FOMO: A Demand-Driven Inference Protocol for Decentralized AI

FLock.io

December 2025

Abstract

In this paper, we present the FLock Open Model Offering (FOMO), a demand-driven inference protocol that completes the decentralized AI (DeAI) lifecycle within the FLock ecosystem. Today, AI consumption is structurally misaligned: inference is typically purchased as a stateless commodity from centralized gateways, resulting in rigid pricing, limited cost programmability, and near-total value capture by intermediaries. Meanwhile, model developers and operators struggle to translate real usage into sustainable, transparent economic outcomes.

FOMO addresses these frictions through a franchise-style token economy in which each model deployment is mapped to a revenue-backed class of tokens, called Model Tokens (MT). Through inference-linked buybacks, token burns, emissions, and usage-weighted rewards, FOMO converts model consumption into measurable economic signals and routes value back to the participants who drive adoption. Integrated with AI Arena, a supply-side protocol for decentralized model training, FOMO establishes a closed-loop architecture that jointly incentivizes the production, deployment, and use of AI models in a decentralized setting.

1 Introduction

AI inference has become a foundational input for modern software, yet the economic structure of AI consumption remains brittle. As decentralized AI (DeAI) systems mature,[1] the missing piece is not model supply alone, but a demand-side mechanism that coordinates inference, incentives, and value distribution without centralized intermediaries. Most users access models through centralized cloud or gateway providers, purchasing inference as a stateless commodity: users pay per request, prices are largely non-programmable, and the economic upside of adoption accrues almost entirely to intermediaries. As a result, the entities that drive real demand—developers, integrators, and power users—are treated purely as cost centers rather than aligned stakeholders.

This creates two persistent frictions:

(1) Rigid pricing and limited cost programmability. Inference costs tend to be dictated by centralized providers and are difficult to structurally reduce beyond short-term discounts. For AI-native products whose margins depend on inference unit economics, pricing rigidity becomes a scaling bottleneck: increasing usage increases cost in a near-linear fashion.

(2) No upside for usage, and weak reinvestment loops. In centralized settings, increased adoption primarily increases provider revenue. Users do not participate in the upside they create, and model-level economies lack native mechanisms for reinvesting demand into long-term sustainability, such as funding future hosting, improving liquidity, or rewarding the stakeholders who bootstrap adoption.

These problems also exacerbate the commercialization gap for specialized models, including domain-specific small language models (SLMs). While such models can deliver high domain value, they often lack efficient distribution channels, transparent market signals, and sustainable incentive structures that tie usage to ongoing deployment viability.

Against this background, we address the demand-supply imbalance in decentralized AI through a two-layer architecture:

- **AI Arena**[2]: A supply-side protocol that incentivizes the *production of intelligence* through decentralized model creation and training competitions.
- **FOMO**: A demand-side protocol that incentivizes the *dissemination of intelligence* through economically programmable, usage-linked inference markets.

AI Arena bootstraps training and evaluation, while FOMO transforms inference usage into token burns, buybacks, and reward pathways that route value to the participants responsible for adoption. Together, they form a full-stack DeAI framework in which both model creation and model consumption are cryptoeconomically optimized.

FOMO restructures the economics of inference by coupling usage with programmable cost reduction and value routing:

- **Lower, programmable inference costs.** Users can stake the deployment-specific Model Token (MT) to unlock discounted API usage for that model. In addition, the protocol bootstraps early adoption by routing rewards to productive deployments via $FLOCK$ emissions, and by distributing vesting-based incentives (including MT incentive emissions) based on usage-weighted allocation.
- **Usage translates into value for users, not only providers.** A portion of inference revenue is programmatically routed to buy back and burn the relevant MT, strengthening the economics of the model that users rely on. In parallel, protocol-level revenue supports $FLOCK$ buybacks

and treasury funding, enabling reinvestment into ongoing operations and future hosting of model deployments.

2 System Architecture Overview

Specifically, FOMO operationalizes a **Franchise Model for AI Inference**, where each model deployment becomes its own micro-economy with predictable incentives, transparent performance signals, and automated token sinks tied directly to usage.

2.1 Entities

FOMO and AI Arena operate through a set of onchain assets and participant roles. To avoid conceptual ambiguity, we separate these into **Token Assets** and **Participant Roles**.

2.1.1 Token Assets

FLOCK Token ($FLOCK$)

- Network-wide utility and governance asset for the entire DeAI ecosystem.
- Serves as the medium for protocol-level buybacks, burns, and emissions.
- Represents macro-level demand for intelligence production and usage.

Model Token (MT)

- Deployment-specific token representing the economic state of a single model, with a given deployment set-up. In other words, a specific model can have various set-ups and thus be represented by different MT.
- Backed by inference revenue, buyback-and-burn pressure, and model-level emissions.
- Functions as the coordination asset for each model deployment's micro-economy.

2.1.2 Participant Roles

FLock Platform

- Provides the infrastructure for model deployment, revenue accounting, buyback logic, and emission distribution.
- Ensures cryptographic verification of inference usage and transparent execution of all tokenomic flows.
- Maintains platform-wide economic parameters (e.g., discount caps, emission schedules, fee structures).

Real Model Assets (RMA) Issuer

- Initiates a Real Model Offering (RMO) by configuring architecture, hosting tier, and pricing floors. In essence, this is the equivalent of issuing RMAs, as shown in Figure 1.
- Operates the deployment and receives a share of emissions, creator fees, and the 10% inference yield.
- Acts as the economic steward of the model's micro-economy.

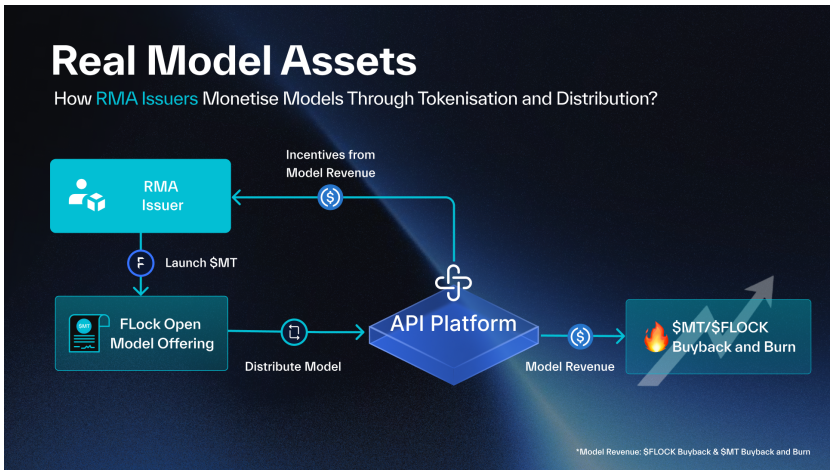

Figure 1: Interaction between RMAs Issuers and FOMO.

Users and RMAs Supporters

- As shown in Figure 3, users pay for inference, generating revenue and triggering burn mechanics.
- They lock MT to access inference discounts and receive rewards proportional to model usage.
- As shown in Figure 2, RMA supporters acquire MT during the RMO and can stake it on the FOMO platform to earn incentives (e.g., usage-linked emissions) while benefiting from MT and $FLOCK$ buyback-and-burn driven by model revenue.
- Together they form the demand-side capital base that sustains the model economy.

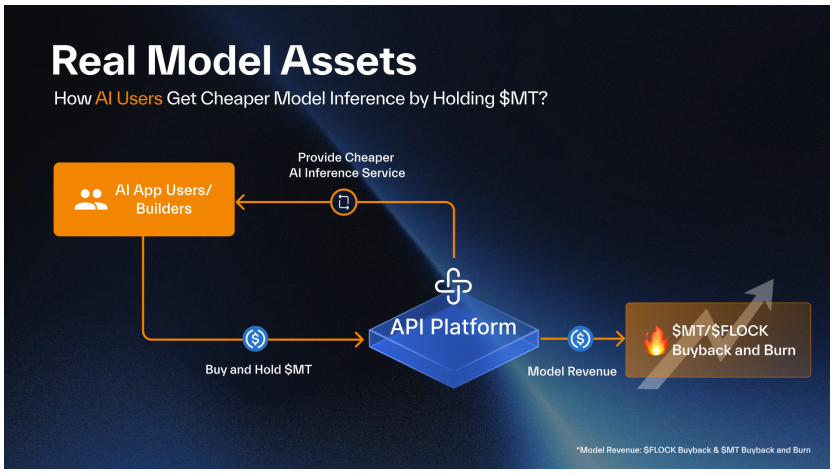

Figure 2: Interaction between RMAs Supporters and FOMO.

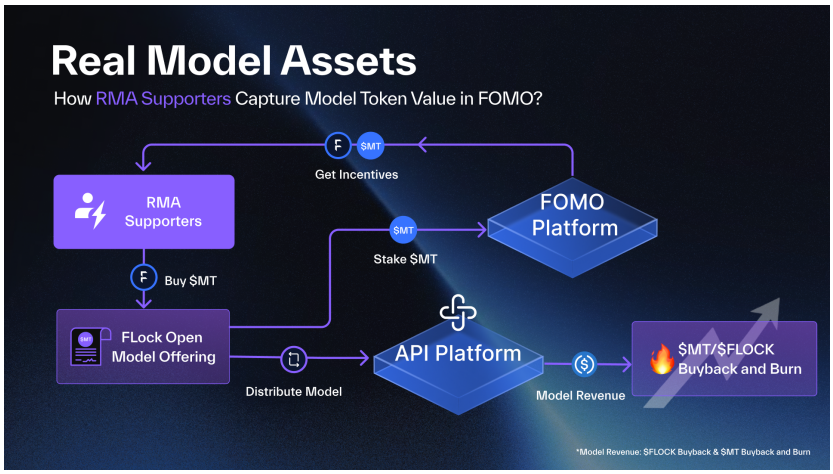

Figure 3: Interaction between API Users and FOMO.

AI Arena Contributors

- Train, fine-tune, or enhance SLMs within AI Arena competitions.
- Receive $FLOCK$ emissions for verified training contributions.
- Provide the supply of models that ultimately become FOMO deployments. Note however models hosted on FOMO are not limited to models trained on AI Arena.

3 End-to-End Lifecycle of a FOMO Deployment

To provide an intuitive understanding of the protocol, we outline the complete lifecycle of a model deployment within FOMO. Each stage corresponds to an economic or cryptographic transformation, ultimately linking model usage to token value and participant rewards.

3.1 Model Creation and Registration

A model originates either from AI Arena on the FLock platform or other model vendors. High-performing models or fine-tuned variants are candidates for FOMO deployment. A prospective RMA Issuer selects a model and configures deployment parameters:

- Model architecture (e.g., Qwen2.5-72B, Llama-3-8B) and inference configuration
- Hosting tier (defining minimum OpEx and performance guarantees)
- Minimum inference price (economic floor)
- Fundraise bounds (minimum and maximum raise)

This configuration initializes a RMO. Hosting tier selection ensures that tokenholders understand the cost structure and operational limits of the deployment.

3.2 RMO

The RMO serves as a transparent fundraising phase for the deployment. A constant-product internal market allows participants to purchase Model Tokens (MT) using $FLOCK$.

Each deployment mints a fixed supply:

$$1{,}000{,}000\ MT$$

Allocated as follows:

- 40% Public Sale

- 20% Initial Liquidity Pool
- 20% Incentive Reserve (6-month vest)
- 10% RMA Issuers' Allocation (6-month vest)
- 10% Platform Treasury (6-month vest)

Similar to Uniswap v2, the internal sale in FOMO uses a constant-product market maker:

$$x \cdot y = k$$

Users purchase MT using $FLOCK$. At this stage, selling back into the internal market is disabled to maintain fundraising integrity and a clear forward-only capital formation process.

Per trade during the RMO:

- 0.30% Creator Fee $\rightarrow$ RMA Issuer
- 0.95% Protocol Fee $\rightarrow$ Treasury

If the raise fails to meet the minimum target, all raised funds are refunded. If successful, liquidity—combining raised $FLOCK$ and the liquidity allocation of MT—is migrated to a public DEX, and the deployment proceeds to live service.

3.3 Bonding Curve Pricing Mechanism

During the RMO, MT tokens are sold via a constant-product automated market maker (CP-AMM) with *virtual reserves*. This mechanism provides continuous price discovery while enforcing forward-only capital formation (i.e., purchases are enabled during fundraising, while sell-backs into the internal pool are disabled until post-RMO secondary trading).

3.3.1 Constant-Product Market

Let x denote the $FLOCK$ reserve (virtual + real) and let y denote the MT reserve (virtual). The pool maintains the invariant:

$$x \cdot y = k.$$

Buyers deposit $FLOCK$ into the pool and withdraw MT. Let D be the cumulative amount of $FLOCK$ raised so far (with $0 \leq D \leq G$). If the pool is initialized with virtual reserves (x_0, y_0), then:

$$x(D) = x_0 + D, \qquad y(D) = \frac{x_0 y_0}{x_0 + D}.$$

The cumulative number of MT tokens sold by the curve at raise level D is:

$$S(D) = y_0 - y(D) = y_0 - \frac{x_0 y_0}{x_0 + D} = \frac{y_0 D}{x_0 + D}.$$

Fundraising terminates when either (i) the cumulative raise reaches the *graduation target* G, or equivalently (ii) the cumulative sale reaches the target number of tokens sold S (e.g., $S = 400{,}000$ for the 40% public sale).

3.3.2 Parameterization

The curve is parameterized by:

- S: target amount of MT to be sold during the RMO (e.g., 400,000),
- G: graduation target in $FLOCK$ to be raised,
- P_0: desired initial marginal price in $FLOCK$ per MT at the start of the curve,
- x_0: initial virtual $FLOCK$ reserve,
- y_0: initial virtual MT reserve (typically $y_0 > S$).

At the start of the curve, the instantaneous (marginal) price is approximately:

$$P_0 \approx \frac{x_0}{y_0}.$$

Imposing that exactly S tokens are sold when $D = G$ yields:

$$S = \frac{y_0 G}{x_0 + G} \quad \Rightarrow \quad y_0 = \frac{S(x_0 + G)}{G}.$$

Substituting $x_0 = P_0 y_0$ gives closed-form expressions for the virtual reserves:

$$y_0 = \frac{GS}{G - SP_0}, \qquad x_0 = P_0 y_0 = \frac{GSP_0}{G - SP_0}.$$

This construction implies the feasibility constraint:

$$G > SP_0,$$

i.e., the graduation target must exceed the starting price multiplied by the number of tokens sold, otherwise the curve would be undefined.

3.3.3 Operational Degrees of Freedom

At launch, the protocol can expose (or internally set) the key economic knobs:

- **Graduation target** G: how much $FLOCK$ must be raised before liquidity is migrated to a public DEX.
- **Starting price** P_0: the initial marginal price of MT in $FLOCK$.

In practice, choosing SP_0 close to (but strictly below) G produces a smoother curve, reducing early/late buyer price dispersion while retaining monotonic price appreciation.

3.3.4 Worked Example

Consider a deployment with:

$$S = 400{,}000, \qquad G = 50{,}000, \qquad P_0 = 0.1 \ FLOCK \text{ per } MT.$$

The feasibility check holds:

$$SP_0 = 400{,}000 \cdot 0.1 = 40{,}000 < 50{,}000 = G.$$

Virtual reserves are therefore:

$$y_0 = \frac{GS}{G - SP_0} = \frac{50{,}000 \cdot 400{,}000}{50{,}000 - 40{,}000} = 2{,}000{,}000, \qquad x_0 = P_0 y_0 = 200{,}000.$$

If a user purchases with $D = 100$ $FLOCK$ total raised so far, then:

$$S(100) = \frac{y_0 \cdot 100}{x_0 + 100} \approx \frac{2{,}000{,}000 \cdot 100}{200{,}000 + 100} \approx 999.5,$$

corresponding to an average execution price of approximately:

$$\frac{100}{999.5} \approx 0.10005 \ FLOCK \text{ per } MT.$$

As D increases toward G, the curve monotonically increases the marginal price, reflecting increasing demand and decreasing remaining MT inventory.

3.3.5 DEX Graduation and Initial Liquidity Price

Upon graduation, a portion of raised $FLOCK$ and the designated liquidity allocation of MT are migrated to a public DEX pool. If 50% of raised $FLOCK$ is paired with 200,000 MT for initial liquidity, then the initial DEX price is:

$$P_{\text{DEX}} = \frac{0.5G}{200{,}000} = \frac{25{,}000}{200{,}000} = 0.125 \ FLOCK \text{ per } MT.$$

This provides a transparent, onchain transition from primary issuance (RMO) to secondary market trading with protocol-defined liquidity depth.

3.4 Capital Formation and Use of Proceeds

During the RMO, 40% of the MT supply is sold. The $FLOCK$ raised is allocated:

- 25% Protocol Allocation (6-month vest)
- 25% MT Staking Incentives (6-month vest)
- 50% Liquidity Provision (paired with 20% MT)

The liquidity provision is locked for 48 months to ensure deep, stable markets for MT and to align long-term model performance with token liquidity.

3.5 Active Inference and Revenue Generation

Users invoke a given model, which is paired with a graduated MT, via the FLock API platform, paying for inference in USDC or fiat-equivalent assets. Every inference call produces verifiable usage records.

Let P_{gross} be total inference spend in USDC and let OpEx denote compute costs paid to providers:

$$P_{\text{net}} = P_{\text{gross}} - \text{OpEx}$$

The protocol settles P_{net} according to a deterministic revenue waterfall:

60% Protocol Fee

- 30% of P_{net} is used to buy back $FLOCK$.
- 30% of P_{net} is retained by the treasury.

30% Model Buyback Used to purchase and burn MT from the DEX pool:

$$MT_{\text{burn}} \propto 0.30 \cdot P_{\text{net}}$$

10% RMA Issuers' Yield 10% of P_{net} is distributed in USDC to the Model Token Owner as operational yield.

This stage is where real-world demand becomes onchain economic impact, creating a direct link between usage, macro-token deflation, and model-level scarcity.

3.6 Staking and Discount Dynamics

Users may stake MT to reduce their inference costs and participate in the model's reward flows. Staking simultaneously:

- Reduces circulating supply of MT
- Increases scarcity and potential price stability
- Entitles stakers to model-specific $FLOCK$ emissions and incentive streams

Users stake MT to receive inference discounts governed by:

$$\text{Discount} = \min\left(X\%,\ \frac{\text{User Staked MT}}{\text{Total Circulating MT}} \cdot M\right)$$

where:

- $X\%$ is the protocol-wide maximum discount.
- M is a multiplier defining discount sensitivity.

Discounts are capped via $X\%$ to preserve net revenue discipline, ensuring that the protocol maintains a robust base of P_{net} to fund buybacks, burns, and yields.

3.7 Deployment Score and Emission Routing

Daily $FLOCK$ emissions are distributed proportionally to each deployment's **Deployment Score (DS)**:

$$DS_i = \text{Revenue}_i \cdot \text{AgeFactor}_i$$

in which

$$\text{AgeFactor}(t) = \max\left(0.2,\ 1 - \frac{t}{180}\right)$$

Aging reduces emissions over time while preserving a 20% floor to ensure long-term viability and to prevent older but still useful deployments from being fully displaced by new entrants.

In other words, newer and higher-revenue deployments thus earn disproportionately higher rewards, reflecting both freshness and performance.

3.8 Internal Redistribution of Emissions

Each model's share of the daily global $FLOCK$ emission is further distributed:

- 10% RMA Issuers' Bonus (franchise operator reward)
- 90% Staker Rewards

To prevent penalizing power users who utilize their discounts, rewards are calculated based on *gross* revenue (what the cost would have been without discount):

$$\text{UserReward} = \text{Pool} \cdot \frac{\text{GrossSpend}_{\text{user}}}{\text{GrossSpend}_{\text{total}}}$$

where Pool is the total reward pool for a given model and epoch.

Two additional vesting-based reward sources use the same weighting:

- 25% Raised $FLOCK$ (Staking Fund)
- 20% MT Incentive Reserve

This ensures that:

- High-usage deployments earn more emissions
- Stakers benefit in proportion to their economic contribution
- The protocol rewards productive deployments instead of static capital

3.9 Long-Term Equilibrium

Over time, MT supply decreases through burns, $FLOCK$ becomes scarcer through protocol buybacks, and successful deployments grow into sustainable, revenue-backed micro-economies. Underperforming models naturally receive fewer emissions and attenuated token demand, guiding the ecosystem toward efficient resource allocation at both the model and network level.

3.10 Miscellaneous Fees and Costs

In addition to the primary fundraising and pricing mechanics, participants should be aware that launching a MT and participating in an RMO incurs several ancillary costs. These fees are designed to support protocol operations, discourage adversarial behavior, and align early participation with long-term deployment health.

- **One-time launch fee.** A fixed, non-recurring fee payable by the RMA Issuer at the time of initiating the RMO. This fee covers deployment setup, onchain configuration, and protocol overhead associated with launching a new model economy.
- **Trading and platform fees.** A percentage-based fee applied to all MT purchases during the RMO, including the initial bundle purchased at launch. These fees are deducted at the time of each transaction and are routed to the protocol according to predefined fee splits.
- **Anti-sniping fee.** To mitigate opportunistic early trading and promote fair price discovery, an anti-sniping tax is applied immediately following launch. The tax starts at a high rate and decays linearly over a fixed time window until it reaches the base trading fee.

All fees described above are payable in $FLOCK$ and *non-refundable*. Fees remain payable and are not returned even if MT does not successfully graduate (i.e., fails to reach the fundraising target required for liquidity migration). This policy ensures economic finality, discourages frivolous launches, and preserves protocol-level incentive integrity.

4 Example: QwenALICE

To demonstrate how FOMO functions in practice, this section presents a complete, end-to-end walkthrough of a model deployment called "QwenALICE." The example follows both the Model Token Owner (Alice) and a representative user and staker (Bob) through the lifecycle of fundraising, usage, emissions, and rewards.

4.1 Launch and Fundraising

Alice configures a high-performance deployment and initiates an RMO for QwenALICE. The parameters are:

- Total MT supply: 1,000,000 QwenALICE tokens
- RMO sale allocation: 400,000 QwenALICE (40% of supply)
- Target raise: \$100,000 denominated in $FLOCK$

The internal bonding-curve sale clears successfully:

- The RMO raises \$100,000 in $FLOCK$
- All 400,000 sale tokens are purchased by participants

4.2 Use of Proceeds

The \$100,000 in raised $FLOCK$ is allocated according to FOMO's standard funding structure:

- \$25,000 (25%) $\rightarrow$ Protocol allocation (6-month vest)
- \$25,000 (25%) $\rightarrow$ QwenALICE staking incentives (6-month vest)
- \$50,000 (50%) $\rightarrow$ Liquidity provision, paired with 200,000 QwenALICE (20% of supply) on a DEX

Liquidity is locked for 48 months, anchoring a deep and predictable secondary market for QwenALICE.

4.3 Month 1 Inference Economics

In its first month, QwenALICE gains traction and generates:

- Gross inference revenue: included in P_{gross}
- Compute (OpEx) paid to providers
- Net revenue:
$$P_{\text{net}} = \$10{,}000$$

The inference revenue waterfall allocates this net revenue:

Protocol Capture (60% of P_{net})

- \$3,000 used to buyback $FLOCK$
- \$3,000 retained by the FLock Treasury

Model Buyback (30% of P_{net})

- \$3,000 used to buy and burn QwenALICE from the DEX

RMA Issuers Yield (10% of P_{net})

- \$1,000 paid directly to Alice in USDC

Thus, from \$10,000 net:

- \$3,000 increases $FLOCK$ scarcity
- \$3,000 increases QwenALICE scarcity
- \$3,000 supports the protocol treasury
- \$1,000 flows to the RMA Issuers

4.4 Age Factor and Emission Capture

Assuming QwenALICE has been live for one month ($t \approx 30$ days), its Age Factor is:

$$\text{AgeFactor}(30) = \max\left(0.2,\ 1 - \frac{30}{180}\right) = 0.83$$

The Deployment Score becomes:

$$DS_{\text{QwenALICE}} = \text{Revenue}_{\text{QwenALICE}} \cdot \text{AgeFactor}$$

Because QwenALICE is both young and high-performing, it earns a relatively large share of the daily global $FLOCK$ emission.

4.5 Emission Distribution and Alice's Position

Once QwenALICE receives its portion of the global emission, the internal split is:

- 10% RMA Issuers' bonus $\rightarrow$ Alice
- 90% $\rightarrow$ stakers, weighted by gross spend

Two additional reward streams use the same weighting:

- 25% of raised $FLOCK$ (staker incentives)
- 20% of MT supply (incentive reserve)

If Alice stakes her 10% allocation, she captures:

- The guaranteed 10% RMA Issuers' bonus bonus
- A proportional share of the 90% staker pool

Effectively, she may capture 20% of emissions depending on staker participation.

4.6 RMA Issuer Outcome Snapshot (Alice)

Alice's position now includes:

- 10% QwenALICE allocation (vested)
- $1,000 per month from the RMA Issuer revenue share
- A share of staker emissions (if she stakes)
- 10% of the deployment's daily $FLOCK$ emissions

This yields:

- Deflationary upside from MT buybacks
- Macro-token alignment from $FLOCK$ buybacks
- Direct USDC yield
- Emission-backed token flows

Together, these components define the economic role of a franchise operator within FOMO.

4.7 Staker Perspective: Bob

To illustrate the demand-side incentives, consider Bob, a user who participates in two ways:

1. He buys QwenALICE during the RMO.
2. He stakes to receive discounts and emissions.

RMO Participation Bob purchases:

$$\text{50,000 QwenALICE} \quad \text{(12.5\% of the 400k sale allocation)}$$

Staking for Discounts Assume circulating supply at launch:

$$\text{Circ}_0 = 400{,}000 \text{ (sale)} + 200{,}000 \text{ (LP)} = 600{,}000.$$

Bob's stake share:

$$\frac{50{,}000}{600{,}000} = 0.0833 \ (8.33\%).$$

With protocol parameters:

- Max discount: $X = 30\%$
- Multiplier: $M = 2$

Bob receives:

$$\text{Discount}_{\text{Bob}} = \min(30\%, 2 \times 8.33\%) = 16.67\%.$$

Reward Share via Gross Spend Weighting Assume:

- Total gross spend: \$50,000
- Bob's gross spend: \$2,500

Bob accounts for 5% of usage:

$$\frac{2{,}500}{50{,}000} = 0.05.$$

If QwenALICE receives:

$$20{,}000\ FLOCK \text{ per day}$$

then the staker pool is:

$$18{,}000\ FLOCK.$$

Bob's reward:

$$18{,}000 \times 0.05 = 900\ FLOCK \text{ per day}.$$

Bob's Net Position Bob benefits from:

- 16.67% inference discount
- 900 FLOCK/day in rewards
- Exposure to MT deflation through buybacks
- Liquidity optionality on the DEX

His behavior increases:

- Model usage
- Revenue
- Buyback-and-burn pressure
- Emissions routed to QwenALICE

Thus, Bob's incentives are fully aligned with the growth of the deployment.

4.8 Summary

This example shows how:

- RMA Issuers earn yield, emissions, and deflationary upside.
- Stakers earn discounts and emission rewards weighted by usage.
- Users reinforce the economic loop through inference spend.
- The protocol continuously buyback $FLOCK$ and burn MT.

QwenALICE illustrates how a successful deployment becomes a self-sustaining, market-driven micro-economy within the larger FOMO ecosystem.

5 Participant Incentives and Economic Alignment

FOMO is designed such that every participant receives rewards aligned with the value they contribute to the system. This section outlines the incentive structure for each role.

5.1 RMA Issuers

RMA Issuers serve as franchise operators. They are incentivized through:

- **10% of net inference revenue** as recurring cash yield
- **Creator fees** during the RMO bonding curve sale
- **10% of model deployment-level** $FLOCK$ emissions
- **10% MT allocation**, vested over 6 months

By selecting performant models and competitive pricing, RMA Issuers directly increase revenues, buyback pressure, and long-term token appreciation.

5.2 RMA Supporters (Stakers and Power Users)

Stakers act as the demand-side liquidity base for a model's micro-economy. Their incentives include:

- **Tiered inference discounts**, scaled by their share of staked MT
- **90% of model-level** $FLOCK$ **emissions**
- Access to vested reward streams:
  - 25% of raised $FLOCK$ allocated to staking incentives
  - 20% MT incentive reserve
- **Exposure to MT scarcity**, strengthened by buybacks and burn pressure

Because emissions are weighted by gross spend, heavy users are rewarded without being penalized for using their discounts.

5.3 Investors and Liquidity Providers

Investors participate due to:

- Deflationary MT supply driven by revenue-based buybacks
- Liquidity depth and long-term locked pools on DEXes
- Predictable issuance and vesting schedules

This creates a transparent environment for price discovery.

5.4 Compute Providers

Compute providers benefit from:

- Steady, verifiable inference demand
- Clear margins defined by hosting tiers
- Onchain settlement and risk-mitigated revenue collection

This expands the supply of verifiable compute.

5.5 FLock Platform

The platform itself earns:

- 30% of net inference revenue
- 0.95% protocol fee during RMO sales
- Long-term treasury growth through vesting

The platform is structurally aligned with model usage and ecosystem expansion.

5.6 AI Arena Contributors

Contributors indirectly benefit from FOMO growth:

- Increased $FLOCK$ demand strengthens Arena training incentives
- Arena output gains commercialization pathways through RMOs
- Market feedback loops direct training toward high-demand domains

Arena and FOMO thus form a dual-sided incentive loop reinforcing each other.

5.7 General Users

Even users who do not stake receive value indirectly:

- Access to highly specialized SLMs with transparent pricing
- Confidence that protocol fees route toward token sinks and ecosystem growth

6 Integration with AI Arena

Together, FOMO and AI Arena form a closed-loop DeAI economy.

6.1 Arena → FOMO (Supply Drives Demand)

- Arena participants train SLMs and receive $FLOCK$ rewards.
- High-quality models are deployed through FOMO RMOs.

6.2 FOMO → Arena (Demand Funds Supply)

- Inference usage generates burns and buybacks.
- Buybacks increase the value of $FLOCK$, strengthening Arena incentives.
- Usage data informs future Arena reward calibration.

The system forms a positive-sum flywheel:

Training → Deployment → Usage → Burns & Rewards → More Training

7 Conclusion

To conclude, FOMO provides a deterministic, incentive-aligned mechanism for DeAI inference and usage. Combined with AI Arena's training incentives, it forms the first full-stack DeAI system where both production and consumption of intelligence are cryptoeconomically rewarded. SLMs and vertical models—previously constrained by narrow commercialization pathways—can now operate as self-sustaining, onchain micro-economies.

References

[1] Zhipeng Wang, Rui Sun, Elizabeth Lui, Vatsal Shah, Xihan Xiong, Jiahao Sun, Davide Crapis, and William Knottenbelt. Sok: Decentralized ai (deai). *arXiv preprint arXiv:2411.17461*, 2024.

[2] Zhipeng Wang, Rui Sun, Elizabeth Lui, Tuo Zhou, Yizhe Wen, and Jiahao Sun. Aiarena: A blockchain-based decentralized ai training platform. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 1375–1379, 2025.