V4-Pro (1.65T, ~825 GB) — exceed single-host serving even at 8×80 GB and are outside the catalog's scope by design (§3.1). Tier E (CPU-only/edge) serves embeddings, rerankers and 1–3B models via llama.cpp — low value per node, but it widens the host funnel and feeds the delegation layer.

---

4. The executable: `mahout` (the person who drives an elephant)

Single static binary (Rust), Linux/Windows/macOS, open source (auditable — hosts must trust what runs on their machine). It embeds NO inference code itself; it is a supervisor.

4.1 Components

```
mahout
├── probe        hardware inventory: GPUs (CUDA / ROCm / Vulkan / Metal),
VRAM,
│                RAM, CPU, NVMe free space, bandwidth test, NAT type
├── bench        standardized benchmark: loads a reference model for the
│                hardware class, runs a fixed prompt set, measures tok/s,
│                TTFT, records a TOPLOC activation commitment (§5) —
│                capabilities are MEASURED, never self-reported (io.net's
│                Apr 2024 incident: ~1.8M fake GPUs via spoofed metadata)
├── runtime      engine supervisor, per hardware class:
│                • NVIDIA (CUDA) and AMD (ROCm ≥6.3, RDNA3+) → vLLM in a
│                  pinned OCI container (PagedAttention, continuous
batching,
│                  AWQ/GPTQ/FP8, multi-LoRA) — full support, both, day one
│                • AMD without ROCm / Apple Silicon / Windows / <16GB / CPU
→
│                  llama.cpp (GGUF, Vulkan/Metal/CUDA offload)
│                Engine images + weights are content-addressed (SHA-256) and
│                signature-verified against the registry — the host never
│                executes arbitrary code from us or from clients
├── cache        disk quota set by host; LRU over model weights; P2P chunk
│                seeding to nearby hosts; adapters hot-loaded without
restart
├── net          outbound-only QUIC + NATS (no inbound ports required);
│                WireGuard mesh optional for multi-node host clusters
├── worker       job leases: accept → stream tokens → emit signed receipt
│                {job_id, model_hash, token_counts, TOPLOC commitment, sig}
├── policy       host controls: availability windows, max power draw, price
│                floor bid, VRAM reservation, one-key pause ("game mode"
│                auto-pause on fullscreen app / user activity)
└── update       signed auto-update channel, staged rollout, rollback
```

4.2 Onboarding flow (target: <10 minutes to first earning)

1. Download binary → `mahout up` → browser opens, host links account (email + payout details, identity verification handled by the payment processor).
2. `probe` + `bench` run (~5–10 min incl. reference model pull). Node gets a tier + per-model capability list + provisional reputation.
3. Node goes eligible for `batch` class immediately; `interactive` unlocks after 48h of clean operation (reputation ramp — sybil damping).
4. Random re-benchmarks (~weekly, and on any driver/hardware change) keep the capability list honest.

4.3 Trust in both directions

- **Host protected from us/clients**: open-source agent; engines run in containers with no filesystem access outside cache and no network egress; **no arbitrary client code, ever** — the attack surface is a prompt string

entering a signed engine. This is a real differentiator vs GPU marketplaces that hand renters SSH into your machine.

- **Client protected from host**: (a) zero-retention enforced by agent design (prompts in memory only) — but memory is ultimately host-controlled, so (b) verification (§5) catches wrong-model/wrong-precision fraud, and (c) sensitive workloads go to the **attested tier**: anchor fleet + hosts with TEE-capable hardware (H100 Confidential Computing / Intel TDX), where prompts are decrypted only inside the enclave. The EU-sovereign tier (§6) runs only on attested or contractually-bound EU hosts.

---

5. Verification & reputation (how we pay only for real work)

The verification cost ladder, from the research (sourced in appendix):

Method	Overhead	Catches
TEE attestation	~18–30% latency/throughput (H100 CC + TDX, arXiv 2607.19353); **datacenter cards only — consumer GPUs have no CC mode**	wrong engine/weights, host snooping (hardware-trust caveat)
TOPLOC activation hashing	«1× validate, 258 B per 32 tokens	wrong model, wrong prompt, wrong precision — 100% detection, 0 false ± in evals (arXiv 2501.16007, Prime Intellect; production-used in INTELLECT-2 across 100+ untrusted nodes)
Sampled replication (PoSP)	~<1% (verify a random subset, slash on fraud; honesty is the Nash equilibrium — Hyperbolic, arXiv 2405.00295)	everything, statistically
zkML	>1000×	everything, cryptographically — **not viable** for serving

Our scheme: **every job emits a TOPLOC commitment** in its receipt; the verification service replays a random ~1–2% of jobs on anchor-fleet capacity (plus 100% of a new host's first jobs and any client-disputed job). Mismatch → payout for the epoch withheld, reputation slashed, repeat → ban + forfeiture of pending balance. Sybil economics: reputation ramps (§4.2) make a farm of fake nodes earn ~nothing during the window in which it must survive replay checks. This is the io.net lesson (proof-of-work-delivered over self-reported

telemetry) plus the Templar/Gauntlet lesson (score contributions by measured effect, pay zero for garbage).

---

6. GDPR / EU positioning

- **Zero-retention default** (OVH markets exactly this; we match): prompts/completions never persisted; only counts, timings, hashes in billing records.
- **EU-sovereign tier**: routing constrained to attested/contracted hosts physically in the EU, operated by De Buck Technologies (BE) — no US CLOUD Act exposure (unlike Nebius). DPA + SCCs templates for enterprise. Priced below every sovereign competitor (§2.1) with a gross margin still >50% (§8) — and carrying 2026 models the sovereign clouds don't list at all.
- **No training on customer API data. Period.** Contractual, and the only defensible read of EDPB Opinion 28/2024 for B2B API traffic. Should the network ever train or fine-tune models, it would use public/licensed corpora, synthetic data, and *explicitly opt-in* traffic only — under the production-proven privacy stack (federated updates + differential privacy + secure aggregation, the Gboard template: >20 models shipped under formal DP, arXiv 2305.18465).
- **EU AI Act**: serving open models makes us a deployer/distributor (light duties); GPAI-provider duties (Art. 53 documentation, training-content summaries) attach only to actors that train models, with cumulative-compute thresholds we track by design.

---

7. Payments & the public-receipt layer

Primary rails are fiat. Clients: prepaid API credits via Stripe (cards/SEPA), usable only for Elephant Pool services, unused balances refundable. Hosts: EUR balance, monthly SEPA payout from €50 via Stripe Connect (identity verification included) — hosts are paid **service fees for work performed**, nothing else.

The blockchain is an audit layer, not the money layer. Sourced facts: transaction fees at BSV's 1 sat/kB floor make a receipt-anchoring transaction cost $\approx 0.0000002; mint Bluewrote 50.5 M reale - invoice transactions in 24h for \ €400; BRC-105 standardizes HTTP-402 per-request micropayments; SPV proofs (BEEF/BUMP) make anchored receipts verifiable offline against block headers alone. But also$:

BSVmarketcap 300M, ~0.03% of BTC's SHA-256 hashrate, a 100-block reorg in its history (Aug 2021), shrinking exchange access. Therefore:

1. **Usage receipts**: every signed job receipt goes into a Merkle tree; the root is anchored on-chain every 10 minutes via ARC; BEEF/BUMP proofs stored with the receipt. Any host or client can independently prove "this work happened, this much was owed, at this time." Cost: ~$0.001/month. A genuine trust feature for a marketplace that pays thousands of small parties.
2. **Model provenance**: every catalog build's weights hash and eval report anchored on-chain — an immutable public lineage answering "which weights served my request on date X" (also useful for AI-Act record-keeping).
3. **Never**: customer balances or host earnings in crypto; custody of any kind; treasury exposure beyond operational dust. EUR in, EUR out.

---

8. Unit economics (the spread)

Marginal cost on community hardware, 70B-class, from §2.3 numbers: 2×4090 at 467 tok/s ≈ 1.68M tok/hr; power 920W at Belgian €0.35/kWh → €0.32/hr → **electricity cost ≈ €0.19/Mtok**. Host hardware amortization on already-owned gaming rigs ≈ 0 (that's the whole arbitrage). Single-card 27B-class (Qwen3.8/Gemma 4 on one 4090 or RX 7900 XTX) runs proportionally cheaper still.

Line (70B-class, blended in/out)	€/Mtok
Sell: standard pool	0.40
Sell: EU-sovereign tier	0.35 in / 0.55 out
Host payout (80%)	~0.34
Host's own electricity (EU worst case)	0.19
Host net	**~0.15/Mtok → ~€0.26/hr saturated**
Platform gross (20%)	~0.09/Mtok

Sanity checks: a saturated 2×4090 host nets ~€190–210/mo (50% utilization: ~€100/mo) — right in the Salad/Vast realistic band (§2.2), so the payout is market-credible without being charity. Our 20% take sits exactly in the verified market band: below TensorDock's 25% and Vast's ~20–25% effective markup, far above crypto-subsidized outliers (io.net 2%, Akash 4%) that compensate with token emissions we don't need. OpenRouter's pure-routing fee is far lower (~5% of payment volume, no markup on inference — official

FAQ) — but OpenRouter doesn't operate supply; we capture the hosting margin too, like RunPod (whose gross margins Sacra estimates at 65–78%).

Anchor fleet (Tier D): H100-class serves 70B INT4 at ≈$0.77/Mtok on-demand-rental cost (Spheron 2026-03); owned nodes do better (no rental margin, Belgian industrial power ~€0.10–0.15/kWh) — est. €0.25–0.40/Mtok all-in for DeepSeek-V4-Flash/GLM-5.2-class, sold at €0.25–0.60 in / €0.55–1.80 out — comfortably positive, and it back-stops SLAs when the swarm is thin.

---

9. Long-term direction

A network that *serves* open models on pooled hardware is also, structurally, a network that can *improve* them: the feasibility of collective training on distributed, permissionless hardware is now established at scale (INTELLECT-1: 10B trained on 1T tokens across 3 continents at 83–96% utilization; INTELLECT-2: 32B async RL from permissionless workers, TOPLOC-verified; Templar's Covenant-72B: 72.7B pretrained by 70+ independent nodes; DiLoCo-class methods cut inter-node bandwidth needs to 1–5 Gbit/s bursts — all sourced in the appendix). Elephant Pool's architecture keeps that door open by design — training-shaped work fits the same job, verification and receipt rails as inference. Concrete plans in this direction will be published as they mature.

---

10. Risks (the honest table)

Risk	Reality	Mitigation
EU consumer electricity squeezes host margin	€0.29–0.40/kWh vs US $0.18	pay per token not per hour; batch saturation; recruit low-tariff regions; solar households (BE is full of them)
Price war from US floor (DeepInfra $0.08–0.10 in)	permanent	don't fight the floor; win on sovereign tier + newest-catalog coverage + delegation economics + host community
Fake/gamed hosts	io.net: 1.8M spoofed GPUs	measured benchmarks, TOPLOC receipts, replay sampling, reputation ramp, payout withholding
Big-model demand vs consumer supply mismatch	most revenue is 70B+	anchor fleet carries big models; swarm monetized via delegation + the strong 2026 sub-30B catalog
Restrictive licenses on some flagship models	MiniMax community license; Kimi K3 revenue-tiered	license linked per model in the API; Apache-2.0/MIT models prioritized in catalog
Chain risk on the receipt anchor	documented (mcap, reorg history)	anchors are an audit layer only, zero custody; dual-anchoring (e.g. OpenTimestamps/BTC) possible at any time
Two-sided cold start	classic	anchor fleet = supply floor from day one; guaranteed starter work for new hosts; aggregator listings for demand

---

Appendix — primary sources

Pricing/economics: deepinfra.com/pricing · together.ai/pricing · fireworks.ai + novita.ai + Groq (via OpenRouter endpoints API) · scaleway.com pricing (Generative APIs) · OVHcloud AI Endpoints catalog · IONOS AI Model Hub · european.cloud (StackIT, IONOS EUR) · vast.ai host-earnings article (2026-05-18) · docs.tensordock.com supplier hosting agreement · salad.com host pages · console-api.akash.network/v1/gpu-prices · io.net docs · Render BME (RNP-001/018) · runpod.io/pricing + TechCrunch $120M ARR (2026-01-16) + sacra.com/c/runpod · cloudrift.ai RTX benchmark (2025-10-09) · spheron.network cost-per-token benchmark 2026 · gputracker.dev 2026 · EIA Electricity Monthly Update (2026-07) · Eurostat electricity prices (H2-2025). Models (verified on Hugging Face, 2026-08-17): Qwen/Qwen3.8-27B · Qwen/Qwen3.8-2.4T-A95B · deepseek-ai/DeepSeek-V4-Flash-0731 · deepseek-ai/DeepSeek-V4-Pro-0813 · MiniMaxAI/MiniMax-M3 · moonshotai/Kimi-K3 · zai-org/GLM-5.2 · google/gemma-4-26B-A4B-it · mistralai/Mistral-Large-3-675B · openai/gpt-oss-120b. Distributed inference systems: Petals arXiv:2209.01188 ·

exo · llm-d (Red Hat, 2025) · Ray Serve LLM · SGLang RadixAttention arXiv:2312.07104 · RouteLLM (lm-sys) · FrugalGPT arXiv:2305.05176 · EAGLE-3 arXiv:2503.01840 · cold starts: Spheron 2026 + FlashServe (ACM 2026) · TEE overhead arXiv:2607.19353 · PoSP arXiv:2405.00295 · SVIP arXiv:2410.22307. Verification/incentives: TOPLOC arXiv:2501.16007 · Gensyn Verde arXiv:2502.19405 · PoL is broken arXiv:2208.03567 · Gauntlet arXiv:2505.21684 · io.net incident (theblock.co + ionet postmortem, 2024-04) · Bittensor consensus docs · arXiv:2507.02951. Decentralized training: DiLoCo arXiv:2311.08105 · Streaming DiLoCo arXiv:2501.18512 · DiLoCo scaling laws arXiv:2503.09799 · OpenDiLoCo arXiv:2407.07852 · INTELLECT-1 arXiv:2412.01152 · INTELLECT-2 arXiv:2505.07291 · DisTrO/DeMo arXiv:2411.19870 · Templar Covenant-72B (2026-03) · IOTA arXiv:2507.17766. Privacy/regulatory: EDPB Opinion 28/2024 · Gboard DP-FL arXiv:2305.18465 · VaultGemma arXiv:2510.15001 · Secure Aggregation (ACM CCS 2017) · EU AI Act GPAI guidelines + training-summary template (2025-07). Receipts/anchoring: Teranode release (2025-10) · TAAL 1 sat/kB · mintBlue 50M tx/24h (2023-03) · BRC-62/74/105 (bsv.brc.dev) · bitinfocharts (2026-08-17) · reorg history (coindesk/protos, 2021).