NFA

Stop Trading on Instinct

Verify the trade before you make it. AI advocates argue each possible outcome, an impartial judge weighs the evidence independently of the market, and you get a probability you can audit before you trade.

WHITEPAPER · VERSION 1.2 · JULY 2026 · NFA.CLUB

NFA · Trade Verification for Polymarket

Verify the trade before you make it. AI advocates argue each possible outcome, an impartial judge weighs the evidence independently of the market, and you get a probability you can audit before you trade.

Abstract

NFA verifies the trade before you make it. A user selects a Polymarket market; the engine researches it into a sourced evidence corpus, then runs a structured debate in which one AI advocate argues the strongest case for each possible outcome, and the advocates rebut each other across rounds. An impartial judge weighs the competing cases against the evidence and returns a calibrated probability distribution across the outcomes, an independent forecast rather than an echo of the market. Every run is fully logged, and accuracy is scored against resolved markets.

The product addresses a specific gap in the prediction-market ecosystem: traders on Polymarket have access to price feeds, historical data, and news, but lack structured tools for forecasting outcomes driven by actor behavior under pressure: geopolitical events, regulatory decisions, policy outcomes, narrative dynamics, and behavioral markets. This class of markets represents 30 to 40 percent of active prediction-market inventory by count and a higher share of volume during periods of crisis, elections, and major news events. NFA serves this gap, and the same engine forecasts statistical and microstructure markets too.

Three design principles shape the architecture. **First**, accuracy is the single metric that matters: all product, economic, and incentive decisions optimize for measurable forecasting accuracy. **Second**, the engine is production-grade from day one. Transparent end-to-end logging, per-call cost reconciliation, per-purpose LLM routing, prompt caching, and within-round parallelism are architectural primitives, not optimizations. **Third**, the forecast is independent: NFA produces its own estimate from the evidence, so the divergence between NFA and the market is the signal to trade rather than an echo of the crowd.

NFA launches on **Base**. The token, settlement, and vesting are deployed as audited on-chain Solidity contracts; the forecasting engine is a proprietary stack of permissively licensed open-source components glued together by a custom orchestration layer.

This whitepaper describes the product, the technical architecture, the token model, and the path to launch.

1 · Problem

1.1 The prediction market category is growing fast

Prediction markets have matured from fringe experiments into a significant category of financial infrastructure. Polymarket crossed billions in volume during the 2024 US election cycle and has kept expanding across politics, economics, crypto, and culture. New markets continue launching around cultural events, crypto outcomes, corporate actions, and geopolitical developments. Both retail and institutional trading activity has grown consistently.

The category works because prediction markets aggregate information efficiently. Prices reflect the collective best estimate of thousands of participants, often outperforming polls, expert forecasts, and traditional surveys. But the efficiency of market-based aggregation has a ceiling set by the quality of information available to participants. Where participants have access to strong tools, markets price efficiently. Where participants rely on intuition, markets reflect that limitation.

1.2 No single tool gives traders a calibrated probability

Prediction markets broadly fall into three categories by what determines their resolution:

- **Statistical markets** where historical data carries most of the signal: sports outcomes, base-rate questions, weather. Strong single-purpose models exist (ELO, xG, weather, actuarial data), but they are fragmented and rarely output one calibrated, tradeable probability for the specific market in front of you.
- **Microstructure markets** where price action over short horizons determines resolution: crypto price movements on 5-minute or hourly intervals. Traders use on-chain analytics, orderbook data, and funding rates.
- **Actor-driven markets** where outcomes depend on how specific actors behave under specific pressures: geopolitical events, regulatory decisions, policy outcomes, corporate actions, behavioral markets involving specific individuals or organizations. Traders have essentially no structured tools. They rely on news reading, intuition, and crowd-sourced takes on social media.

NFA serves all three. Its research-debate-judge loop grounds every forecast in researched evidence and base rates, so it returns one calibrated probability whether the question is statistical or actor-driven, folding the relevant data and models into the judge's reasoning rather than ignoring them. Its **deepest edge is in actor-driven markets**, the third category, where no structured tool exists today and where volume concentrates during elections, crises, and regulatory cycles. But the same engine forecasts a statistical market just as cleanly, giving traders a single tool across the board.

1.3 Why existing approaches fall short

Expert panels and forecasting tournaments. Services like Good Judgment Project have demonstrated that trained human forecasters can outperform intelligence agencies on specific question classes. But expert panels are slow, expensive, and don't scale. They cannot be applied in real-time to the thousands of markets active on Polymarket at any given time.

Single-LLM forecasters. Direct LLM prompting ("will this peace deal happen?") produces shallow, unreliable forecasts. Single models lack the structured reasoning required for multi-actor dynamics, don't maintain consistent mental models of adversarial incentives, and hallucinate specifics that undermine credibility. LLMs are necessary but insufficient.

News aggregators and sentiment trackers. Tools that aggregate news coverage or track social sentiment provide useful signal but don't produce forecasts. A trader still has to translate "sentiment is 65 percent negative" into "probability is X percent." The translation is the hard part, and it's exactly what traders need help with.

Generic multi-agent frameworks. Running several LLM agents against a question improves on single-LLM forecasting but produces inconsistent, unauditable results without structure. NFA's structure is adversarial and adjudicated: one advocate is required to argue each possible outcome from a sourced evidence corpus, and a separate judge weighs the competing cases against that evidence. The arguments are comparable across runs, the judgment is auditable, and the forecast is made independently of the market price.

1.4 The opportunity

A structured forecasting engine that argues every possible outcome and adjudicates the evidence independently of the market, with a rigorously measured accuracy record, fills a real gap, most acutely in actor-driven markets where no structured tool exists today, but general across the board. No incumbent occupies this position. The distribution channel (MCP, prediction-market frontends) is open. The technical substrate has matured to production-grade in the last 18 months. The audience is measurable and growing. The timing is right.

2 · Solution

2.1 Research, advocates, and a judge

NFA's core engine produces a forecast in three steps. **First**, it researches the market into a sourced **evidence corpus**: the actors involved, the timeline, the established facts, and the open questions. **Second**, it runs a structured **debate**, instantiating one AI advocate for each possible market outcome; each advocate argues the strongest evidence-based case for its outcome, and the advocates rebut one another across several rounds. **Third**, an impartial **judge** weighs the competing cases against the evidence and emits a calibrated probability distribution across the outcomes.

The design is deliberately adversarial. A single model collapses a hard question into one estimate and drifts toward the obvious answer; forcing a dedicated advocate to argue each outcome surfaces the strongest case on every side, and a separate judge must reconcile them against the evidence before ruling. Crucially, the judge is **never shown the market price**: it forecasts from the evidence and base rates alone, so the output is an independent estimate, an edge to trade against the market rather than a reflection of it.

The engine is general-purpose. It does not hard-code any market or domain; it researches whatever market it is given and argues whatever outcomes that market defines. Reusable category templates (§2.2) can tune how a class of market is researched and judged, but the same research-debate-adjudicate loop runs on every market.

2.2 Category templates

Every run executes against a **category template**: a reusable configuration for a recurring kind of market, a negotiation, a regulatory vote, a central-bank decision, an election. A template tunes how that class of market is researched and how the debate and judge are parameterized; it never names who plays in any specific market, because the actors, evidence, and outcomes are discovered per run from the market itself. The same election template applied to two different races draws a different field, different evidence, and a different verdict.

These templates are maintained internally. The engine selects the template that fits each market automatically; there is nothing for the user to configure or choose.

2.3 Distribution

NFA reaches users through three surfaces:

- **Web frontend** at `nfa.club`. Retail traders select a market, run forecasts, and inspect the debate and the full reasoning trace. The primary consumer surface.
- **MCP server.** Machine-facing API. Trading agents, algorithmic desks, and AI assistants like Claude and ChatGPT will consume NFA programmatically. MCP is the standardizing protocol for

AI agent tooling; early positioning in public registries is a durable distribution advantage.

3 · Product

3.1 The user flow

A trader's primary interaction with NFA is simple: select a Polymarket market from the in-app catalogue. The engine fetches the market's metadata and outcomes, selects the category template that fits the market, researches the market, runs the advocate debate, and returns the judge's probability distribution with a full reasoning trace. There is no link to copy, no configuration step, and no template to choose: one click, one forecast.

The in-app catalogue is the way in, and it keeps the whole workflow on the platform. NFA scans Polymarket for the markets where it has a potential edge and presents them grouped by category with the live price, so a trader browses to a market and forecasts it in one click, without copying links between tabs or leaving for another site. Programmatic callers can still address a specific market directly through the MCP server (§3.4).

Every market is served by NFA's own templates. There is no template to pick and no setup: the engine routes the market to the right template itself.

3.2 Forecast output

When a forecast runs, the output includes:

- **Probability distribution** across every market outcome.
- **Reasoning trace**: each advocate's case, the rebuttals exchanged, and the judge's written rationale, fully auditable.
- **Divergence summary** comparing NFA's probability to the current market price, outcome by outcome, highlighting where, and by how much, the engine disagrees with the market.
- **Accuracy context**: the engine's historical accuracy on similar markets, so the user can calibrate confidence in the output.

> ▸ The reasoning trace is a critical product feature. Traders do not trust black-box forecasts. An auditable trace lets users evaluate not just the final probability but the quality of the reasoning that produced it.

3.3 The frontend

Live market gallery. A continuously updated feed of active Polymarket markets that NFA has forecast. Each entry shows the current market price, NFA's independent probability, and the divergence between them. Sortable by divergence magnitude, accuracy, recent activity, or market volume. This is the content engine: every large divergence is a shareable piece of content.

Forecast runner. Pick a market from the catalogue and the run streams in real-time, showing the advocates argue and the judge rule. Final output includes the probability distribution, reasoning trace, divergence summary, and accuracy context.

3.4 The MCP server

For programmatic consumption, NFA will offer an MCP (Model Context Protocol) server. Trading agents, AI assistants, and custom bots will integrate via standardized tool calls, with the server listed in public MCP registries.

Exposed tools include:

```
forecast_market(market_url)                // research, debate, judge → distribution
get_forecast(market_url)                   // latest cached forecast + divergence
compare_markets(market_urls)               // forecast across related markets
explain_forecast(forecast_id)              // advocate cases + judge rationale, from the l
list_forecasts(category, min_accuracy)     // recent forecasts by category
```

Access draws on the same credit balance. Trading agents pay per forecast at the same per-run price as the web frontend.

3.5 Referrals

Anyone can share a referral link. When someone they refer buys credits, the referrer earns **10% of that purchase, paid out in USDC**, and it applies to **every** purchase that referral makes, not just the first, so bringing in active traders compounds over time. Self-referral is blocked, and the reward is funded from platform revenue as a customer-acquisition cost rather than minted, so it never dilutes the token.

3.6 Acting on a forecast

A forecast is only useful if a trader can act on it, so NFA lets a trader take the position without leaving the app. Trading is **non-custodial**: the trader signs every order with their own wallet, and NFA never holds, moves, or has access to their funds.

From a forecast, or from the portfolio, a trader can place both **market** and **limit** orders on either side of a market, set a protective exit to close a position at a target price, and cancel any resting order before it fills. The same order ticket works whether the trader is opening a new position or closing an existing one.

Paper trading is available today, so a trader can follow the engine and track hypothetical positions with no funds at risk; non-custodial live trading with real funds is being brought online.

4 · Technical architecture

The forecasting engine is production-grade by design. The properties below, transparent logging, per-purpose routing, caching, evidence grounding, and reconciled cost tracking, are architectural primitives, not afterthoughts.

4.1 Transparent logging

Every run is logged end-to-end: the research corpus, each advocate's case, the rebuttals, and the judge's rationale, alongside the per-call model and cost metadata. Nothing about a forecast is hidden. The full trace is persisted and inspectable, so any score can be traced back to the exact evidence and reasoning that produced it. This is what makes the accuracy record auditable rather than something taken on faith.

4.2 Per-purpose LLM routing

LLM calls within a run serve different purposes with different cost-quality elasticity: research, advocacy, and judging are routed independently. NFA routes each call class to the appropriate model tier rather than running everything on a single model, yielding a meaningful cost reduction at no measurable quality loss versus all-frontier execution. Routing is configurable per run; advanced traders can opt into all-frontier routing for marginally higher accuracy at higher compute cost, and category templates can require frontier models for call classes where the forecast depends on it.

4.3 Prompt caching

A typical run reuses the same evidence corpus, advocate instructions, and judging context across dozens of LLM calls. Prompt caching exploits this structurally on every supported provider, substantially reducing both input-token cost and time-to-first-token after the first call in a window. Combined with per-purpose routing, this is the difference between optimized and unoptimized engine economics.

4.4 Parallel debate

The debate parallelizes by design. Every outcome's opening case is dispatched at once, each rebuttal round fans out across all advocates simultaneously, and the judge runs as a parallel self-consistency ensemble, all with bounded provider concurrency. Rebuttal rounds are the only sequential dependency. Rate-limit and transient-failure handling are first-class. The net effect is the difference between a forecast that returns in tens of seconds and one that takes many minutes.

4.5 Cost record persistence and reconciliation

Every LLM call emits a cost record with full provenance, provider, model, call class, and the source from which the cost was computed. A reconciliation job runs on a regular cadence comparing engine-reported run totals against provider billing API records, with any discrepancy flagged for

manual review before settlement. This is essential: the per-run credit price is derived from metered model cost, which must accurately reflect actual spend.

4.6 Evidence grounding and adversarial defense

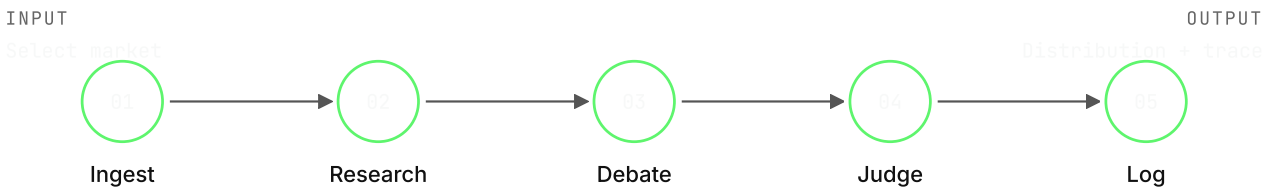
Every advocate's claims are checked against the research corpus before the judge weighs them. The validator catches unsupported assertions, contradictions with the established facts, and arguments that lean on fabricated specifics rather than the evidence, the failure mode that makes raw LLM forecasts untrustworthy. The judge is instructed to treat the corpus as ground truth and to discount any case that strays from it, so a persuasive-but-baseless argument cannot move the verdict.

This grounding step is intentionally routed to frontier models because its accuracy directly determines forecast integrity. The cost is justified by the trust it protects.

Two further defenses protect the forecast's independence and integrity. The engine works from a **blind brief**: the market's own price is stripped out of the evidence the advocates and the judge see, and the pipeline guards against that price leaking back in through quoted news or related-market context, so the forecast is formed from the evidence rather than anchored to the crowd. And at ruling time, the **decisive claims** the verdict actually turns on are checked back against the sourced corpus before the ruling stands, so a verdict resting on a claim the evidence does not support is caught rather than published.

4.7 Data flow

A forecast proceeds through five stages.



Stage 1: Ingest. The trader selects a market. The system pulls its metadata from the Polymarket API: rules text, resolution criteria, deadline, current prices, volume, category, the outcome set, and related markets. Parsed into a structured representation, and the category template for the market is selected.

Stage 2: Research. The research layer runs targeted web research on the specific market and synthesizes a sourced **evidence corpus**: the actors involved, the timeline, the established facts,

and the open questions. The corpus is frozen for the rest of the run, so every run works from the same fixed evidence base.

Stage 3: Debate. One advocate is instantiated per market outcome. Each argues the strongest evidence-based case for its outcome from the corpus, and the advocates rebut one another across several rounds. Every claim is checked against the evidence (§4.6), so unsupported assertions carry no weight.

Stage 4: Judge. An impartial judge, run as a self-consistency ensemble, weighs the competing cases against the corpus and emits a normalized probability distribution across the outcomes. Returned with the full reasoning trace and a divergence summary versus the market.

Stage 5: Log. The run is logged with full state for audit. When the underlying market later resolves, the forecast is scored by Brier against the realized outcome, feeding the engine's accuracy record.

4.8 Live-metric adapters

A large share of markets resolve on a single measurable number: a closing price, a daily high temperature, a CPI print, a Treasury yield, a chokepoint transit count. For these, the decisive input is not an argument but the **current value of the resolution metric**, read from the source the market actually settles against. A frozen research corpus or a one-off web lookup can miss or misread that number, and on metric-resolved markets that is the single largest source of error.

NFA closes this with a registry of deterministic **live-metric adapters**: structured fetchers that pull the exact resolution figure directly from its authoritative source, keyless wherever possible, and inject it into the shared evidence both the advocates and the judge see. The current value of the metric goes from hopefully-retrieved to always-present-and-correct.

#	Adapter	Covers
1	**PortWatch**	Maritime chokepoint transit counts, including the Strait of Hormuz
2	**Yahoo v8**	Equities, indices, commodities, and FX, in one adapter
3	**BLS**	CPI, jobs, and unemployment, the highest mechanical edge, keyless
4	**NWS · CLI**	Daily city high and low temperatures, the weather-market source
5	**FRED**	Fed-funds level, PCE, and oil and yield mirrors
6	**Treasury XML**	Treasury yields, read from the exact resolution source
7	**Binance**	Crypto spot, the literal resolution source, with a US fallback

#	Adapter	Covers
+	Long tail	NHC storms · Frankfurter FX · USGS earthquakes · FiscalData debt · ESPN standings · NCEI climate records · Open-Meteo companion

TABLE 1 · LIVE-METRIC ADAPTER REGISTRY · EACH ADAPTER TARGETS THE LITERAL SOURCE A MARKET RESOLVES AGAINST

The registry covers an estimated **90 to 95 percent** of metric-resolved markets with **zero paid API keys**, and a generic web fallback handles the remainder. Because each adapter reads the same figure the market settles on, the engine reasons from the resolution number itself rather than a stale or approximate proxy, the highest-leverage accuracy improvement available on the largest and most liquid market categories.

Coverage is defended in the build itself: a permanent coverage matrix runs in continuous integration, so a market type the registry is meant to reach cannot silently regress to the generic web fallback. And when an adapter fires for a run, the forecast is marked **anchored** (§5.8), so a trader can see at a glance that the estimate rests on the live resolution figure rather than on argument alone.

4.9 Run pricing and credits

Run cost scales with market complexity, the number of outcomes, and the routing choices. With per-purpose routing and prompt caching, the metered model cost per run stays well below all-frontier execution.

Forecasts are paid in **credits**, topped up in **USDC** with a **$10 minimum**. Each run draws down credits equal to its **compute fee**: the run's metered model cost plus a platform fee. Because model cost varies with market complexity, the fee is quoted as a band before the run and settled within it. Larger top-ups carry volume discounts (**30% above $100** and **50% above $500**), so heavy users pay less per run. The platform's net revenue from runs funds the $NFA buy-and-burn described in §6.1.

Grading also governs billing. A forecast the engine flags as a **no-go** (§5.8), one its own evidence audit cannot stand behind, is **not charged**: the run's credits are waived and the balance is left untouched. A trader pays for the forecasts the engine is willing to stand behind, not for the ones it flags as unreliable, which keeps the platform's incentive aligned with the trader's.

5 · Accuracy as the north star

5.1 Why accuracy matters more than other metrics

Prediction-market trading is fundamentally about expected value. A trader who can identify mispriced markets earns money over time; a trader who cannot does not. The tool that helps traders identify mispricings is only valuable to the extent it is more accurate than the market it is being used against.

Accuracy is the only metric that matters at the foundational level. Every other decision, the engine's design, pricing, and coverage, is downstream of accuracy. The product is designed to produce accurate forecasts and to measure accuracy rigorously.

5.2 Measuring the track record

Every forecast NFA produces is recorded, and when the underlying market resolves the run is scored against the outcome. The record breaks accuracy down by market category, with full details of every resolved run, and it is complete: no cherry-picking, no selective reporting. The engine is measured this way from day one; the track record is published once its consistency, edge, and sample size are locked in, so the numbers reflect a validated engine rather than a launch-week sample.

Measuring completely, not selectively, is a discipline from the start. **User trust** depends on a record that cannot be cherry-picked, and when the record is published it will be the whole of it, including the categories and periods where NFA underperforms. A complete, honestly measured record is also the cleanest regulatory footing: NFA reports analysis backed by measurement, not opaque claims.

5.3 Scoring methodology

Brier scores are the primary accuracy metric. For a binary market with resolution outcome `o ∈ {0, 1}` and predicted probability `p ∈ [0, 1]`, the Brier score is `(p - o)²`. Lower is better; `0.0` is perfect, `0.25` is the score achieved by always predicting 50/50.

For multi-outcome markets, scoring generalizes to mean squared error across all outcome categories.

Accuracy is displayed as a 0-to-1 score where 1.0 represents perfect forecasting and 0.0 represents random baseline (0.25 raw Brier). More intuitive for non-expert users while preserving statistical properties.

5.4 Category-specific accuracy

A single aggregate accuracy number is misleading, so NFA measures accuracy per market category rather than in aggregate. Once a category has a proven record, a user submitting a

market from it sees the engine's accuracy specifically on that category, not one blended figure.

5.5 Comparative benchmarking

NFA benchmarks its forecast accuracy against:

- Polymarket market consensus at forecast time;
- Naive baselines (always 50/50, category base rate);
- Single-LLM forecasts (frontier models on identical questions).

These benchmarks are the honest check on whether the platform adds value; the results are validated as the resolved-market sample grows, and published once the record proves out, including where NFA underperforms.

5.6 Continuous improvement mechanics

- **Automatic category expansion.** When a category reaches sufficient activity without good coverage, the platform flags the gap and builds a new template for it.
- **Staleness detection.** Templates whose recent accuracy trends down are flagged for refresh.
- **Winning pattern extraction.** Analysis of top-performing forecasts informs engine-level improvements.
- **Governance feedback.** Accuracy data surfaces platform parameters that may need adjustment.

5.7 Base rates and calibration

Two mechanisms keep the judge honest about how often a class of event actually happens. A **reference-class base rate** anchors the forecast to how often comparable events have resolved in the past, so a vivid argument cannot pull the estimate far from the historical frequency without evidence that this case is genuinely different. And for the markets where the mechanism is well understood, notably short-horizon crypto price levels, NFA prices the outcome with a **calibrated statistical model**, a first-passage model over the live price, rather than argument alone. Across the board, the engine's raw probabilities are periodically recalibrated against how its past forecasts actually resolved, so a stated 70% means the event happened about seven times in ten.

5.8 The engine grades each forecast, and can veto itself

An accuracy record tells a trader how the engine does on average. Grading tells them how much to trust *this* forecast, before the market resolves. Every run carries a grade that reflects how well-supported the estimate is, so a trader can weight a strongly-evidenced forecast differently from a thin one.

The grade is not self-assessment by the same reasoning that produced the forecast. An **independent evidence audit** re-checks the decisive claims the verdict rests on against the sourced corpus, and the engine can **veto its own forecast** when that audit does not hold up. The bands:

- **Audited edge.** The decisive claims are individually sourced and the divergence from the market survives the audit. The engine's highest-conviction call.
- **Anchored.** A live-metric adapter (§4.8) supplied the resolution figure, so the estimate rests on the number the market settles against, not on argument alone.
- **Standard.** A well-formed forecast the audit stands behind, without a decisive sourced edge or a live anchor.
- **No-go.** The audit cannot stand behind the forecast, so the engine declines to publish an edge rather than serve one it does not trust.

Grading is what lets NFA hold coverage honestly: rather than force a low-quality forecast on every market, the engine flags the ones it cannot stand behind, and a no-go of that kind is not charged (§4.9).

6 · Token economics

6.1 Token utility

$NFA is an ERC-20 token on **Base**. As an Ethereum L2, Base offers the low fees and fast confirmation that make continuous open-market buy-and-burn viable at retail price points, while keeping the token in the same EVM ecosystem as Polymarket and NFA's on-chain trading integration. Runs are paid in **credits** topped up in USDC, which keeps pricing legible and removes token-acquisition friction for new users; $NFA is the value-capture and access layer on top of that economy. Its core roles are below.

Buy-and-burn from platform revenue. Net revenue from forecast runs is used to buy $NFA on the open market and burn it. The buyback & burns will happen randomly to avoid frontrunning.

Usage-mining. A five-month launch program distributes the Airdrop allocation to the wallets that spend credits on forecasts, weighted by that spend, turning early usage directly into token distribution (§6.4).

Governance. $NFA holders govern platform parameters: pricing, accuracy weighting, treasury policy, category weighting, and protocol upgrade decisions. Governance is weighted by tokens held.

Token demand scales with real usage rather than speculation: platform revenue continuously buys and burns supply as forecasts run.

6.2 Supply and distribution

Total supply is fixed at **1,000,000,000 NFA** tokens.

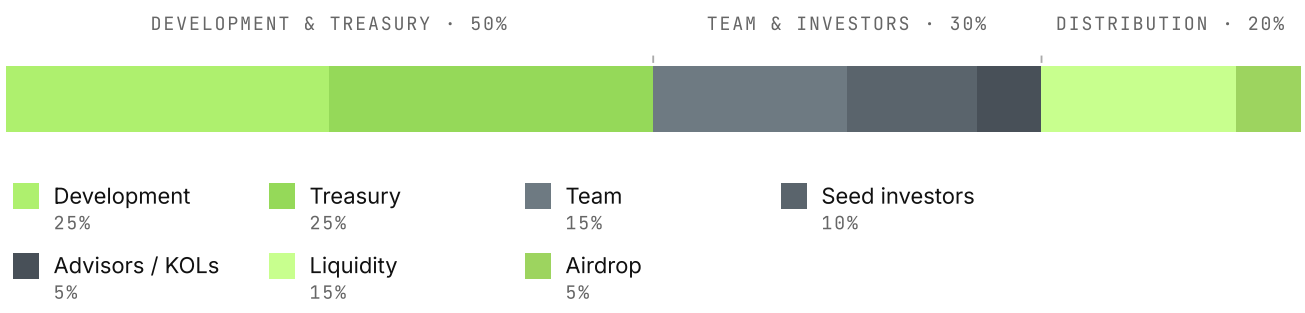


FIG. 2 · TOKEN ALLOCATION · 1,000,000,000 NFA

- **Development, 25%.** Engineering, infrastructure, security audits, and scaling the product beyond the initial MVP.

- **Treasury, 25%.** Protocol-owned reserve, governed on-chain: accuracy and ecosystem bounties, trader-onboarding incentives, integrations and partnerships with prediction-market venues, AI-agent platforms and data providers, and a long-term growth fund for expansion not anticipated at launch.
- **Team, 15%.** Core contributors, long-term alignment, governance and operations.
- **Seed investors, 10%.** Honor commitments from the initial raise.
- **Advisors / KOLs, 5%.** Strategic contributors and distribution partners.
- **Liquidity, 15%.** Initial liquidity provisioned across Base DEXs at TGE (Aerodrome, Uniswap), ensuring efficient price discovery and depth for the continuous buy-and-burn flow.
- **Airdrop, 5%.** Community distribution via the five-month usage-mining program (§6.4), rewarding the wallets that fund early forecast runs.

6.3 Vesting schedules

All vesting is enforced on-chain through immutable contracts. No party has governance discretion to alter vesting after launch. TGE-unlock and vesting percentages are of each allocation; vesting is linear from the end of any cliff.

ALLOCATION	SUPPLY	TGE UNLOCK	CLIFF	VESTING
Seed investors	10%	10%	—	9-month linear
Team	15%	10%	—	18-month linear
Advisors / KOLs	5%	0%	3 months	9-month linear
Development	25%	33%	—	12-month linear
Treasury	25%	0%	1 month	24-month linear
Liquidity	15%	50%	—	remainder unlocks at month 6
Airdrop	5%	0%	1 month	5-month linear (usage-mining, §6.4)

6.4 Usage-mining

NFA runs a usage-mining program from launch that rewards the act of running forecasts. For the first **five months** following public availability, the protocol emits **0.033333% of total supply per day** (about **1% per month**). Each day's emission is distributed **by the weight of credits spent on forecasts** that day: a wallet's share of the day's pool equals its share of all credits spent running forecasts, so the reward tracks how much forecasting each wallet actually paid for, not a flat per-run split. Five months of daily emissions draw the entire **Airdrop allocation** (§6.2), with no separate budget line.

Because the daily pool is fixed, each credit's share is richest exactly when total usage is lowest, a built-in incentive to be early, and the open-market activity it drives feeds the same buy-and-burn

flow that runs in steady state. The subsidy comes from token issuance the protocol controls, not treasury cash it does not.

7 · Governance

7.1 Governance philosophy

NFA progressively transitions from core-team-managed operations to a DAO-governed protocol in phases. Pragmatic rationale: the early phase requires rapid parameter iteration based on real-world data; the mature phase broadens control to the token-holder base.

7.2 Governance scope

- Accuracy weighting formula and time windows
- Per-purpose LLM routing defaults
- Usage-mining and referral parameters
- Category taxonomy and weighting
- Treasury allocation decisions
- Partnership and integration decisions over a defined threshold
- Protocol upgrade approval

Core mechanics affecting user/author funds require supermajority approval (two-thirds of voting tokens). Parameter adjustments within defined ranges require simple majority.

7.3 Governance structure

Voting power is held by NFA token holders, weighted by tokens held. Proposals can be created by any token holder meeting a minimum threshold. Each proposal runs through three stages: **discussion** (7 days), **voting** (7 days), **execution** (48-hour timelock).

7.4 Transition phases

- **Phase 1.** Core team retains decision-making. DAO advisory only.
- **Phase 2.** Non-core parameters transition to binding DAO governance.
- **Phase 3.** Core mechanics progressively transition to DAO.
- **Phase 4.** Full DAO governance. Core team operates infrastructure under protocol-defined parameters.

8 · Roadmap

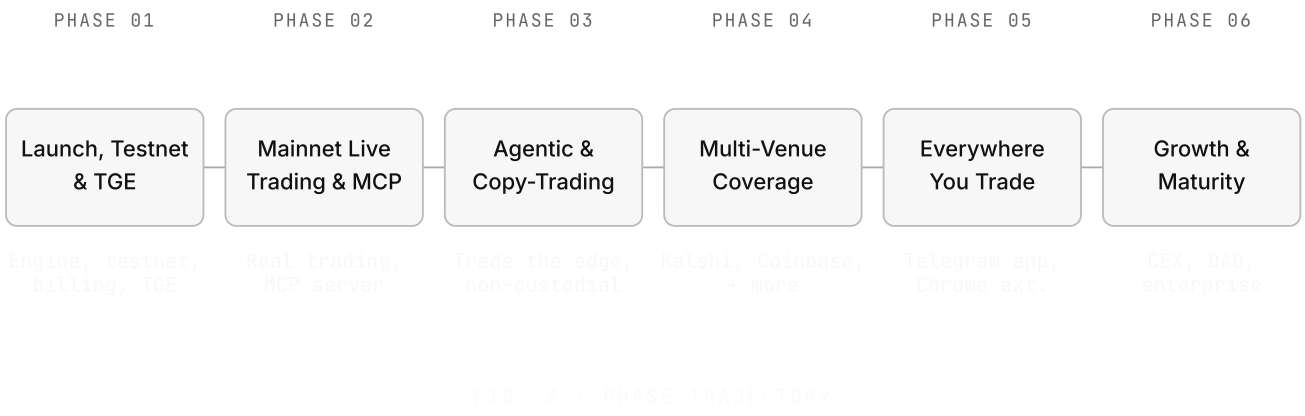


FIG. 3 · PHASE TRAJECTORY

8.1 Phase 1: Launch, testnet & TGE

TGE, liquidity, and vesting execute on Base, and the product goes live alongside them: the forecasting engine (research, advocate debate, judge) with transparent end-to-end logging and the in-app market picker, per-category accuracy tracking, metered credit billing with $NFA buy-and-burn from run revenue, and a public testnet for the trading flow. The five-month usage-mining program (§6.4) opens. Legal opinion finalized.

8.2 Phase 2: Mainnet live trading & MCP

Trading goes live on mainnet: non-custodial, real-money positions taken on top of the forecasts, with NFA never holding funds. The MCP server ships alongside it, so agents and the product alike can act on a forecast programmatically. Together these are the foundation the agentic layer builds on.

8.3 Phase 3: Agentic & copy-trading

A user, or their AI agent, can act on the engine's edge within their own limits, or copy-trade top forecasters, always non-custodially. It builds on the live mainnet trading and MCP surface from Phase 2, so the same forecast that flags a mispricing can size and place the position, with the engine's accuracy record as the signal.

8.4 Phase 4: Multi-venue coverage

Extend the engine beyond Polymarket to additional prediction-market venues, including Kalshi and Coinbase, behind the same in-app market picker. The binding layer is venue-abstracted, so each new venue is an adapter rather than a rebuild, and accuracy is tracked per venue. One engine, every market.

8.5 Phase 5: Distribution surfaces

Meet traders where they already are. A Telegram app to forecast, trade, and track positions in-chat, and a Chrome extension that overlays NFA's probability against the live market price directly on a market page. Both run on the same engine, accuracy record, and credits.

8.6 Phase 6: Growth & maturity

Tier-2 CEX listings, with Tier-1 targets contingent on volume. First DAO governance proposals, transitioning to full DAO governance per §7.4. Enterprise tier. The long-term accuracy track record becomes the primary growth asset.

9 · Risks and mitigations

This section exists because most token launches treat risk disclosure as boilerplate. Done properly, it is a competitive advantage. The risks below are the real ones.

9.1 Accuracy risk

The product thesis depends on NFA producing forecasts meaningfully better than market consensus on a durable basis.

Mitigations. The five-month usage-mining program seeds early usage and accuracy data; rigorous accuracy tracking and honest reporting including underperformance; category-specific accuracy display; continuous feedback loops; honest coverage gaps rather than forced low-quality forecasts.

9.2 Regulatory risk

Prediction-market tooling operates in complex regulatory environments. SEC, MiCA, CFTC interactions. Influencer promotion, token distribution, cross-border access introduce surface.

Mitigations. Platform operates as forecasting infrastructure not as a market; the platform produces analysis, not advice, and disclaims liability for specific forecasts; geographical restrictions at hosted-frontend level (US blocking); KYC on token claims above thresholds; legal opinion procured before TGE; advisor and KOL allocations publicly disclosed and vested.

9.3 Licensing and intellectual property risk

Mitigations. Open-source components (frontend, MCP server, settlement contracts) under Apache 2.0 / MIT; proprietary engine clearly delineated; `NOTICE` files and attribution maintained; license compliance reviewed in audit scope.

9.4 Compute economics risk

Simulation costs scale with LLM inference pricing.

Mitigations. Per-purpose routing optimizes cost-quality tradeoff per call class; prompt caching materially reduces input-token costs on supported providers; OpenRouter provides provider-agnostic routing; the platform's competitive position improves as LLM prices fall over time.

9.5 Cost tracking risk

Run pricing depends on accurate per-call cost tracking. Production experience in adjacent systems shows cost-tracking failures can underreport spend by orders of magnitude when fallback chains are hit silently.

Mitigations. Per-call cost-record persistence with provenance for how each cost was computed; recurring reconciliation against provider billing APIs with discrepancies hard-flagged for manual review before settlement; settlement gated on reconciliation confidence.

9.6 Market category coverage risk

Different categories have different accuracy characteristics.

Mitigations. Honest coverage decisions; category-specific accuracy display; the engine only forecasts categories where it has demonstrated accuracy; clear platform communication about which categories the engine suits.

10 · Closing

NFA sits at the intersection of three things that have matured in the last 18 months: prediction markets as a credible financial category, multi-agent AI systems as production-grade infrastructure, and MCP as the standardizing protocol for AI agent distribution. The product occupies a specific niche, actor-driven forecasting for Polymarket traders, that no incumbent serves.

The platform's design reflects a deliberate choice: **accuracy is the north star, everything else follows**. The engine architecture, pricing, governance transition, coverage decisions, and token economics all optimize for measurable forecasting accuracy. Opaque metrics and narrative-first positioning are rejected in favor of rigorous measurement and honest disclosure.

Today the engine forecasts every market itself, paid with credits topped up in USDC, with net run revenue flowing to $NFA buy-and-burn; broader market-category coverage and the transition to full DAO governance are the next phases. The technical engine is **production-grade from day one**: transparent end-to-end logging, per-call cost reconciliation, per-purpose LLM routing, prompt caching, and parallel debate, because forecast integrity demands it.

The risks are real and described openly. The mitigations are concrete and reflected in architecture, policies, and planned execution. The cost of doing this properly is higher than shipping a typical token launch. The cost of doing it improperly is unbounded.

NFA is multi-agent intelligence made literal as product: a panel of specialist agents arguing every outcome and an impartial judge ruling on the evidence, to produce accurate forecasts in a transparent economy. What the brand has been promising for a year, now delivered.

NFA

MULTI-AGENT INTELLIGENCE · NFA.CLUB