Dream Lottery

Turning dreams into on-chain lottery tickets

Whitepaper • Version 0.9 • September 24, 2025

Dream Lottery is the world’s first on-chain lottery where participants enter using their dreams. Each submitted dream functions as a lottery ticket, fees are paid in native $DREAM tokens, and a provably random winner takes the jackpot at the end of each cycle.

This document outlines the motivation, protocol design, tokenomics, governance, security posture, and the long-term vision for a culture-first, deflationary lottery economy.

1. Executive Summary

Dream Lottery blends transparent crypto-economic primitives with a universal human motif: dreams. Participants submit a short description of a recent or favorite dream; that entry is hashed on-chain and counts as a ticket for the current draw. At the end of each cycle (daily or weekly), an unbiased, verifiable random function selects a winner who receives the jackpot. A portion of entry fees funds the treasury and a protocol-level burn, creating long-term scarcity for $DREAM.

- A blockchain-powered lottery where dreams are tickets.
- Provably fair randomness via Chainlink VRF (with secure orchestration).
- Deflationary tokenomics through systematic burns tied to entry volume.
- DAO-driven governance and community-curated culture (galleries, contests, leaderboards).

Distribution of each cycle’s fees: 70% to jackpot, 20% to treasury (development, marketing, DAO grants), 10% burned. All flows are enforced by smart contracts and observable on-chain.

2. Problem Statement

Traditional, centralized lotteries struggle to earn trust. Opaque margins, unverifiable draws, jurisdictional friction, and limited two-way engagement reduce user confidence and long-term retention. Meanwhile, Web3 communities expect auditability, credible neutrality, and mechanisms that reward participation. They want skin-in-the-game with culture, memes, and fun—not just cold numbers.

3. Vision & Solution

Story: From Night to Chain.

You wake from a vivid dream—purple skylines, a fox in a spacesuit, a door that opens onto an ocean of stars. You jot a single line, connect your wallet, and submit. Instantly, that dream becomes a cryptographic artifact: hashed on-chain, recorded forever as a ticket to a provably fair draw. The jackpot ticks upward with each new entry. When the countdown hits zero, randomness is requested on-chain, and a winner is selected in the open. Win or not, your dream contributes to a collective gallery—fuel for memes, art, and community lore.

Mechanics.

- Submit a Dream — Pay a fixed fee in $DREAM and submit a short description (off-chain text, on-chain hash).
- Automatic Ticketing — Each entry is stored with owner address, ticket ID, dream hash, and timestamp.
- Prize Pool Growth — Fees are split: 70% jackpot, 20% treasury, 10% burn.
- Fair Draw — Chainlink VRF selects the winning ticket at cycle end; transfer executes trustlessly.

- Community Layer — Dream gallery, voting, AI art integrations, and playful leaderboards.

4. Tokenomics

Token: $DREAM (ERC-20; EVM chains; L2-first).

Utility: Lottery entry, governance voting, staking boosts/discounts, and treasury-backed rewards.

Category	Allocation
Community Rewards & Incentives	30%
Treasury & Ecosystem	25%
Team & Advisors (vesting)	15%
Liquidity & Partnerships	10%
Reserve / Future Development	10%
Early Supporters & Airdrops	10%

Total Supply: 1,000,000,000 $DREAM

Deflationary Model: 10% of entry fees are burned every cycle, tying token scarcity to protocol usage.

Illustrative Cycle Economics

Let F be the per-ticket fee in $DREAM and N the number of entries in a cycle. Then total fees = F×N. Allocation per cycle: Jackpot = 0.70(F×N), Treasury = 0.20(F×N), Burn = 0.10(F×N). Expected value per ticket (ignoring staking boosts and external incentives) is proportional to the jackpot size divided by entries.

5. Technology & Architecture

Smart Contracts

- Lottery Contract: entries, pools, VRF requests, winner selection, payouts, burns, and events.
- Token Contract: ERC-20 compliant; supports burn; compatible with staking contracts.
- DAO Governance: token-weighted proposals & votes; parameterization of fees, splits, and treasury uses.
- Optional Extensions: marketplace for dream packs, collaborative games, and seasonal quests.

Randomness & Security

Chainlink VRF provides tamper-proof randomness. The Lottery contract requests randomness at the end of each cycle, and a callback completes winner selection and payout. Testnets may temporarily use blockhash randomness during development, but mainnet requires VRF or a similarly secure oracle. A third-party audit precedes mainnet launch.

Storage

- On-chain: ticket ID, owner address, dream hash (keccak256 of salted text), entry timestamp.
- Off-chain: dream text and optional AI-generated image stored on IPFS/Arweave; CID referenced on-chain.

Reference Pseudocode

```
function submitDream(bytes32 dreamHash) external { require(token.allowance(msg.sender, address(this)) >= fee, "Approve $DREAM"); token.transferFrom(msg.sender, address(this), fee); // splits uint256 jackpotDelta = (fee * 70) / 100; uint256 treasuryDelta = (fee * 20) / 100; uint256 burnDelta = fee - jackpotDelta - treasuryDelta; jackpot += jackpotDelta; treasury += treasuryDelta; token.burn(burnDelta); // record entry entries.push(Entry({owner: msg.sender, hash: dreamHash, ts: block.timestamp})); emit DreamSubmitted(msg.sender, dreamHash); } function finalizeCycle() external onlyKeeper { require(block.timestamp >= cycleEnd, "Not yet"); requestId = VRF.requestRandomWords(...); state = WAITING_FOR_VRF; } function fulfillRandomWords(uint256 requestId, uint256[] memory words) internal override { uint256 winningIndex = words[0] % entries.length; address winner = entries[winningIndex].owner; token.transfer(winner, jackpot); emit WinnerSelected(winner, jackpot); jackpot = 0; _startNextCycle(); }
```

6. User Flow (MVP)

- Connect wallet and approve $DREAM.
- Submit dream text; client computes salted keccak256 hash; contract logs entry.
- Jackpot grows with each entry.
- At cycle end, VRF selects winner; 70% jackpot paid automatically.

7. Roadmap

Phase 1 — Development

- Token creation ($DREAM).
- MVP lottery contract & off-chain submission pipeline.
- Simple front-end for submissions and jackpot view.

Phase 2 — Beta Launch

- Deploy to an L2 for low fees; integrate Chainlink VRF.
- Public testnet draws & security reviews.
- Initial marketing and community seeding.

Phase 3 — Mainnet Launch

- Full audits, launch lottery + burn mechanics.
- Introduce DAO governance (Governor-style).
- DEX listings for $DREAM liquidity.

Phase 4 — Expansion

- AI Dream Visualizer and curated gallery.
- Dream Battles, meme contests, seasonal quests.
- Partnerships with NFT/game communities; multi-chain support.

8. Governance Model

$DREAM holders govern the protocol. Voting power is proportional to token holdings (with potential staking multipliers). Governance can tune fee amounts, distribution ratios, cycle cadence, treasury allocations, and feature rollouts. Over time, curation of community content and the introduction of gamified mechanics also move on-chain via proposals.

9. Security & Compliance

Audits: Third-party audits precede the mainnet launch; a permanent bug bounty runs thereafter.

Transparency: All draw events, transfers, and burns are logged and queryable on-chain.

Compliance:

- Positioned as a tokenized game of chance and cultural experience; no fiat handling.
- Regional regulatory analysis informs market rollouts; geo-controls may apply where necessary.
- Self-exclusion tools and age gates are provided in the interface.

10. Risks & Mitigations

Risk	Mitigation
Regulatory Uncertainty	Clear positioning as entertainment, gradual rollouts, legal consulta
Smart Contract Bugs	Audits, formal reviews, staged releases, canary deployments, bug
Low Adoption	Lean into culture: memes, contests, airdrops, and social integratio
Sybil/Abuse	Rate limits, streak caps, optional checks, and governance-driven p
Oracle/Infra Dependencies	Multiple keepers, alerting, circuit-breakers, and fail-safe settlemer

11. Community & Culture

Dream Lottery is a social experiment. Beyond jackpots, it forges a living archive of our subconscious—curated, remixed, and celebrated. Weekly highlights feature the funniest and wildest dreams. Founding Dreamers earn recognition and unique perks. Culture compounds as stories, art, and lore feed back into the protocol, driving flywheel effects.

12. Token Utility Deep Dive

- Entry Fees: $DREAM is the native currency for tickets; fees route automatically to jackpot/treasury/burn.

- Governance: voting weight may be augmented by lockups/staking to reward conviction.
- Staking Boosts: stakers may receive fee discounts or additional non-monetary perks (badges, ranking).
- Treasury Rewards: community grants for builders, artists, curators, and evangelists.

13. Economic Considerations

The protocol ties scarcity to usage. Burns scale with participation; if daily entries rise, deflation accelerates. Treasury captures resources to fund growth and resilience. Parameters are tunable by governance to maintain balance: for instance, lowering the entry fee during growth campaigns or increasing burns in mature phases.

Example Sensitivities

- Higher participation → larger jackpots → stronger viral pull.
- Lower fees on L2s → more entries → more frequent community moments.
- Aggressive burns → faster scarcity but reduced treasury runway; governance calibrates trade-offs.

14. Implementation Notes

- Use OpenZeppelin libraries for ERC-20, access control, and Governor modules.
- Follow checks-effects-interactions and pull payment patterns; minimize external calls.
- Emit rich events for every state change; index by cycle, winner, and participant.
- Front-end: wallet connectors, gas estimation, cycle countdown timers, and real-time jackpot updates.
- Monitoring: VRF request/fulfillment dashboards, on-chain alerts, error budgets.

15. Legal & Ethical Considerations

This whitepaper is for informational purposes only and does not constitute investment advice or a solicitation. Participation may be restricted in certain jurisdictions. Users are responsible for complying with applicable laws. The protocol promotes responsible play—time/amount reminders, self-exclusion, and transparent odds.

16. Conclusion

Dream Lottery fuses verifiable fairness with collective storytelling. By turning dreams into tickets, it transforms a universal, deeply human experience into a transparent, community-owned game. With deflationary tokenomics, DAO governance, and an expanding cultural layer, Dream Lottery aims to become the most delightful, credibly neutral lottery in Web3—where imagination itself is part of the prize.

Contact & Next Steps: publish MVP contracts to testnet, invite public testing, commission audits, and schedule a phased mainnet launch.