

Finding Memeland

The First Autonomous AI Treasure-Hunt Memecoin on Base

Litepaper v0.5 — July 5, 2026

Abstract

Finding Memeland is a memecoin powered by an autonomous AI agent that runs a recurring, on-chain treasure hunt. On each hunt, the agent dresses a brand-new, single-use X (Twitter) account with a freshly generated identity, drops progressively more obvious clues on the project's public channel, and waits. The first eligible holder who locates the hidden persona, reads the unique claim code displayed on its profile, and submits it by Direct Message — together with their wallet address — receives an automatic on-chain prize payout in \$FIND, the project's native token.

The game itself runs with no human in the loop: identity generation, clue generation and pacing, publishing, winner validation and prize disbursement are all executed by the agent. Each hunt opens with a published **integrity hash** that cryptographically commits the agent to its hidden target before the first clue drops — anyone can verify it once the hunt ends. The human operator provisions infrastructure and decides *when* hunts fire; the operator and associates are ineligible to win, never see a clue before it publishes, and cannot move the goalposts mid-game.

AI agents built by users to *help solve* clues are explicitly welcome; accounts that publicly identify themselves as bots or agents are not eligible to win. This split — agents help, humans win — keeps the game accessible to solo players while turning each hunt into a public competition over tooling.

Finding Memeland launches on Base via a **Doppler multicurve auction**: a pure fair launch with an ascending price curve, permanently locked liquidity and zero team allocation. Trading fees fund ongoing operations and prize-pool top-ups in perpetuity; a 20% treasury allocation, vesting linearly over 180 days, funds the hunts.

1. The Opportunity

The two largest narratives in crypto consumer culture today — autonomous AI agents and memecoins — have until now grown in parallel rather than together. Agent platforms (Virtuals, Clanker, Bankr) have produced billions of dollars in trading volume and tens of millions in protocol fees, while memecoins continue to be the dominant entry point for new retail capital. Yet the two have mostly remained separate categories: agents are infrastructure stories, memecoins are pure speculation.

The gap is product. Most memecoins offer holders nothing to do after they buy. Most agent tokens offer holders impressive technology but little reason to interact with the agent personally. Holders churn out within weeks because the token does not produce ongoing surface area for attention.

Finding Memeland sits in that gap. The token is the access pass to a recurring AI-run game; the agent is the engine that creates recurring reasons to hold, engage and share. **The product is the loop, not the chart.**

The timing is favorable. Clanker has processed over \$50M in cumulative protocol fees since late 2024 and an \$8M+ week in February 2026 alone. Virtuals reports \$75M+ cumulative revenue with 18,000+ agents deployed. Base remains the lowest-friction L2 for consumer crypto, with Coinbase distribution and Farcaster as a native social layer. The infrastructure for autonomous agent-driven economies is, for the first time, production-grade and at scale.

2. The Game

Core loop

Each hunt is a complete, self-contained game.

1. **Persona creation.** The agent dresses a brand-new, never-before-used X account with a fictional identity — display name, bio, avatar (image-generated), posting style — generated at launch time and applied programmatically. The account is single-use: it plays exactly one hunt and never plays again. The persona is plausible: a niche meme account, a hobbyist, an “anon” trader. It is not announced anywhere.
2. **Clue 1 — the opening post.** The agent publishes Clue 1 on the main @FindingMemeland account. This post does four things at once: it announces the hunt is live, it carries the first oblique puzzle thread, it is the **reshare gate** — to be eligible to win this hunt, your X account must have reshared this exact post (retweet or quote-tweet) — and it carries the hunt’s **integrity hash** (see *Provable integrity* below). Each hunt has its own reshare gate; a reshare of a previous hunt does not carry over.
3. **Continued clues.** The agent drops further clues on the main account at random intervals between one and three hours. The clues become **progressively more obvious** — an aggressive easing of roughly 30% per clue. The first three clues stay oblique; from clue 4 onwards, they may become structurally direct. There is **no fixed cap on clue count** — clues keep coming until someone cracks it. A typical hunt resolves in five to eight clues, somewhere between five and twenty-four hours. If a hunt somehow remains unsolved for **72 hours**, it is publicly voided: the agent announces the end, the persona keeps its secret, and the prize returns to the treasury for the next hunt.
4. **The hunt.** Eligible holders work the clues — solo, in groups, or with their own AI agents — to identify the hidden persona. The persona’s profile displays a unique **claim code** for the hunt (in its bio or pinned post). Finding the persona is the only way to obtain the code.
5. **The claim.** When a candidate has found the persona, they send a Direct Message to the **main @FindingMemeland account** containing the claim code and their on-chain wallet address. No passphrase, no riddle. The race is to be first: submissions are processed strictly in order of arrival, and the submission queue is drained completely on every read — a viral spike cannot cause an early submission to be skipped.

6. **Validation and payout.** For each incoming DM, the agent validates: (a) the claim code matches the current hunt; (b) the sender’s X account has reshared Clue 1 of the current hunt; (c) the provided wallet has held at least the minimum balance of \$FIND continuously for the minimum holding period (24 hours) — proven directly from the chain (see *Eligibility* below); (d) the sender’s account is not publicly identified as a bot or AI agent, and bot-signal screening passes. Wallet addresses are checksum-validated at submission time, so a mistyped address is caught immediately — the submitter is asked to correct it rather than discovering the error after winning. Submissions that fail a check receive a courtesy notice explaining which gate they missed. The first DM satisfying all checks triggers an on-chain transfer of the prize in \$FIND to the winner’s address — and ends the hunt at that instant; submissions arriving after the winning claim are not processed.
7. **Reveal.** Once a winner is paid, the agent publishes a Winner Announcement naming the hidden persona’s handle, the prize transaction hash — and the ingredients of the integrity hash, so anyone can verify the hunt was honest. The account is then permanently retired: it never plays again, and **its dressed profile remains publicly visible as the hunt’s artifact** — a trophy anyone can browse, screenshot and verify long after the hunt ends.

Provable integrity

Every hunt opens with a cryptographic commitment. At launch time the agent — autonomously — selects an account from the prepared pool, generates the persona identity, applies it, and generates the claim code. It then computes:

```
integrity_hash = SHA-256(persona_user_id + claim_code + secret_salt)
```

and publishes this hash inside the Clue 1 post, before any other clue exists. When the hunt ends, the Winner Announcement reveals the persona’s identity, the claim code and the salt. Anyone can recompute the hash and confirm it matches Clue 1. This proves the target was fixed before the first clue dropped and never changed mid-game.

Three further commitments back this up. **Insider ineligibility:** the operator and anyone associated with the project cannot win, ever. **Operational blindness:** the system is designed so the operator does not see the persona’s identity or any clue before it publishes — all game posts publish autonomously, with no human approval step. **Auditable submission log:** the agent logs every submission it processes, with arrival timestamps and per-filter validation outcomes; the submission count is published when each hunt closes, and a public log dashboard ships in Phase 2.

We are explicit about the limit of these guarantees: cryptography proves the target was fixed in advance; it cannot prove what no human ever saw. Closing that final gap by running the agent inside a trusted execution environment (TEE) — making “not even the founder can know” externally verifiable — is on the roadmap.

Eligibility, anti-sniper, anti-bot

Three filters apply to every submission.

Holding floor. Participation requires holding a minimum balance of \$FIND continuously for a minimum duration. The floor is a **standing, public token amount** — published at

launch and in force at all times, deliberately decoupled from hunt timing so hunts stay unpredictable: you can buy at any moment and be eligible 24 hours later. As the market cap grows, the project announces **reductions** to the floor, keeping the dollar cost of entry accessible; the floor only ever moves down, so nobody who is eligible today loses eligibility tomorrow. The initial floor and every reduction are announced on the project's public channel. Continuity is not sampled or estimated: it is **proven, exactly, from the chain**. When a claim arrives, the agent reads the wallet's current balance and replays every Transfer event that touched the wallet inside the holding window, reconstructing its balance at every point in time. If the balance dipped below the floor for even one block, the claim fails. Buying, selling and re-buying at the last minute does not qualify — the dip is in the logs forever.

Minimum holding period — 24 hours. To be eligible, your wallet must have held the floor balance continuously for **at least 24 hours** before you submit. This single, flat rule is the project's core **anti-sniper and anti-bot** filter, and it exists for one reason: to keep the game honest. You cannot watch the clues, snipe a buy at the last second, claim the prize and dump — the Transfer-event replay proves continuity to the block, so last-minute buy-claim-sell behaviour is filtered out mathematically. The same window also blunts automated farms that spin up funded wallets on demand. We deliberately keep the period **short** (24 hours, not days) so the game stays open and accessible to anyone who genuinely holds, while still making sniping and last-second automated claims impossible. The rule is fixed, public, and applied identically to everyone — including from Hunt #1. Transparency is the point: one clear number, no moving goalposts.

Public reshare. The winner's X account must have reshared (retweeted or quote-tweeted) Clue 1 of the current hunt. This converts every participant into a distribution node for that specific hunt, and it forces each candidate to expose a real, persistent X identity that has interacted publicly with the project.

Bot defences. The primary defences are structural: submissions are processed strictly in arrival order, the 24-hour proven-holding window makes disposable wallets expensive, and the reshare gate forces every candidate to expose a persistent public identity. On top of that, every submission passes the bright-line self-identification screen described below, and the operator holds a kill switch that can pause a hunt — publicly disclosed — if abuse is detected mid-game.

Humans win, agents help

Building AI agents to help you solve clues — image analysis, context scraping, candidate scoring, pattern-matching across posts — is not only allowed but encouraged. The agent-builder community is part of the project's culture, and tooling competition makes the game better over time. **What the rule protects is the identity of the winner:** the account that claims the prize must not itself be an AI agent or bot.

The project enforces this through a clean, externally-verifiable test: **accounts that publicly identify as agents or bots cannot win.** Specifically, the following are ineligible regardless of speed or holding: (a) accounts with terms like "bot", "agent", "AI" or equivalent in their display name, handle or bio — screened automatically on every submission; (b) accounts that publicly present as automated elsewhere (X's "Automated" label, registration as an autonomous agent on platforms such as Virtuals or Bankr) — enforced as policy, by review; and (c) Finding Memeland's own main and persona accounts. This rule

is objective and does not require us to judge whether a given account is “really” human. Beyond that bright-line rule, ambiguous cases fall to review: account age and organic posting history, public reshare interaction with Clue 1, and submission patterns. The operator can pause a hunt before payout where signals conflict — pauses are disclosed publicly — and the project reserves the right to disqualify a submission and award the prize to the runner-up where there is convincing evidence of fully automated submission.

The intent is plain. If you are an attentive solo human, you can win. If you are a builder who points your own private agent at the clues and then submits from your own personal account, you can win — and likely faster. If your account is publicly an agent, or you are running a covert farm of freshly-spun accounts, you will not.

Why this works

A memecoin succeeds when it produces recurring attention loops. Finding Memeland engineers them at three levels: each clue drop is shareable content, the search is collaborative on social media, and the agent-builder community treats every hunt as a public competition over tooling. The token is the cover charge to play. The per-hunt reshare gate ensures each hunt creates its own organic distribution wave. The dynamic clue cadence — no cap, progressively more obvious — keeps the game from becoming stale or unwinnable, and the 72-hour void rule guarantees every hunt has an ending worth talking about. The integrity hash gives every hunt a visible, verifiable fairness ritual. The retired personas accumulate into a public gallery of past hunts — a growing museum of the game’s history that new players can explore.

3. The AI Stack

The agent is built as a modular pipeline. The game loop — identity generation, clue generation, publishing, validation, payout — runs autonomously, with no human approval on any game action. The human operator provisions infrastructure (accounts, funds), triggers hunts, and can halt the system; the operator cannot win and does not see game content before it publishes.

Persona generator. An LLM-driven module that produces a plausible, internally-consistent identity — display name, bio, avatar (image-generated), voice — sampled at launch time from a tunable distribution of archetypes, and applied to an account from the prepared pool.

Persona lifecycle. Accounts are provisioned continuously, warmed organically, used in exactly one hunt, then permanently retired. A retired persona keeps its dressed profile as the hunt’s public artifact, but never plays again — so tracking a past persona’s account yields nothing. At higher cadence, persona accounts are planned to migrate to Farcaster, where programmatic account creation is natively supported by the protocol.

Clue scheduler. Plans a clue trajectory that resolves to the persona, distributes drops at random one-to-three-hour intervals, and applies an aggressive easing curve so each clue is roughly 30% more obvious than the last. There is no fixed cap on clue count; the scheduler keeps generating clues until the hunt is won (or the 72-hour void deadline passes). Pre-

publication guardrails verify that no clue literally contains the persona’s identity — a clue that cannot clear the guardrails is regenerated, and if necessary skipped, never published.

Content publisher. Posts to X through the official API. Farcaster publishing is planned as the community grows.

DM listener & validator. Polls inbound DMs on the main account throughout the hunt, with full pagination — every submission is read, in arrival order, even during viral spikes. For each submission, parses the claim code and wallet address (checksum-validated on the spot), then runs the eligibility checks: claim code match, on-chain continuous-holding proof via Transfer-event replay (Base RPC), public reshare of Clue 1 of the current hunt (X API), and bot-signal screening including the public-self-identification disqualification rule. Every attempt is logged with timestamps and per-filter outcomes.

Payout executor. Sends the prize from a dedicated hot wallet that holds only enough for one to two hunts at any time, with a hardcoded per-hunt cap. Payouts are **idempotent by construction**: the agent journals a payout intent before broadcasting any transaction, and at most one transfer can ever execute per hunt — a crash, restart or retry can never pay twice. The bulk of the treasury sits in a separate Safe on Base controlled by the operator; the agent has no access to it. A worst-case compromise of the agent is bounded by the hot wallet’s balance.

Reliability layer. Once a hunt is live, people are playing — so the game loop is engineered never to die mid-hunt. Every phase (DM reading, validation, clue publishing) is isolated: transient failures are retried with backoff and alerted to the operator, a submission that cannot be processed is retried without losing its place in the arrival order, and a failed clue skips one round instead of ending the game. If the process itself is restarted, the agent **resumes the live hunt exactly where it stopped** — same persona, same claim code, same reshare gate, same submission marker — from state persisted on every step. Before any hunt fires, a pre-flight check verifies the AI services, the X API, the RPC, and the money itself: gas in the hot wallet and a token balance sufficient for the prize. The operator additionally holds a kill switch that pauses the game (no clues, no processing, no payouts — DMs keep accumulating on X with arrival order preserved); pauses are disclosed publicly.

Hunts are triggered manually by the project admin via a private command channel rather than running on a fixed schedule. This is deliberate: it lets each hunt be coordinated with marketing pushes, partner announcements, or simply the rhythm of the community, instead of firing on autopilot.

4. Tokenomics

Parameter	Value
Token	\$FIND — Finding Memeland
Network	Base
Launch platform	Doppler multicurve auction (direct deployment)
Total supply	100,000,000,000

Parameter	Value
Liquidity	80,000,000,000 (80%) — sold on the multicurve; pool liquidity permanently locked
Prize Treasury allocation	20,000,000,000 (20%) — to the Treasury Safe, linear vesting over 180 days
Team allocation	0%
Trading fee	1% per swap
Fee split	95% Project Treasury / 5% Doppler Protocol
Treasury custody	Safe (multisig) on Base — signer set and threshold published at launch

Launch model

Finding Memeland launches via a **Doppler multicurve auction**, deployed directly by the project on Base. The multicurve model sells the 80% liquidity allocation along an ascending sequence of bonding curves: the price can only step upwards as demand fills each curve, there is no presale, no allowlist and no team bag, and the resulting pool liquidity is **permanently locked**. The opening valuation is set deliberately low, so the earliest believers get the fairest entry the mechanism allows.

The remaining 20% — the Prize Treasury allocation — vests **linearly over 180 days** to the project's public Safe. Nothing unlocks in a cliff; roughly 0.55% of supply becomes available per day. This is a deliberate anti-rug structure: the treasury physically cannot dump its allocation, and the vesting stream doubles as a runway discipline for prize funding. Hunt #1 fires within the first week after launch, on a date announced in the Genesis countdown — and with the flat 24-hour holding rule (see §2), anyone who bought at least a day before it fires is eligible to win it.

Fee economics

Every swap of \$FIND on the launch pool generates a 1% fee. Of that fee, **95% is routed to the Project Treasury** at a publicly-disclosed Safe address; the remaining 5% flows to the Doppler protocol. The 95% share is the project's recurring operational revenue and accrues continuously in proportion to traded volume — without any sale of treasury tokens. As long as the game generates trading volume (which is the whole point of the design), the project remains self-funding.

Treasury policy

The Treasury is held in a Safe on Base. The signer set and threshold are published at launch and upgrades to the signer policy are announced in advance.

Trading fees accrue in both currencies of the pool — ETH and *FIND*, *in proportion to swap direction*. **ETH-side fees fund the project**: *AI and infrastructure operating costs and pre-announced development*. ***FIND-side fees flow to the prize treasury***: they fund hunts, in full. Every ETH fee runs the machine; every \$FIND fee goes back to the players. The vesting 20B allocation seeds the prize treasury from launch.

During the growth phase, prizes are deliberately treated as growth spend: the first hunts are funded to grow the game and its community, not to maximize treasury retention. The Treasury will not be used for founder compensation outside pre-announced policy, undisclosed token sales, or unrelated investments.

Prizes are denominated in dollar value (provisionally \$200-500 per hunt) and converted to \$FIND at market price when each hunt fires, so prize meaningfulness does not depend on the token's price level.

All Treasury movements are visible on-chain. The Safe transaction history is the project's public ledger.

Why no team allocation

A pre-allocated team bag would undermine the project's primary credibility claim — that the agent and the community own the game, and that the founder benefits only through the same fee mechanism that benefits the protocol and the holders. We accept the trade-off: the founder has no token upside outside of treasury-funded compensation that is itself bounded by public policy.

5. Roadmap

Phase 0 — Build (May - July 2026). AI stack assembly and red-team testing across all modules — including a full independent code audit (game mechanics, security, money paths) and live end-to-end rehearsals of the complete hunt on X. Hunt Integrity Protocol implementation (commitment hash, submission log, operational blindness). Crash-resume, payout idempotency and kill-switch hardening. Brand, voice and primary channel (X; site). Legal review of the game mechanics and token issuance. Pre-launch community seeded through the Farcaster and Base agent-builder networks.

Phase 1 — Launch & First Hunts (August 2026). Deployment via Doppler multicurve on Base. Public verification of vesting parameters, treasury address and signer set. Genesis countdown, Hunt #1 within the first week. Prize size, eligibility rules and clue cadence published from day 0.

Phase 2 — Cadence and Growth (Q4 2026 - Q1 2027). Regular cadence of hunts, triggered to align with marketing pushes. Public leaderboard of winners. Public submission-log dashboard. Open APIs for community-built solver agents. Partnerships with Farcaster channels and Base ecosystem voices. Monthly treasury transparency post. Exploration of TEE deployment for verifiable agent autonomy, and migration of persona accounts to Farcaster at higher cadence.

Phase 3 — Memeland-as-a-Service (Q2 2027+). Once the game has audience and a proven format, external projects, companies and product launches can commission custom hunts featuring their brand — the agent runs the hunt, their product is the target, our community is the audience. Clients pay a service fee; a portion of every fee is used to acquire \$FIND on the open market to top up the prize treasury, so platform revenue feeds directly back into the game economy. This converts Finding Memeland from a single-token game into a marketing-as-a-game platform. The fee split and mechanics will be published before the first commissioned hunt and are subject to the legal guidance described in §6.

The exact schedule will adapt to market conditions and execution. Roadmap revisions will be published transparently.

6. Risks and Disclaimers

Platform risk. The game currently runs on X. Account suspension or API policy changes could interrupt or end hunts. The project maintains contingency procedures (published hunt-abort rules, reserve persona accounts, community channels outside X) and a planned migration path for persona accounts to Farcaster. A hunt interrupted by platform action is voided and re-run; this is disclosed in the game rules.

Agent failure risk. The autonomous agent can produce unintended outputs (off-brand clues, scheduling errors, validation edge cases). Pre-publication guardrails, the phase-isolated reliability layer, crash-resume and the human admin’s kill switch bound the damage. Pauses are disclosed publicly.

Integrity limits. The integrity hash proves each hunt’s target was fixed before the first clue; it does not, by itself, prove that no insider ever saw it. The project mitigates this with insider ineligibility, operational blindness by design, auditable submission logs — and, on the roadmap, TEE deployment for externally verifiable autonomy. Users should weigh these guarantees for themselves.

Regulatory risk. \$FIND is positioned as a utility token providing access to a skill-based game. The project is obtaining legal guidance on the game mechanics and on token issuance requirements in its operating jurisdiction (including EU MiCA), and will restrict participation where legally required. The project does not promise investment returns and is not registered as a financial product. Users are responsible for understanding the rules of their own jurisdiction.

Market risk. Token prices are volatile. Trading volume can dry up, which would reduce the fee stream that funds operations. The vesting 20% allocation provides initial runway, but is not infinite.

Custody risk. The treasury Safe and the agent’s hot wallet are described in §3 and §4; the agent can never access more than the hot wallet’s balance. We treat custody as an ongoing security exercise, not a one-time setup.

This document is not financial advice and not an offer to sell securities. It describes a product and its mechanics. Read carefully, do your own research, and risk only what you can afford to lose.

Contact

Find Finding Memeland at the channels announced on the project’s primary site at launch. Until then, follow the founder for build updates.

Litepaper v0.5 — July 5, 2026. This is a living document and may be revised before launch. Material changes will be versioned and dated.

Changelog v0.4 → v0.5

- **Launch platform: Doppler multicurve (direct deployment)** replaces “Clanker v4 via Bankr”. Bankr migrated its launcher away from Clanker, and the Doppler multicurve model won on the merits: ascending-price fair launch, permanently locked liquidity, and a 95/5 fee split in the project’s favour (vs 40/60). Supply and split unchanged: 100B, 80% liquidity / 20% prize treasury / 0% team.
- **Prize treasury vesting: linear over 180 days** replaces the 7-day lockup + instant vest. Anti-rug by construction (~0.55% of supply per day) and a built-in runway discipline. Hunt #1 timing is now set by the Genesis countdown (first week), no longer tied to a vault unlock date.
- **Anti-sniper check upgraded from sampling to exact on-chain proof.** Holding continuity is now proven by replaying the wallet’s Transfer events across the full window — exact to the block, no sampling gaps. A dip below the floor at any point disqualifies; the litepaper’s promise got stronger, not weaker.
- **72-hour void rule added.** A hunt unsolved after 72 hours is publicly voided; the persona keeps its secret and the prize returns to the treasury.
- **Retired personas keep their dressed profiles** as public artifacts of each hunt — a growing, verifiable gallery of the game’s history. (Accounts still never play twice.)
- **Reliability layer documented:** phase-isolated game loop that survives transient failures, crash-resume of live hunts from persisted state, idempotent payouts (at most one transfer per hunt, journaled before broadcast), money pre-flight (gas + prize balance checked before launch), and a public-disclosure kill switch.
- **Bot-defence and submission-log language aligned with the shipped system:** the bright-line self-identification screen is automated; label/platform-registration rules are enforced as policy by review; the hunt ends at the first valid claim (later submissions are not processed); the full submission log is retained and auditable, with the count published at hunt close and a public dashboard in Phase 2.
- **Wallet checksum validation at submission time** — a mistyped address is caught when submitted, not after winning.
- **Holding floor redefined as a standing public token amount**, decoupled from hunt timing (hunts stay unpredictable; you can buy anytime and be eligible 24h later). It only ever moves **down**: as market cap grows, reductions are announced publicly, keeping entry accessible while the cost of disposable wallets tracks the value of the prizes.
- **Ticker finalized: \$FIND.**