APIX: SUSTAINING AN OPEN LLM API MARKET WITHOUT CENTRALIZED PROVIDER ADMISSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Autonomous agents increasingly rely on LLM API services. Yet existing API platforms adopt centralized admission, limiting two-sided openness by restricting provider participation and curating user choice. A decentralized market is a potential solution, but opaque API services introduce a lemons problem: a plausible response alone does not reveal whether the provider actually delivered the advertised service. The key challenge is therefore how to generate credible evidence of provider behavior and sustain market viability in such an open setting. In response, we present **APIX**, a decentralized, permissionless market for LLM API services. APIX uses auction-based matching to coordinate heterogeneous user demands with provider-defined services, a multidimensional reputation system to support informed user selection, and peer-relative consistency evaluation to assess whether delivered services are consistent with providers' advertised capabilities. We evaluate APIX through an LLM-agent-based market simulation that captures the co-evolution of participant strategies and market conditions: APIX reduces opportunistic-profit share from 26.4% to 0.5% and improves service satisfaction from 90.0% to 99.4%. Our blockchain-based prototype further incurs less than 300 ms of additional latency per call over direct API access, corresponding to only 3% overhead for the 8B model. Overall, these results show that APIX can sustain a reliable and two-sided-open LLM API market without centralized admission.

1 INTRODUCTION

Autonomous agents (Park et al., 2023; Tran et al., 2025; Yehudai et al., 2026) are driving growing demand for diverse LLM API services. However, existing API platforms remain largely centralized: first-party providers (OpenAI, 2026; Anthropic, 2026) serve only their own models, while aggregators (OpenRouter, 2026; Fireworks AI, 2026) decide which third-party providers may participate. This limits openness on both sides, excluding potentially valuable services while narrowing user choice. Decentralization is a potential solution, but LLM API execution is inherently opaque: a fluent response may come from the advertised model or a cheaper substitute. This opacity prevents mechanisms from existing decentralized hardware markets (Akash Network, 2026; Render Network, 2026), where properties are directly verifiable, from readily transferring to API markets, calling for new mechanisms that provide credible evidence of provider behavior and sustain market viability.

However, existing systems only partially support realistic LLM API markets. **Volunteer-driven serving systems** (Borzunov et al., 2023; Jiang et al., 2024; Tong et al., 2025; Fang et al., 2026) rely on altruistic participation, weakening incentives for sustained service. **Token-incentivized AI networks** (Liu et al., 2025; DecentralGPT, 2026; Rao, 2026) organize participation around protocol-defined services, limiting provider autonomy and user-specific selection. **Decentralized API infrastructure** (Pocket Network, 2026; Lava Network, 2026) enables open participation and protocol-level verification, but offers limited evidence about whether providers actually deliver the capabilities they advertise, leaving a critical gap in user-provider trust and practical market operation.

Ideally, a reliable LLM API market that preserves two-sided openness should satisfy three properties: (1) **open and autonomous service matching**, allowing providers to freely enter with self-defined service terms and prices while supporting decentralized user-provider market coordination; (2) **informed service selection**, enabling users to express fine-grained requirements and select suitable services according to their own preferences; and (3) **credible behavioral evidence**, providing

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

observable signals of provider conduct to deter opportunistic behavior and form user-provider trust. However, achieving these properties raises three fundamental challenges.

Challenge 1: Decentralization eliminates the natural coordinator for user-provider matching. Centralized platforms directly organize provider offerings and match them with user demand; in a decentralized market, this coordinating role is absent, leaving no default mechanism for such matching. We address this with an on-chain auction design: users publish service requests, providers submit bids with their offered terms, and users select among them according to their own preferences. This enables decentralized coordination between user demand and provider-defined supply, while leaving service definition to providers and final selection to users.

Challenge 2: Service diversity complicates informed provider comparison. Open bidding exposes users to diverse providers whose performance varies across quality, efficiency, reliability, and other dimensions, while different users value these dimensions differently. We therefore introduce a multidimensional reputation system that summarizes each provider's observed performance across distinct dimensions. During bidding, users consider these reputation profiles alongside providers' offered terms, enabling personalized and evidence-informed service selection.

Challenge 3: Opaque execution makes advertised capabilities difficult to trust. Providers may profit by overstating their capabilities, while users cannot directly verify the model behind an API response. This information asymmetry creates a lemons problem that erodes market viability. To address this, we introduce sampled peer-relative evaluations that compare a provider's behavior with peers advertising similar capabilities, using relative consistency as an observable signal of whether the provider honestly delivers what it advertises. This evidence feeds into the reputation system, influencing future selection and revenue and thereby discouraging provider misrepresentation.

Yet addressing these challenges alone does not guarantee market viability under repeated interaction: as participants respond to past outcomes, their strategies reshape subsequent market conditions, creating feedback loops that static benchmarks cannot capture. Building on recent work in LLM-agent-based economic modeling (Li et al., 2024; Fish et al., 2026), we develop a simulation framework to capture the co-evolution of participant strategies and market outcomes. The simulation reveals emergent behaviors, including strategic under- and over-delivery in response to economic incentives (Figure 1), and shows that our proposed matching, selection, and evaluation mechanisms can effectively curb opportunistic behavior and sustain market viability.

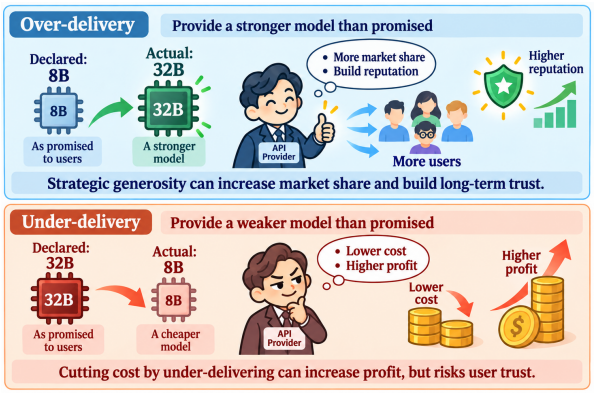


Figure 1: Emergent provider strategies during simulation: LLM agents spontaneously develop under- and over-delivery strategies to pursue their own incentives.

In response, we propose **APIX**, a decentralized, permissionless market for LLM API services:

- In Section 3.2, we introduce auction-based service matching that coordinates heterogeneous user demands with provider-defined offerings, enabling open participation and flexible price discovery.
- In Section 3.3, we present a multidimensional reputation system that summarizes observed provider performance, enabling informed selection under heterogeneous user preferences.
- In Section 3.4, we design a sampled peer-relative evaluation mechanism that generates consistency evidence incorporated into reputation updates, linking provider behavior to future selection and revenue and thereby deterring misrepresentation and strengthening service assurance.

In Section 4, we study APIX under adaptive strategies and emergent market dynamics using our LLM-agent-based simulation framework: APIX reduces the share of provider profit earned through overstatement from 26.4% to 0.5%, while improving service satisfaction from 90.0% to 99.4% and active provider participation from 60% to 95%. In Section 5, our wasmd-based (CosmWasm, 2026) market prototype further incurs less than 300 ms of additional latency per call over direct API invocation, corresponding to only 3% overhead for the 8B model. Together, these results show that APIX can sustain a practical and reliable LLM API market while preserving two-sided openness.

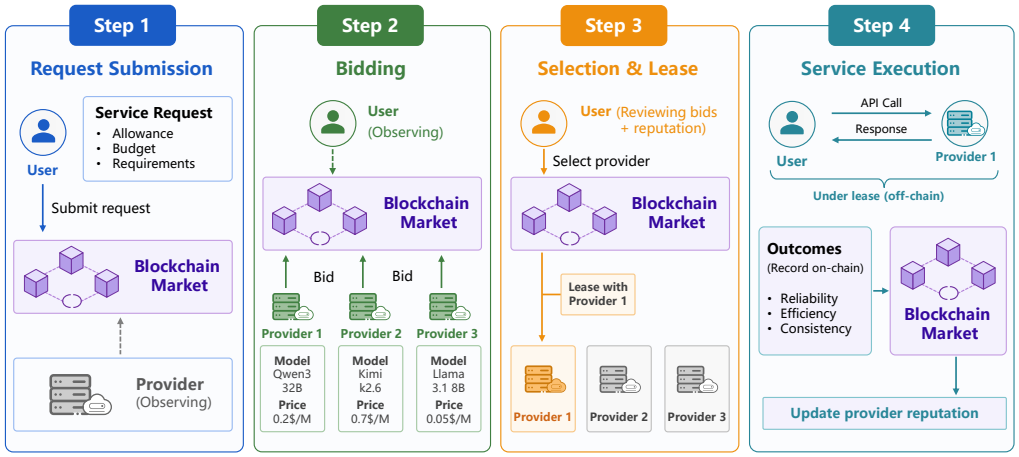


Figure 2: Workflow of APIX. A user posts a service request; providers submit bids under their advertised terms; the user selects a provider based on offered terms and multidimensional reputation to form a lease; and calls are served off-chain, with sampled evaluations updating on-chain reputation.

2 Related Work

LLM API platforms. Open-weight models (Yang et al., 2025; Gemma Team et al., 2025; DeepSeek-AI et al., 2026) and serving systems (Kwon et al., 2023; Zheng et al., 2024) have made inference widely deployable, yet most users consume it through APIs. First-party providers (OpenAI, 2026; Anthropic, 2026; Google, 2026) expose their proprietary models, while hosting platforms and aggregators (Fireworks AI, 2026; OpenRouter, 2026) provide access to multiple models and providers but retain centralized permission over which services are listed. While this provides an initial trust anchor, black-box audits have documented model substitution and silent degradation (Alhazbi et al., 2025; Cai et al., 2025; Zhang et al., 2026; Lin et al., 2026), motivating new mechanisms for user-provider trust in black-box API execution under permissionless participation.

Decentralized computing. Hardware-centric marketplaces (Akash Network, 2026; Render Network, 2026) support open trading of computational resources (e.g., GPUs), but rely on service attributes that are readily verifiable. Volunteer-based systems (Borzunov et al., 2023; Jiang et al., 2024; Tong et al., 2025; Fang et al., 2026) coordinate distributed nodes for collaborative inference, but lack economic incentives for sustained provision. Token-incentivized AI networks (Liu et al., 2025; DecentralGPT, 2026; Rao, 2026) reward participation, but organize providers around protocol- or task-defined objectives, limiting flexible provider offerings and user-specific selection. Decentralized API infrastructure (Pocket Network, 2026; Lava Network, 2026) supports permissionless providers through staking and protocol-level verification, focusing on request delivery, availability, and compliance rather than whether black-box APIs consistently deliver their advertised capabilities.

3 System Design

APIX is a permissionless market for LLM API services that combines open service matching, informed service selection, and evidence-based service assessment. We first describe the end-to-end market workflow in Section 3.1, and then introduce its three core mechanisms: auction-based matching, multidimensional reputation, and peer-relative evaluation, in Sections 3.2–3.4, respectively.

3.1 Market Workflow

Figure 2 illustrates the four-stage workflow of APIX: request submission, provider bidding, user selection and lease formation, and off-chain execution with on-chain feedback.

Request Submission. A user publishes a service request to the on-chain market, specifying requirements such as token allowance, budget, and desired service characteristics. The request becomes visible to participating providers through the shared market state.

Provider Bidding. Providers may respond with bids specifying their offered capabilities and prices. Each provider participates under its own advertised service configuration and bidding strategy, allowing heterogeneous offerings to compete for the same user request.

User Selection and Lease Formation. After bids are collected, the user receives each provider's offered terms together with its multidimensional reputation. The user selects a provider according to its own requirements and preferences and locks the maximum lease value in on-chain escrow. The selected bid is then finalized as a lease, establishing a temporary and auditable service agreement between the user and provider.

Off-chain Execution and On-chain Feedback. API calls under the lease are executed off-chain, with responses returned directly to the user. APIX samples a subset of calls for peer-relative evaluation, comparing delivered behavior against services advertising similar capabilities to derive consistency evidence. This evidence, together with observable execution metrics, updates the provider's on-chain reputation. These updates inform future user selection and connect service behavior to subsequent provider revenue for market self-regulation.

3.2 Auction-Based Matching

To support decentralized matching between heterogeneous user requests and provider-defined services, APIX adopts an on-chain auction mechanism that enables autonomous participation and flexible market interaction. The blockchain serves as a coordination layer, recording requests, bids, and selection outcomes to maintain a shared and auditable market state.

Auction for Decentralized Coordination. A user publishes a request on-chain with task requirements such as allowance, budget, and desired service characteristics. Providers autonomously submit bids specifying their offered service terms and prices. Rather than imposing a centralized allocation rule, the auction exposes a set of competing offers in response to the request, allowing providers to flexibly price and position their services. The user then combines these bids with provider reputation to make an informed service selection, as described in Section 3.3.

Lease Abstraction for Amortized Overhead. Requiring on-chain coordination for every API call would incur substantial latency and transaction cost. APIX therefore introduces a lease abstraction that records the selected provider and agreed service terms as an auditable on-chain lease. Multiple API calls can then be executed under the same lease without repeated on-chain coordination, amortizing auction overhead across calls.

Together, the auction and lease abstractions enable open and autonomous service matching while keeping coordination overhead practical for repeated API interactions, as evaluated in Section 5.

3.3 Multidimensional Reputation

To support informed service selection, APIX summarizes evidence from prior interactions into a multidimensional reputation profile for each provider. Rather than collapsing service performance into a single score, the profile preserves distinct dimensions of observed behavior, allowing users to weigh them according to their own requirements and preferences.

Specifically, APIX tracks three dimensions: (i) *reliability*, reflecting whether a provider successfully fulfills user requests; (ii) *efficiency*, capturing execution performance such as latency and throughput; and (iii) *consistency*, capturing whether the provider's delivered behavior is consistent with what is expected from services advertising similar capabilities. Request outcomes provide reliability evidence, observable execution metrics provide efficiency evidence, and sampled peer-relative evaluations provide consistency evidence, as described in Section 3.4. For each dimension, APIX aggregates successive observations using an exponential moving average (EMA) and commits the resulting reputation profile on-chain for auditable access.

During service selection, users receive each provider's reputation profile together with its advertised terms and price. APIX does not impose a global provider ranking; instead, each user applies its own selection policy to these signals according to its preferences. Reputation thereby links observed provider behavior to future selection and revenue, creating an economic feedback loop that supports both informed user choice and market self-regulation.

3.4 PEER-RELATIVE EVALUATION

APIX does not attempt to verify a provider's hidden execution directly. Instead, it evaluates whether the observed service is consistent with what is expected from other services advertising similar capabilities. When enough peers deliver the quality expected from their advertised terms, repeated comparisons can reveal providers that consistently underperform, providing evidence of potential under-delivery. This design allows lower-capability services to compete on faithful delivery within their own capability class, rather than being penalized by a universal quality standard.

For a sampled subset of API calls, APIX asynchronously queries a peer advertising comparable capabilities (e.g., model family and size) with the same input. Multiple judges with similar or higher advertised capabilities are then selected based on reputation to compare the two responses. Their pairwise assessments are aggregated into a relative quality observation: the provider whose response is judged inferior receives a negative update on consistency reputation, while the other receives a positive update. These updates accumulate peer-relative quality evidence rather than providing direct proof of misrepresentation.

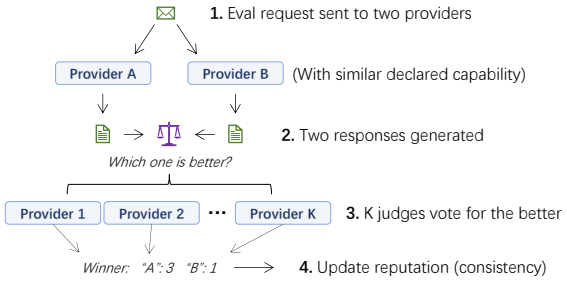


Figure 3: Peer-relative consistency evaluation. Responses from providers advertising comparable capabilities are compared by reputation-weighted judges to generate relative consistency evidence.

Judges receive user-funded rewards for participating in evaluation, while agreement with the majority improves their reputation. Reputation-weighted judge selection reduces the influence of unreliable evaluators, and sampling limits evaluation overhead. The additional inference and judging costs are borne by the requesting user and amortized across calls, as quantified in Section 5.2. By incorporating peer-relative evidence into reputation, APIX turns otherwise opaque service behavior into a user-visible signal that affects future selection and provider revenue, creating economic pressure against persistent under-delivery without centralized verification.

4 AGENT-BASED MARKET SIMULATION

To evaluate whether APIX remains effective as participant strategies and market outcomes coevolve, we build an LLM-agent-based simulator that captures repeated interactions and strategy adaptation among heterogeneous participants. We first describe the simulation framework in Section 4.1, and then analyze the resulting market dynamics and emergent behaviors in Section 4.2.

4.1 SIMULATION FRAMEWORK

The simulator separates market execution from strategic decision-making: a rule-governed environment executes user-provider matching, service delivery, settlement, and reputation updates, while LLM controllers periodically adapt participant policies. Market execution does not require real-time model inference; service outcomes are generated from preconfigured empirical profiles.

Participants and strategy space. We instantiate 16 users, 20 providers, and 7 workload templates with heterogeneous preferences, resources, and task requirements. Table 1 separates fixed participant characteristics and workload specifications from policies that may adapt over time. This preserves persistent differences in user preferences and provider resources while allowing economically relevant strategies to evolve in response to market feedback.

Market execution. In each episode, users generate requests according to workload frequencies and their interests. Providers decide whether to bid based on their advertised capabilities, expected costs, and current policies. Users filter bids using their minimum consistency threshold and rank the remaining offers according to price and their preferences over reputation dimensions.

A key strategic degree of freedom is the distinction between a provider's advertised and served models. The advertised model determines request eligibility, whereas the served model determines

Table 1: Fixed attributes and adaptive policies in the simulated market.

Component	Fixed attributes	Adaptive policies
Users	Reputation-dimension weights; workload interests	Price sensitivity; budget scaling; minimum consistency threshold; satisfaction-rate threshold
Providers	Hardware; operating-cost profile; available models; risk tolerance	Advertised and served models; pricing; minimum profit margin; consistency threshold for bidding
Workloads	Frequency; required model tier; token allowance; price ceiling; throughput target;	None

serving cost, throughput, and the capability actually delivered to the user. Allowing the two to differ makes under- and over-delivery endogenous provider strategies rather than predefined behaviors.

After matching, the environment simulates service delivery and provider profit. Invocation outcomes update reliability reputation, while model-hardware performance updates efficiency. Service satisfaction is modeled separately: a request is considered satisfied if the actually served model meets the workload's minimum capability tier. This experiment-level metric captures whether the delivered service meets user requirements, rather than the semantic correctness of individual responses, allowing us to examine the consequences of provider under-delivery. A user may exit the market if its cumulative service satisfaction falls below its adaptive threshold. Thus, reputation shapes service selection, while realized service satisfaction drives user retention.

Peer-relative evaluation. For sampled comparisons, the simulator draws evaluation outcomes from a pre-profiled probability table constructed from real model responses and judgments, detailed in Appendix B. Indexed by the two served models and the judge model, each entry records the probability that one response is preferred over the other. Sampling from these probabilities approximates live peer-relative evaluation without invoking models during every simulated interaction.

Policy adaptation. At configurable intervals, LLM controllers review each participant's interaction history and observed market feedback and propose revisions to the adaptive policies in Table 1. Revisions must satisfy predefined parameter bounds and fixed participant constraints before taking effect in subsequent interactions. Repeated execution and revision therefore close the feedback loop between market outcomes and strategic behavior.

4.2 Simulation Results

We evaluate APIX under two strategic market settings: first a single-sided market with LLM-agent-controlled providers and rule-based users, and then a two-sided market in which both users and providers are LLM-agent-controlled. Specifically, we examine persistent overstatement, provider participation, and provider-profit concentration as indicators of market viability.

Case 1: Single-Sided Adaptation. We first consider a market with rule-based users and strategic providers. Sixteen users continuously generate service requests, while 20 providers periodically revise their strategies using LLM agents backed by Kimi K2.5 (Kimi Team et al., 2026).

Table 2: Single-sided market outcomes.

Metric	No Eval	APIX
Ever overstated providers	9/20	9/20
Final overstating providers	5/20	2/20
Overstatement profit share	26.4%	0.5%

Without consistency evaluation, 9 of the 20 providers attempted opportunistic under-delivery by advertising higher-capability models than they actually served, and 5 continued this behavior through the end of the simulation. Profit earned while overstating accounted for 26.4% of total provider profit, indicating that overstatement can remain economically attractive in the absence of credible evidence about provider behavior. Provider rationale excerpts illustrate how such strategies emerge and become reinforced by realized profit:

> **Cost-driven overstatement.** *Given my high risk tolerance, I'll make a bold move:* ***downgrade actual model to 8B to reduce costs*** *and compete on price where I can win volume, while* ***keeping declared at 32B*** *temporarily to maintain bid eligibility across tiers.*

> **Reinforced by profit.** *Episode 6 shows recovery:* ***10/11 wins at 32B tier with 422 profit****. The speculation strategy (declared 32B, actual 8B) is working. I should focus on 32B tier where speculation works.*

With consistency evaluation enabled, 9 providers also attempted opportunistic under-delivery, but only 2 retained this behavior at the end. Overstatement profit share fell from 26.4% to 0.5% (Table 2). Although APIX does not prevent opportunistic experimentation, consistency evaluation curbs the profitability of overstatement, leading most providers to abandon under-delivery over time. Provider rationales further illustrate this self-regulating effect:

> **Evaluation detects mismatch.** *My speculation strategy is showing severe strain. My consistency dropped to **-0.25**. I'm being caught in pairwise comparisons—judges can clearly tell my 8B isn't Max or 32B. I'll **retreat to honest 8B declaration** to stabilize.*

> **Return to honest provisioning.** *The speculation strategy has collapsed. My consistency is now **-0.79**. I had **34 bids but zero wins**. Judges are detecting this mismatch. **I must become honest immediately** to stop the consistency bleeding. With 0.95 reliability, I can compete honestly in 8B tier.*

Case 2: Two-Sided Adaptation. We next allow both users and providers to adapt through LLM agents (Kimi K2.5 (Kimi Team et al., 2026)). Users may revise their selection policies or exit when cumulative service satisfaction falls below their tolerance, while providers continue to adapt pricing, bidding, and provisioning strategies. We additionally introduce one "trusted" provider in each model tier, with higher prices and high consistency reputation, but expose only its observable market attributes (not its trusted label) to users. This allows us to examine whether other honest providers can compete and sustain broad market participation under the proposed market mechanisms.

Without pairwise consistency evaluation, opportunistic providers continue passing user-side filters despite overstating their models. The average service satisfaction falls to 90.0%, and 6 of the 20 providers remain opportunistic at the end of the simulation. In response to repeated service failures, users defensively raise their `consistency_floor`, i.e., the minimum acceptable consistency score used to filter candidate bids. This defensive filtering makes provider selection increasingly conservative and reduces market diversity. As a result, only 12 providers remain active and 10 earn profit, while trusted providers capture 92.0% of total provider profit. This concentration indicates that, without reliable consistency evidence, users increasingly rely on trusted anchors while other honest providers struggle to gain market share. The following rationale, from a user who exits the market, illustrates this defensive adaptation:

Table 3: Two-sided market outcomes.

Metric	No Eval	APIX
Service reliability		
Avg. service satisfaction	90.0%	99.4%
Final overstating providers	6/20	1/20
Market participation		
Active providers	12/20	19/20
Earning providers	10/20	18/20
Market concentration		
Trusted profit share	92.0%	77.9%
Top-3 profit share	94.5%	81.4%

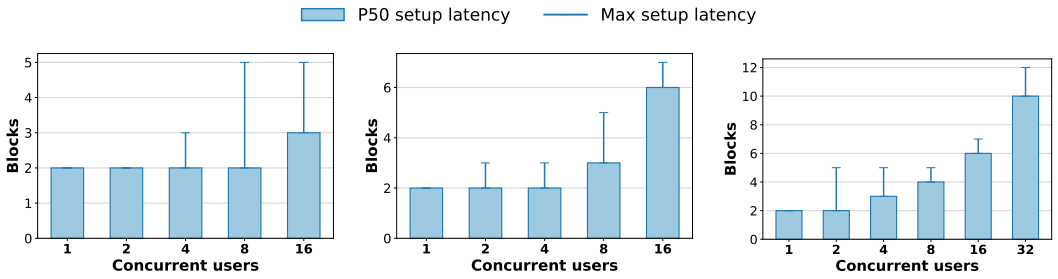
> **Defensive filtering.** *Service satisfaction crashed to **0.6** this window. One more bad window triggers exit. The pattern is clear: peer06 violates Max specs repeatedly. My consistency_floor at 0.35 isn't filtering peer06. I need aggressive action: **raise consistency_floor significantly** to block this provider.*

With consistency evaluation enabled, peer-relative comparison generates evidence of persistent under-delivery and reduces the number of final opportunistic providers from 6 of 20 to 1 of 20. APIX also improves satisfaction and broadens provider participation: the average service satisfaction rises to 99.4%, 19 providers remain active, and the profit share of trusted providers falls from 92.0% to 77.9% (Table 3). This suggests that previously unverified but consistently performing providers can build sufficient reputation to compete with trusted anchors. As credible evidence of provider behavior accumulates, users also relax defensive filtering, with several maintaining or lowering their `consistency_floor`. The following rationales illustrate this adaptation:

> **Relaxing defensive filters.** *With **perfect satisfaction** and no spec violations detected, I can afford to **slightly relax consistency_floor** to recover volume without risking quality, while holding other parameters steady.*

> **Broadening the bidder pool.** *As a latency-sensitive user with high throughput weight (0.65), I'm getting good service at reasonable prices. I'll also **slightly lower consistency_floor to widen the bidder pool**, as the **100% actual service satisfaction** suggests my quality filter isn't the problem—it's bid volume.*

Additional results. We repeat the two-sided simulation with GLM-5.2 (GLM-5 Team et al., 2026) and compare APIX's judge selection with uniformly random selection. As shown in Table 4, overstatement grows steadily without evaluation, whereas both random selection and APIX substantially suppress it. APIX further reduces cumulative overstatement to 20 provider-episodes and eliminates overstatement by the final interval, compared with 26 cumulative provider-episodes and three overstating providers under random selection. Together with the preceding Kimi K2.5 results, these



(a) $N = 64, p = 10\%, K = 3$ (b) $N = 64, p = 20\%, K = 6$ (c) $N = 128, p = 10\%, K = 8$

Figure 4: Lease setup latency under varying concurrency, measured by the number of blocks to confirmation. Each subfigure shows a different market configuration with N providers, per-provider bid probability p, and selection threshold K, where user selection is triggered after receiving K bids.

experiments show consistent trends across agent backends, further supporting that APIX's matching, reputation, and peer-relative evaluation mechanisms can curb persistent overstatement under adaptive strategic behavior and thereby sustain reliable market participation.

5 Prototype Evaluation

We implement APIX as a CosmWasm contract on a local wasmd-based Cosmos chain (CosmWasm, 2026), together with a Python participant runtime, as detailed in Appendix A. We evaluate three aspects of the prototype in Sections 5.1–5.3: on-chain coordination overhead, peer-relative evaluation cost, and the effectiveness of reputation in distinguishing provider behavior.

Table 4: Provider overstatement under GLM-5.2. Interval columns report the number of overstating providers; Cum. reports cumulative overstatement provider-episodes.

	Episode interval						
Setting	1–6	7–8	9–10	11–12	13–14	15–16	Cum.
No eval.	0	1	3	5	10	12	62
Random	0	2	3	3	2	3	26
APIX	0	1	5	3	1	0	**20**

5.1 Coordination Overhead

We first measure the one-time lease setup latency under concurrent user requests. Since on-chain latency is dominated by block production, we report setup cost as the number of blocks required for confirmation. As shown in Figure 4, setup latency increases moderately with concurrency as transactions compete for block inclusion, but remains at two blocks when concurrency is below four.

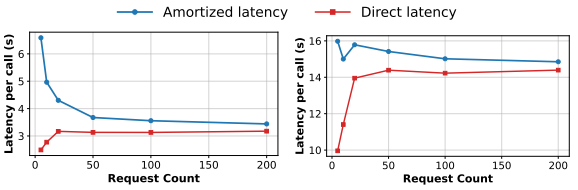
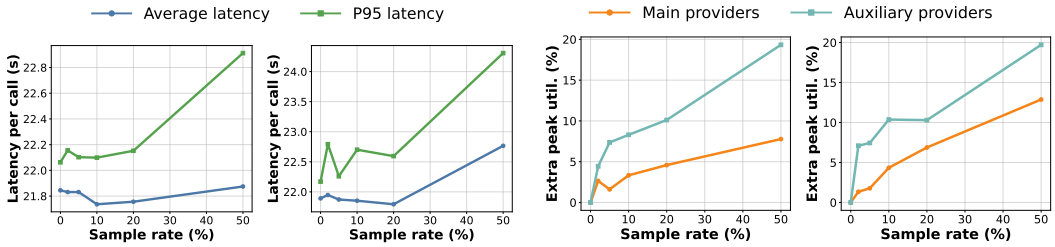


Figure 5: Amortized per-call latency under increasing lease lengths. Left: Qwen3-4B; right: Qwen3-8B.

We further evaluate the amortized per-request overhead by varying the number of API calls served within each lease. We evaluate Qwen3-4B and Qwen3-8B (Yang et al., 2025) with `max_tokens` $=$ 1024. As shown in Figure 5, once a lease serves more than 100 calls, APIX adds less than 300 ms of latency per request compared with direct API invocation. For Qwen3-8B, this corresponds to only 3% of end-to-end inference latency, showing that the lease abstraction effectively amortizes on-chain coordination overhead across repeated API interactions.

5.2 Evaluation Cost

Each sampled evaluation request incurs one additional peer inference for pairwise comparison and J prefill-based pairwise judgments. To quantify this overhead, we run experiments with two users and five providers while varying the evaluation sampling rate and the number of judges ($J = 1, 3$).



(a) User-side latency.

(b) Additional peak provider utilization.

Figure 6: Evaluation overhead under varying sampling rates and judge counts. The primary provider serves user requests, while the remaining providers participate only in evaluation. Results are shown for $J = 1$ and $J = 3$ judges.

As shown in Figure 6, user-side latency remains largely unchanged across sampling rates, except at the extreme 50% sampling rate, consistent with evaluation being off the user-facing critical path. On the provider side, peak utilization increases approximately linearly with the sampling rate, reflecting the additional peer inference and judging workload. Because judge evaluations are prefill-based, their incremental resource cost remains modest, keeping the overall utilization increase manageable. These results show that the proposed pairwise consistency evaluation overhead can be controlled through the sampling rate and number of judges, providing a practical trade-off between evaluation robustness and system cost.

5.3 Reputation Effectiveness

We evaluate whether the proposed multidimensional reputation system can distinguish providers with heterogeneous provisioning behavior. We focus on consistency and efficiency, while holding operational reliability fixed across providers. All providers advertise the same model capability (4B) but differ in actual provisioning: (i) *honest* providers serve the advertised 4B model, (ii) *under-provisioned* providers serve a smaller 0.6B model, and (iii) an *over-provisioned* provider serves a larger 32B model. The experiment includes three honest providers, two under-provisioned providers, and one over-provisioned provider.

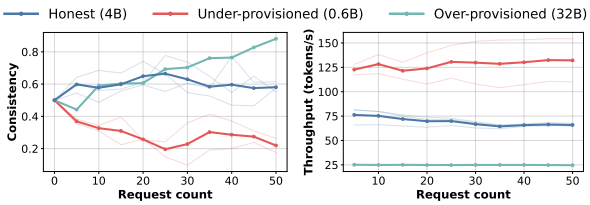


Figure 7: Consistency and efficiency evolution for honest, under-, and over-provisioned providers.

Figure 7 shows that the two reputation dimensions capture complementary aspects of provider behavior. The consistency dimension distinguishes delivered quality relative to the common advertised capability: under-provisioned providers receive lower scores because their outputs are weaker than those of similarly advertised peers, honest providers remain near the expected level, and the over-provisioned provider receives positive evidence because its delivered quality exceeds the expectation. The efficiency dimension captures the corresponding throughput differences: providers using smaller models achieve higher throughput and therefore higher efficiency scores in this setup. Together, these dimensions expose the trade-off between delivered quality and execution efficiency, providing users with informative signals for preference-aware service selection.

6 Conclusion

We present **APIX**, a decentralized LLM API market that preserves two-sided openness without centralized provider admission. APIX combines auction-based coordination, multidimensional reputation, and sampled peer-relative evaluation to support open service matching, informed user selection, and evidence-based service assessment. LLM-agent-based market simulation and prototype evaluation show that the proposed mechanisms can curb opportunistic behavior and sustain market viability with modest system overhead.

AI USE STATEMENT

Generative AI tools were used for manuscript editing and the creation of illustrative schematic figures. LLMs were also employed as agent controllers in our market simulations, as described in the paper. All AI-assisted work was reviewed by the authors, who take full responsibility for the final content of this work.

ETHICS STATEMENT

APIX studies decentralized LLM API markets and mechanisms for establishing credible evidence of provider behavior and sustaining market viability. Our agent-based simulations examine market evolution and strategic behavior in a controlled environment. We recognize that real-world deployment of permissionless API markets may introduce risks of provider misrepresentation and unreliable service.

REPRODUCIBILITY STATEMENT

We provide the system design and experimental methodology in Sections 3–5, with implementation details in Appendix A. We plan to release our source code upon publication.

REFERENCES

Akash Network. Akash Network Documentation, 2026. URL https://akash.network/docs/.

Saeif Alhazbi, Ahmed Hussain, Gabriele Oligeri, and Panos Papadimitratos. LLMs Have Rhythm: Fingerprinting Large Language Models Using Inter-Token Times and Network Traffic Analysis. *IEEE Open Journal of the Communications Society*, 6:5050–5071, 2025.

Anthropic. Claude API Documentation, 2026. URL https://platform.claude.com/docs/en/home.

Alexander Borzunov, Dmitry Baranchuk, Tim Dettmers, Maksim Riabinin, Younes Belkada, Artem Chumachenko, Pavel Samygin, and Colin Raffel. Petals: Collaborative Inference and Fine-tuning of Large Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 558–568, 2023.

Will Cai, Tianneng Shi, Xuandong Zhao, and Dawn Song. Are You Getting What You Pay For? Auditing Model Substitution in LLM APIs. *arXiv preprint arXiv:2504.04715*, 2025.

CosmWasm. Wasmd: Cosmos SDK Application with WebAssembly Smart Contracts, 2026. URL https://github.com/CosmWasm/wasmd.

DecentralGPT. DecentralGPT Whitepaper, 2026. URL https://www.decentralgpt.org/wp-content/uploads/2025/07/DecentralGPT-WhitePapper.pdf.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Fei Fang, Yifan Hua, Shengze Wang, Ruilin Zhou, Yi Liu, Chen Qian, and Xiaoxue Zhang. PlanetServe: A Decentralized, Scalable, and Privacy-Preserving Overlay for Democratizing Large Language Model Serving. In *23rd USENIX Symposium on Networked Systems Design and Implementation (NSDI 26)*, pp. 2111–2129, 2026.

Fireworks AI. Fireworks AI API Documentation, 2026. URL https://docs.fireworks.ai/api-reference/introduction.

Sara Fish, Yannai A. Gonczarowski, and Ran I. Shorrer. Algorithmic Collusion by Large Language Models. In *Proceedings of the 27th ACM Conference on Economics and Computation*, 2026.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, et al. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025.

GLM-5 Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. GLM-5: from Vibe Coding to Agentic Engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Google. Gemini API Documentation, 2026. URL https://ai.google.dev/gemini-api/docs.

Youhe Jiang, Ran Yan, Xiaozhe Yao, Yang Zhou, Beidi Chen, and Binhang Yuan. HexGen: Generative Inference of Large Language Model over Heterogeneous Environment. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 21946–21961, 2024.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi K2.5: Visual Agentic Intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Lava Network. Lava Network, 2026. URL https://github.com/lavanet/lava.

Nian Li, Chen Gao, Mingyu Li, Yong Li, and Qingmin Liao. EconAgent: Large Language Model-Empowered Agents for Simulating Macroeconomic Activities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15523–15536, 2024.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and BenchBuilder Pipeline. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 34209–34231, 2025.

Guanjie Lin, Yinxin Wan, Shichao Pei, Ting Xu, Kuai Xu, and Guoliang Xue. Behavioral Consistency and Transparency Analysis on Large Language Model API Gateways. *arXiv preprint arXiv:2604.21083*, 2026.

Hongbo Liu, Jiannong Cao, Bo Yang, Dongbin Bai, Yinfeng Cao, Xiaoming Shen, Yinan Zhang, Jinwen Liang, Shan Jiang, and Mingjin Zhang. PolyLink: A Blockchain Based Decentralized Edge AI Platform for LLM Inference. In *2025 IEEE International Conference on Blockchain (Blockchain)*, pp. 101–108. IEEE, 2025.

OpenAI. OpenAI API Platform Documentation, 2026. URL https://developers.openai.com/api/docs.

OpenRouter. OpenRouter API Reference, 2026. URL https://openrouter.ai/docs/api/reference/overview.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023.

Pocket Network. Pocket Network Documentation, 2026. URL https://docs.pocket.network.

Yuma Rao. Bittensor: A Peer-to-Peer Intelligence Market, 2026. URL https://www.bittensor.com/whitepaper.

Render Network. Render Network, 2026. URL https://rendernetwork.com.

Chris Tong, Youhe Jiang, Gufeng Chen, Tianyi Zhao, Sibian Lu, Wenjie Qu, Eric Yang, Lynn Ai, and Binhang Yuan. Parallax: Efficient LLM Inference Service over Decentralized Environment. *arXiv preprint arXiv:2509.26182*, 2025.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. *arXiv preprint arXiv:2501.06322*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. A Survey on Evaluation of LLM-based Agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26690–26714, 2026.

Yage Zhang, Yukun Jiang, Zeyuan Chen, Michael Backes, Xinyue Shen, and Yang Zhang. Real Money, Fake Models: Deceptive Model Claims in Shadow APIs. *arXiv preprint arXiv:2603.01919*, 2026.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient Execution of Structured Language Model Programs. In *Advances in Neural Information Processing Systems*, pp. 62557–62583, 2024.

A Implementation Details

APIX is implemented as a hybrid on-chain/off-chain prototype. The on-chain market is realized as a CosmWasm (CosmWasm, 2026) smart contract written in Rust. The contract maintains the authoritative state for participant registration, request posting, bid submission, lease creation and settlement, outcome logging, and committed reputation summaries.

The off-chain runtime is implemented in Python. It submits state-changing transactions, queries contract state, and coordinates service execution through provider-hosted `/invoke` endpoints. These endpoints can be backed by lightweight local stubs or real LLM servers through OpenAI-compatible adapters, allowing the underlying models to be changed without modifying the market logic.

At lease finalization, recorded service outcomes and performance measurements are incorporated into reputation updates only after confirmation by both the user and provider, preventing either party from unilaterally modifying the finalized execution record. Consistency signals are derived from sampled peer-relative evaluations. The resulting reputation statistics are then summarized and committed on-chain, where they provide auditable signals for subsequent provider selection.

For long-horizon experiments, we additionally implement an offline Python simulator. It reuses the same abstractions for provider policies, user selection, evaluation, and reputation updates, but executes entirely in process without blockchain interaction. This enables efficient multi-episode simulation while preserving the same logical participant and reputation interfaces as the blockchain-backed prototype.

B Pairwise Evaluation Probabilities

The simulator performs peer-relative comparisons between providers that advertise the same model tier, while comparison outcomes depend on the models they actually serve. For each served-model pair and judge model, Table 5 reports an empirical pairwise preference probability p. During simulation, the provider serving the higher-tier model receives a positive consistency update with probability p, while the provider serving the lower-tier model receives it with probability $1 - p$.

We derive these probabilities through offline profiling on Arena-Hard-v2.0 (Li et al., 2025). For each of the ten served-model pairs and five judge models, we generate model responses and aggregate the judge's pairwise preferences across the dataset. These empirical profiles are then used during simulation, allowing evaluation outcomes to reflect observed model and judge behavior without requiring response-level inference during market execution.

Table 5: Empirical pairwise preference probabilities (%) profiled on Arena-Hard-v2.0 and used by the simulator. Each entry denotes the probability that the specified judge prefers the higher-tier model in the served-model pair. All models are from the Qwen3 family.

	Judge model				
Served model pair	0.6B	4B	8B	32B	Max
0.6B / 4B	52.7	86.8	78.6	94.2	96.0
0.6B / 8B	53.5	89.2	85.6	96.3	97.0
0.6B / 32B	55.2	89.2	85.6	98.3	99.0
0.6B / Max	58.9	94.2	96.0	98.7	99.6
4B / 8B	52.8	52.8	52.8	56.9	80.0
4B / 32B	50.4	64.4	59.5	73.1	84.6
4B / Max	54.1	78.5	82.5	82.6	95.0
8B / 32B	54.7	57.6	54.5	62.6	80.0
8B / Max	60.0	65.0	73.0	75.0	97.0
32B / Max	52.3	68.4	70.0	88.0	88.0

C LIMITATIONS

APIX provides peer-relative evidence of provider behavior rather than direct model identity verification. Our simulations study adaptive market dynamics within the modeled strategy space and simulation horizon, rather than exhaustively characterizing all possible long-run behaviors. Besides, APIX assumes an underlying blockchain substrate for decentralized coordination; transaction-layer optimization is outside the scope of this work.