Reasoning Capture: The Airbnb for AI Compute

A Verifiable Consumer-Device Inference Network
with Stablecoin Settlement

Reasoning Capture Research

Working Paper v0.1 — July 19, 2026

Abstract

Inference, rather than model training, is becoming the principal source of growth in artificial-intelligence compute demand, with autonomous software agents contributing an increasing share of application-level workloads. At the same time, a large installed base of consumer laptops, desktops, and workstations remains intermittently idle while being capable of serving small and medium open-weight language models. This paper presents Reasoning Capture, a coordinator-mediated network that allows device owners to supply idle inference capacity and receive dollar-denominated compensation per verified unit of work. Requesters access the network through a chat-completions-compatible interface and settle through metered accounts or, optionally, the x402 machine-payment protocol.

The paper describes the network architecture, its versioned wire protocol, and a pragmatic verification mechanism based on randomized seeded re-execution and endogenous device trust. It also identifies an upgrade path toward activation-commitment methods proposed in the recent verifiable-inference literature. The economic design separates compensation from network incentives: USDC functions as the wage, while a capped, work-anchored token (RCAP) is proposed as an incentive and coordination layer. We evaluate Robinhood Chain as the proposed token and identity layer and Tempo as a potential settlement rail, and compare the design with decentralized datacenter-compute networks and intelligence markets. The central contribution is an integrated design for per-request inference on heterogeneous consumer hardware that makes verification and stable-value compensation first-class concerns. Important limitations remain: two-party re-execution cannot unambiguously assign fault, heterogeneous runtimes complicate deterministic comparison, consumer hardware cannot serve all frontier-scale models, prompts are visible to serving devices, and machine-native payment demand remains immature.

Keywords: decentralized inference; verifiable computation; consumer hardware; stablecoin settlement; machine payments; DePIN; language models

Research working paper. Information cutoff: July 19, 2026.

Contents

1 Introduction

Two structural changes motivate a reconsideration of inference infrastructure. First, industry analyses report that inference is becoming the dominant and fastest-growing source of artificial-intelligence compute demand, with iterative and agentic workloads acting as an important growth driver [19, 27]. Second, inference services are increasingly invoked by software agents and automated workflows rather than directly by human users. This change creates demand not only for model access, but also for programmatic service discovery, authentication, metering, and payment [12, 13].

On the supply side, modern consumer laptops, gaming desktops, and workstations form a large but fragmented pool of intermittently idle compute. Many such devices can serve quantized open-weight models in the approximately 3–8 billion parameter range. Existing decentralized-compute networks have generally emphasized datacenter-class accelerators, cluster-hour rentals, general-purpose cloud capacity, rendering, or token-incentivized model-quality markets [19, 20]. The consumer-device segment remains comparatively underserved.

The principal barrier is verification. A market that compensates anonymous devices for inference must determine whether a job was executed, whether the declared model and precision were used, and whether the returned result is sufficiently correct. Self-reported work is not an adequate accounting primitive: device-side counters, points, or receipts can be fabricated unless they are independently checked. This failure mode informed the design of Reasoning Capture following an earlier prototype in which device-reported points were forgeable.

Reasoning Capture addresses the problem through four design commitments:

1. completed work is metered by the coordinator rather than by the serving device;
2. a randomized subset of eligible jobs is re-executed on an independent device under constrained sampling parameters;
3. device compensation accrues in USDC through regulated settlement infrastructure; and
4. any proposed RCAP emission is linked to epochs of verified work through on-chain Merkle claims rather than discretionary or time-based minting.

The requester-facing interface follows the widely used chat-completions pattern, including streaming responses and model discovery [1]. Account-based metering is the primary demand rail; x402 is included as an optional, account-less payment mechanism rather than as a core revenue assumption [12, 18].

This paper makes three contributions. First, it specifies a deployable systems architecture for scheduling and metering inference across heterogeneous consumer devices. Second, it formalizes the security properties and limitations of randomized re-execution as a first-stage verification mechanism. Third, it proposes a two-asset economic design that separates a stable-value wage from a speculative network incentive. The result is a working-paper design, not a claim of trustless or fully decentralized inference.

2 Background and Related Work

2.1 Decentralized compute and intelligence markets

Decentralized compute projects occupy several distinct market categories. io.net aggregates distributed accelerators into clusters and markets capacity for machine-learning workloads; its unit of sale is primarily the cluster-hour rather than the individual completion [19, 20]. Akash provides a general-purpose decentralized cloud and illustrates the sensitivity of supply to provider economics: one third-party report describes a substantial contraction in available GPU inventory during the first quarter of 2026 as rewards declined [21]. Render originated as a distributed rendering network and has expanded toward broader GPU and inference use cases [19]. Bittensor, by contrast, rewards model or subnet output quality and is better understood as an intelligence market than as a direct hardware-rental market [19].

The proposed network differs along four dimensions: its intended supply consists primarily of consumer devices; demand is priced at completion granularity; the interface is compatible with common inference clients; and output verification is treated as an explicit protocol function. These distinctions are architectural targets rather than evidence, by themselves, of superior cost or performance.

2.2 Verifiable inference

Recent work seeks to verify language-model inference without the high prover cost associated with general-purpose zero-knowledge proofs. TOPLOC commits to selected final-layer activations and uses compact encodings to detect model, precision, or output substitution [22]. VeriLLM proposes selective commitments to critical activation states and reports verification cost near one percent of inference cost in its evaluated setting [23]. TensorCommitments reports lower proving and verification costs than TOPLOC under its experimental assumptions [24]. DiFR addresses the additional challenge that production GPU stacks may yield nondeterministic outputs even when high-level inference parameters are fixed [25].

These systems inform, but are not implemented in, the first version of Reasoning Capture. The initial mechanism uses independent re-execution because it can be deployed across unmodified heterogeneous runtimes. Activation commitments are a natural second layer once a meaningful portion of the fleet uses an instrumented and sufficiently standardized inference engine.

2.3 Machine-native payments

x402 adapts the HTTP `402 Payment Required` status to programmatic, per-request payment and supports stablecoin-denominated access without a conventional account relationship [12]. Reported adoption figures are large, but their interpretation is disputed. Independent and journalistic analyses attribute a material share of observed activity to self-dealing or incentive farming and estimate a much smaller genuine-commerce segment [14, 16, 17]. Security researchers have also catalogued attack surfaces in agentic-payment flows [18].

The institutional trajectory is nevertheless relevant. The x402 Foundation has moved protocol stewardship toward an open-governance structure, and major infrastructure and payments firms

have participated in the surrounding ecosystem [15]. Tempo similarly positions stablecoin settlement and machine-initiated payments as first-class protocol functions [9, 10]. Accordingly, this paper treats machine-native payment as a low-cost distribution option, not as evidence of established product demand.

3 System Model and Design Goals

3.1 Participants and trust assumptions

The network contains three principal roles:

Requester.
A human-operated application or autonomous agent that submits an inference request and pays for completed service.

Coordinator.
A trusted service that authenticates requesters and devices, prices and schedules jobs, relays token streams, records metering events, initiates verification, and executes payouts.

Node agent.
A lightweight process on a contributor device that advertises capabilities, accepts eligible jobs, invokes a fixed inference runtime, and streams results to the coordinator.

Version 1 does not eliminate trust in the coordinator. The coordinator can observe prompts and outputs and is trusted to schedule honestly, meter correctly, and publish accurate reward epochs. The verification layer constrains dishonest devices; it does not yet make coordinator behavior trustless.

3.2 Design objectives

The design prioritizes the following objectives:

G1. **Compatibility:** existing clients should require only endpoint and credential changes.
G2. **Constrained execution:** requesters supply data and sampling parameters, never executable code.
G3. **Server-side accounting:** no device-generated counter is sufficient to create earnings.
G4. **Fault tolerance:** device failure, disconnection, and coordinator restart should not silently lose durable jobs.
G5. **Stable-value compensation:** supplier wages should not depend on token-price volatility.
G6. **Incremental verifiability:** the first implementation should support later adoption of stronger cryptographic checks.

4 Architecture

4.1 Request and settlement flow

Figure 1 summarizes the principal data and value flows. Requesters communicate with the coordinator over HTTPS, while node agents maintain authenticated WebSocket sessions. Payment authorization occurs before or alongside dispatch; final metering occurs only after a terminal server-side job event. Supplier balances are accumulated off-chain and settled in batches.

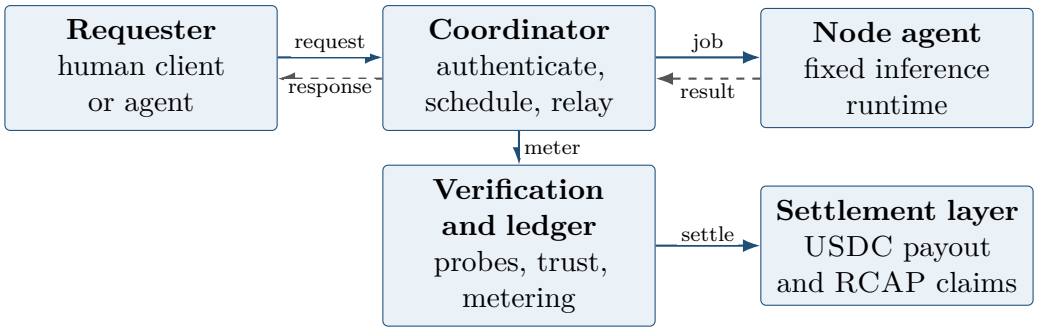

Figure 1: Coordinator-mediated inference, verification, accounting, and settlement. Solid arrows show requests or accounting events; dashed arrows show returned inference data.

4.2 Wire protocol and identity

The device protocol is typed, versioned JSON over WebSocket. Registration includes platform, available memory, accelerator metadata, hosted models, and concurrency limits. An agent whose protocol version is incompatible with the coordinator is rejected with an explicit upgrade instruction. Account API keys and device tokens are issued once and retained by the coordinator only as cryptographic hashes. Operational deployments should additionally use transport security, token rotation, rate limiting, and auditable administrative access.

4.3 Execution boundary

In version 1, a device executes model weights only through a fixed runtime, initially `llama.cpp`. A job contains conversation messages and bounded sampling parameters; it contains no executable program. This reduces the requester-to-device attack surface to malformed or adversarial data supplied to a known parser and runtime. It does not eliminate vulnerabilities in that runtime, model files, drivers, or the node agent itself. Planned defenses include process isolation, resource limits, blocked egress, signed model manifests, and, where available, trusted-execution-environment attestation.

4.4 Scheduling and fault recovery

Jobs are written to a durable queue before dispatch. An eligible device must host the requested model, satisfy its memory floor, expose an unused concurrency slot, and maintain a trust score

above the scheduling threshold. Among eligible devices, the scheduler favors higher-trust and lower-load candidates. A timeout or execution error causes bounded redispatch to a different device. Disconnection requeues in-flight work, and coordinator recovery identifies and requeues orphaned jobs.

The scheduling objective can be expressed as a constrained score. For device i and job j,

$$S(i, j) = w_\tau \tau_i - w_\ell \ell_i - w_q q_i, \tag{1}$$

where $\tau_i \in [0, 1]$ is trust, ℓ_i is normalized load, q_i is an estimated latency or queueing penalty, and $w_\tau, w_\ell, w_q \geq 0$ are policy weights. The scheduler selects the highest-scoring eligible device. This expression describes the policy class; production weights require empirical calibration.

4.5 Demand interface

The requester-facing API follows the chat-completions pattern, including server-sent-event streaming and a model-catalog endpoint [1]. The design goal is compatibility with existing SDKs through a base-URL change. The catalog may include small consumer-servable models and a separate high-memory tier. Moonshot states that full weights for Kimi K3 are scheduled for release by July 27, 2026 [2]. Because that date is subsequent to this paper's July 19 cutoff, K3 support is a prospective integration target rather than a currently demonstrated capability. Its eventual hardware requirements should be evaluated after weights, architecture, license terms, and reproducible benchmarks are available.

5 Verification and Trust

5.1 Seeded re-execution probes

Each device has a trust score $\tau_i \in [0, 1]$, initialized at a neutral value (0.5 in the current design). With probability p (default $p = 0.1$), an eligible completed job is silently re-executed as an unpaid probe on a different device. Probe eligibility requires constrained decoding—including pinned seeds and greedy decoding—and compatible model/runtime configurations.

A normalized match increases the trust of both participants. A mismatch reduces the original executor's trust by a conservative amount and creates a dispute record. If no independent compatible device is available, the probe is marked inconclusive. Timeouts and execution errors also reduce trust. Devices below a policy threshold cease receiving paid work.

If a dishonest device fabricates F independently probed-eligible outputs and each output is sampled with probability p, the probability of at least one probe is

$$P(\text{probe within } F \text{ fabrications}) = 1 - (1 - p)^F. \tag{2}$$

At $p = 0.1$, this quantity exceeds 0.65 after ten fabrications and 0.99 after forty-four. Equation (2) measures selection for checking, not guaranteed detection: correlated devices, collusion, normalization errors, and model nondeterminism can reduce effective detection power.

5.2 Limits of bilateral comparison

Two-party disagreement does not reveal which participant is faulty. Penalizing only the original executor is therefore a heuristic, not an adjudication proof. A stronger dispute procedure should dispatch the same constrained job to $k \geq 3$ independently selected devices and apply a quorum rule, while accounting for correlated software faults and Sybil control. Even a majority cannot prove correctness if a common implementation bug or coordinated adversary dominates the sample.

Bitwise repeatability is also unreliable across heterogeneous GPUs, drivers, quantization schemes, and inference kernels, even at zero temperature [25]. Version 1 therefore normalizes text, restricts probe conditions, and compares only compatible execution classes. These mitigations trade coverage for lower false-positive risk.

5.3 Cryptographic upgrade path

The intended second layer is an activation-commitment mechanism derived from TOPLOC, VeriLLM, TensorCommitments, or subsequent work [22–24]. Such a layer could detect model or precision substitution with less redundant computation and greater tolerance of numerical variation. Adoption requires instrumented runtimes, standardized model manifests, careful benchmarking on consumer hardware, and independent security review. It should therefore complement rather than retroactively justify the properties of the initial implementation.

5.4 Threat analysis

Threat	Version 1 control	Residual risk or planned control
Fabricated output	Randomized re-execution and trust-gated scheduling	Bilateral comparison cannot assign fault; add quorum adjudication
Model substitution	Divergent results may be detected by probes	Direct detection requires model manifests and activation commitments
Device-side earnings forgery	Coordinator creates metering entries after terminal job events	A compromised coordinator can still mis-meter work
Sybil participation	New identities begin at neutral trust and must complete checked work	Add identity friction, stake, diversity-aware probes, and slashing only after robust adjudication
Malicious request payload	Fixed runtime; data-only job schema; bounded parameters	Sandbox runtime, restrict egress, audit parsers, and enforce resource limits
Client disconnect or stream abuse	Server-side lifecycle and metering independent of client connection	Define cancellation, partial-output billing, and refund semantics
Collusion	Independent device selection	Enforce operator and network diversity; use hidden probes and cryptographic checks

Table 1: Threats, first-version controls, and residual risks.

6 Economic Design

6.1 Metering and supplier compensation

Each model has a posted price per thousand completion tokens. Following successful completion, the coordinator records a supplier share—70 percent of gross revenue in the baseline policy—as integer micro-USDC in an append-only ledger. Let r_m be the request price per thousand output tokens for model m, n the accepted output-token count, and $s \in [0, 1]$ the supplier share. The nominal supplier credit is

$$W(m, n) = \left\lfloor 10^6 s r_m \frac{n}{1000} \right\rfloor \quad \text{micro-USDC.} \tag{3}$$

Production accounting must also define input-token charges, failed jobs, partial streams, refunds, taxes, fees, and rounding behavior.

Probe executions are not separately paid. Economically, however, verification is not free: redundant compute consumes provider resources and network margin. A sustainable pricing model must therefore include the expected probe rate and compensate the fleet through the posted supplier share, availability payments, or a separate verification reward. Trust appreciation alone may be insufficient compensation for devices that execute a disproportionate share of probes.

6.2 Stablecoin settlement

Supplier balances accrue in USDC and are intended to settle through Bridge's orchestration APIs, which support fiat and stablecoin transfers and off-ramp workflows [26]. Payouts are batched above a threshold. Ledger entries should be claimed transactionally before submission and associated with idempotency keys so that process failure does not create duplicate transfers.

The governing principle is that compensation for supplied resources should be stable-value and separable from exposure to a network token. This improves price legibility for suppliers but does not remove stablecoin, custody, counterparty, sanctions-screening, tax, or jurisdictional risk.

6.3 Demand-side payment rails

The primary demand model is a metered account authenticated by an API key. x402 is an optional per-request rail for account-less trials and autonomous clients [12]. Given the uncertain share of genuine commerce in present x402 activity [14, 16, 17], no base-case revenue forecast should assume broad machine-payment adoption. Its inclusion is justified only if integration, security, reconciliation, and liquidity costs remain small relative to its option value.

6.4 Work-anchored token proposal

The proposed RCAP token is a hard-capped ERC-20 with maximum supply 10^9. It is not the wage. Its stated purpose is to provide a long-horizon incentive and coordination mechanism linked to verified network contribution.

Per-job accounting remains off-chain. For each epoch e, the coordinator aggregates verified work by operator, constructs a Merkle tree whose leaves encode operator identity, epoch, and claim amount, and publishes root R_e to a distributor contract. An operator submits a leaf and Merkle proof to claim. The distributor enforces the global cap and prevents duplicate epoch claims.

This construction makes published distributions auditable but does not prove that the coordinator's underlying work ledger is correct. Stronger accountability would require signed job receipts, reproducible epoch datasets, challenge periods, multiple attestors, or on-chain commitments to the event stream. Contract audits, multisignature administration, emissions disclosure, and jurisdiction-specific legal analysis are prerequisites to any public token generation or distribution.

7 Chain and Settlement-Layer Selection

7.1 Evaluation criteria

The proposed token and settlement layers are evaluated against five requirements: (R1) EVM compatibility; (R2) low and predictable fees for small claims; (R3) credible operational longevity and neutrality; (R4) access to likely retail node operators; and (R5) alignment with stablecoin and machine-initiated payment.

7.2 Proposed token layer: Robinhood Chain

Robinhood's documentation describes Robinhood Chain as an Arbitrum-based Ethereum Layer 2 that uses ETH for gas and supports standard Solidity deployment workflows; the published network identifiers are 46630 for testnet and 4663 for mainnet [3, 4]. Robinhood announced a public testnet in February 2026, while third-party reporting describes a July 2026 mainnet launch and early ecosystem activity [5–8].

The principal strategic argument is distribution: Robinhood's retail customer base may overlap with the population of technically capable device owners. This is a market-access hypothesis, not established evidence that brokerage users will operate inference nodes or acquire RCAP. The chain's short operating history, evolving decentralization properties, ecosystem concentration, and association with a regulated financial brand create technical and regulatory uncertainty. A testnet-first deployment, capped pilot, upgrade controls, and a documented migration path are therefore appropriate.

7.3 Potential settlement rail: Tempo

Tempo is presented as a payments-oriented blockchain with stablecoin-denominated gas and machine-payment support. Contemporary reports describe a March 18, 2026 mainnet launch, sub-second target finality, and participation by payments and custody firms [9–11]. These features may make it attractive for frequent wage settlement because operators need not maintain a separate volatile gas asset. Settlement should remain modular: RCAP claims may occur on Robinhood Chain while Bridge or another regulated provider routes USDC over any supported network.

7.4 Alternatives

Base has a comparatively mature x402 facilitator ecosystem and is a pragmatic demand-side payment network if facilitator coverage elsewhere is limited [13]. Solana offers high throughput but would require a separate contract and operational toolchain. A sovereign appchain would maximize protocol control while imposing additional validator, liquidity, bridge, and ecosystem burdens. At the present stage, those burdens outweigh the benefits of sovereignty.

Layer	Principal strengths	Principal qualifications
Robinhood Chain	EVM-compatible; low-fee L2 design; potential access to retail device owners	New operating history; decentralization and operator conversion remain unproven
Tempo	Stablecoin-oriented gas and machine-payment focus	Payments-focused ecosystem; token-market depth and long-run neutrality remain uncertain
Base	Mature EVM tooling and comparatively developed x402 ecosystem	Less differentiated for the proposed retail-operator distribution thesis
Solana	High throughput and broad retail and stablecoin adoption	Requires a separate non-EVM implementation and operational toolchain
Sovereign chain	Maximum control over execution, fees, and governance	Highest validator, bridge, liquidity, security, and ecosystem burden

Table 2: Qualitative chain assessment. Ratings are design judgments, not measured performance results.

8 Market Positioning and Timing

The project's timing thesis rests on three developments: Tempo's reported March 18, 2026 launch of machine-oriented stablecoin infrastructure [9]; Robinhood Chain's reported July 2026 mainnet availability and the contemporaneous maturation of x402 infrastructure [7, 15]; and Moonshot's announced plan to release Kimi K3 weights by July 27, 2026 [2]. The third event had not occurred as of this paper's July 19, 2026 cutoff and should be treated as a scheduled milestone.

These developments potentially reduce risk in different layers—settlement, distribution, payment, and open-model supply—but they do not demonstrate demand for the proposed network. The defensible near-term market thesis is narrower: a chat-completions-compatible service can test whether low-cost, geographically distributed consumer inference has buyers, while account-based billing supplies the primary commercial mechanism. Machine-native payments and a retail-oriented chain add distribution options if and when those markets mature.

The relevant comparison is therefore not solely price per accelerator-hour. Evaluation should measure end-to-end completion price, time to first token, throughput, tail latency, model availability, output-verification cost, failure rate, prompt confidentiality, geographic constraints, and supplier retention. These measurements are necessary before claims of advantage over centralized APIs or datacenter-oriented decentralized networks can be sustained.

9 Implementation and Evaluation Plan

The current design is described as a working version 1, but a research-quality evaluation should report reproducible measurements rather than repository status alone. At minimum, an evaluation should include:

1. **Performance:** time to first token, decode throughput, end-to-end latency, and tail behavior by hardware class, model, quantization, and network condition.
2. **Reliability:** completion rate, redispatch frequency, disconnect recovery, coordinator-restart recovery, and duplicate-metering tests.
3. **Verification:** false-match and false-mismatch rates across hardware and runtime classes, detection under deliberate fabrication and model substitution, and cost as a function of probe probability p.
4. **Economics:** energy and opportunity cost to suppliers, gross margin after probes and payment fees, payout latency, and minimum viable batch threshold.
5. **Security:** protocol fuzzing, credential lifecycle tests, model-manifest integrity, sandbox escape review, stream-abuse testing, and coordinator compromise analysis.
6. **Market behavior:** requester retention, supplier retention, model utilization, geographic concentration, and the fraction of demand settled by accounts versus x402.

A staged deployment should begin with a small model set and a controlled hardware matrix, publish benchmark methodology, and expand only after probe comparability and unit economics are measured.

10 Limitations and Future Work

Verification. Bilateral probes deter some fabrication but cannot adjudicate disagreements. Quorum re-execution and activation commitments remain future work. Probe efficacy also depends on independence among devices and compatibility of runtimes.

Hardware scope. Consumer systems are well suited to some quantized 1–8B-parameter models but not to all frontier-scale models. Large mixture-of-experts or long-context systems may require a high-memory multi-GPU tier. Closed consoles are excluded because they generally do not permit the required third-party background execution environment.

Demand uncertainty. Current machine-payment volume may be partially synthetic, and machine-native commerce remains immature [14, 16]. Commercial forecasts should rely on measured account-based demand rather than headline protocol activity.

Centralization. The coordinator is trusted for scheduling, metering, dispute policy, and reward-root publication. Merkle distribution improves claim auditability but does not decentralize the underlying accounting process.

Privacy. Prompts and outputs are visible to the coordinator and serving device in version 1. Sensitive workloads require client-side redaction, confidential-computing techniques, or encrypted inference methods not yet included in the system.

Regulation and compliance. Stablecoin payouts may require identity verification, sanctions screening, tax reporting, money-transmission analysis, and jurisdictional restrictions. Any public distribution of RCAP requires securities, commodities, consumer-protection, and tax analysis specific to the relevant jurisdictions.

Energy and externalities. Idle hardware is not equivalent to zero-cost or zero-carbon hardware. Evaluation must measure incremental electricity use, thermal wear, cooling demand, and regional energy intensity.

11 Conclusion

Reasoning Capture combines a compatible inference interface, durable coordinator-based scheduling, server-side accounting, randomized re-execution, stablecoin-denominated wages, and a proposed work-anchored token distribution mechanism. Its primary contribution is the integration of these components around a disciplined accounting rule: stable-value compensation should be created only from coordinator-observed work that is subject to independent verification.

The design remains intentionally incremental. The first version is neither trustless nor fully private; its verification mechanism is probabilistic, and its chain and machine-payment theses are not yet validated by sustained demand. The appropriate next step is therefore empirical: benchmark heterogeneous execution, quantify probe reliability and cost, validate supplier unit economics, and report real requester behavior. If those measurements are favorable, activation commitments, quorum adjudication, stronger coordinator accountability, and confidential execution can progressively reduce the remaining trust assumptions.

Disclaimer

This working paper describes software under development and an unissued token design. It is not an offer to sell securities, a solicitation, or investment advice. Figures cited from third-party sources are reproduced or summarized as reported and have not been independently audited. The information cutoff is July 19, 2026; events scheduled after that date are identified as prospective.

References

[1] Moonshot AI, "Kimi Platform Documentation—Overview," 2026. [Online]. Available: https://platform.kimi.ai/docs/overview

[2] Moonshot AI, "Kimi K3 Quickstart," 2026. [Online]. Available: https://platform.kimi.ai/docs/guide/kimi-k3-quickstart

[3] Robinhood, "Robinhood Chain Documentation," 2026. [Online]. Available: https://docs.robinhood.com/chain/

[4] Robinhood, "Deploy Smart Contracts on Robinhood Chain," 2026. [Online]. Available: https://docs.robinhood.com/chain/deploy-smart-contracts

[5] Robinhood Newsroom, "Robinhood Chain Launches Public Testnet," Feb. 2026. [Online]. Available: https://robinhood.com/us/en/newsroom/robinhood-chain-launches-public-testnet/

[6] Yahoo Finance, "Robinhood Arbitrum L2 Chain Launches With a Bang: Hits 4 Million Testnet Transactions," 2026. [Online]. Available: https://finance.yahoo.com/news/robinhood-arbitrum-l2-chain-launches-124715466.html

[7] TechTimes, "Robinhood Chain Goes Live With Tokenized Stocks and a Key Ownership Caveat," July 2, 2026. [Online]. Available: https://www.techtimes.com/articles/319564/20260702/robinhood-chain-goes-live-tokenized-stocks-key-ownership-caveat.htm

[8] DEXTools News, "Robinhood Chain: An Ethereum L2 for 24/7 Tokenized Stocks," July 2026. [Online]. Available: https://www.dextools.io/news/robinhood-chain-ethereum-l2-tokenized-stocks-launch-july-2026

[9] CoinDesk, "Stripe-led payments blockchain Tempo goes live with protocol for AI agents," Mar. 18, 2026. [Online]. Available: https://www.coindesk.com/tech/2026/03/18/stripe-led-payments-blockchain-tempo-goes-live-with-protocol-for-ai-agents

[10] Ledger Insights, "Stripe, Paradigm launch Tempo blockchain alongside machine payments standard," 2026. [Online]. Available: https://www.ledgerinsights.com/stripe-paradigm-launch-tempo-blockchain-alongside-machine-payments-standard/

[11] Fortune, "Stripe and Paradigm-backed blockchain Tempo launches advisory unit," Apr. 21, 2026. [Online]. Available: https://fortune.com/2026/04/21/stripe-and-paradigm-tempo-advisory-stablecoin-adoption/

[12] x402, "An Open Protocol for Internet-Native Payments," 2026. [Online]. Available: https://x402.org/

[13] Chainalysis, "Inside x402: 100M Agentic Payments on Base," 2026. [Online]. Available: https://www.chainalysis.com/blog/x402-agentic-payments-adoption/

[14] CoinDesk, "Coinbase-backed AI payments protocol wants to fix micropayments, but demand is just not there yet," Mar. 11, 2026. [Online]. Available: https://www.coindesk.com/markets/2026/03/11/coinbase-backed-ai-payments-protocol-wants-to-fix-micropayment-but-demand-is-just-not-there-yet

[15] InfoQ, "Cloudflare and AWS Embed x402 Agent Payments at the Edge," July 2026. [Online]. Available: https://www.infoq.com/news/2026/07/cloudflare-aws-x402-micropayment/

[16] StartupHub.ai, "x402 Payments: The Real Numbers," 2026. [Online]. Available: https://www.startuphub.ai/ai-news/startup-news/2026/x402-payments-the-real-numbers

[17] Cryptopolitan, "Why AI agents aren't exchanging tokens with x402," 2026. [Online]. Available: https://www.cryptopolitan.com/why-ai-agents-arent-exchanging-tokens-x402/

[18] "Five Attacks on x402 Agentic Payment Protocol," arXiv:2605.11781, 2026. [Online]. Available: https://arxiv.org/abs/2605.11781

[19] Yellow Research, "AI Compute Demand Is Outpacing Supply, And Crypto GPU Networks See the Gap," 2026. [Online]. Available: https://yellow.com/research/ai-compute-demand-crypto-gpu-networks-gap-2026

[20] BuildMVPFast, "DePIN GPUs for Inference: io.net, Akash vs. AWS," 2026. [Online]. Available: https://www.buildmvpfast.com/blog/depin-gpu-inference-io-net-akash-below-aws-2026

[21] Go2Mars Labs, "2026: Real Progress and Investment Opportunities in Decentralized Compute Networks (DePIN)," Medium, May 2026. [Online]. Available: https://medium.com/@Go2Mars/2026-real-progress-and-investment-opportunities-in-decentralized-compute-networks-depin-5791ceed802a

[22] J. Ong et al., "TOPLOC: A Locality Sensitive Hashing Scheme for Trustless Verifiable Inference," arXiv:2501.16007, 2025. [Online]. Available: https://arxiv.org/abs/2501.16007

[23] "VeriLLM: A Lightweight Framework for Publicly Verifiable Decentralized Inference," arXiv:2509.24257, 2025. [Online]. Available: https://arxiv.org/abs/2509.24257

[24] "TensorCommitments: A Lightweight Verifiable Inference for Language Models," arXiv:2602.12630, 2026. [Online]. Available: https://arxiv.org/abs/2602.12630

[25] "DiFR: Inference Verification Despite Nondeterminism," arXiv:2511.20621, 2025. [Online]. Available: https://arxiv.org/abs/2511.20621

[26] Bridge, "API Documentation," 2026. [Online]. Available: https://apidocs.bridge.xyz/

[27] KuCoin Research, "AI Compute + Crypto: The Next $10B Narrative?" 2026. [Online]. Available: https://www.kucoin.com/blog/ai-compute-crypto-depin-narrative