LEMON JET

WHITEPAPER

Lemon Jet Whitepaper: the game, LJT and on-chain liquidity

Lemon Jet's whitepaper explains its crash-style game on Base, LJT token, Chainlink VRF settlement, vault economics, referrals and development direction.

Version 1.0

5 September 2026

lemonjet.xyz

Contents

A guide to the protocol, the LJT token and the records you can verify.

Executive summary

Lemon Jet is an on-chain, crash-style multiplier game on Base. A player chooses an LJT stake and a target multiplier before a round begins. The game requests verifiable randomness from Chainlink VRF, applies its published settlement rules and pays a successful prediction from an LJT vault.

The product connects three forms of participation. Players use LJT for predictions. Liquidity providers deposit LJT and receive shares in the pool that backs payouts. Referrers introduce players and receive vault shares calculated from eligible bets. These roles share an economic system while carrying different exposures.

LJT's utility is present in a working product. Its longer-term opportunity depends on sustained player interest, sufficient liquidity, understandable onboarding and delivered development. Token use is not a guarantee of a future market price or a positive return.

This whitepaper describes the currently configured Base token and game contracts. It provides their economic model, operational assumptions and limits. Published roadmap items are identified as plans; allocation, vesting and token-sale terms require their own definitive disclosures.

1. Product thesis

Multiplier games make a simple decision engaging: how much to stake and what potential payout to target. Lemon Jet pairs that experience with a settlement process that can be inspected independently of the game website.

The product's proposition rests on three practical properties:

- **A decision made before the outcome.** The stake and target are committed in the game transaction.
- **Verifiable randomness.** A Chainlink VRF response supplies the value used by the contract to determine the outcome.
- **Inspectable token movements.** Bets, winning payouts and vault positions are represented on Base.

Lemon Jet's current format fixes the target before the flight. It does not provide an interactive cash-out button during an already submitted round. The animated experience presents the outcome; it does not change the contract's decision.

The intended experience is straightforward enough for a player to understand and transparent enough for an independent reader to examine. Its economic design also makes the source of liquidity and the incentives paid from game activity visible.

2. System architecture

Component	Function
Base	Executes transactions and records the deployed token, game and vault state.
LJT contract	Provides the ERC-20 asset used for stakes, payouts and vault deposits.
Game and vault contract	Accepts predictions, reserves payout capacity, settles results and manages vault shares.
Chainlink VRF wrapper	Handles the direct-funded randomness request and authorized callback.
Player wallet	Authorizes token spending and transactions; holds tokens and shares outside active contract positions.
Website and supporting services	Present balances and history, construct transactions and manage invitation attribution.

The game and vault share one deployed contract. This connects the pool's assets directly to accepted game obligations.

Non-custodial interaction means users authorize transactions with their own wallets. It does not mean assets remain in the wallet after use: an accepted bet transfers LJT to the game contract, and a vault deposit transfers LJT into the pool. Funds committed to those operations are exposed to the corresponding contract rules.

The website, indexing services and referral attribution infrastructure are separate from on-chain settlement. An interface delay can leave a balance display behind the chain's current state. Off-chain services remain relevant to availability and usability even though they do not generate the game's winning random value.

3. Player journey

1. Connect a supported wallet and select Base.
2. Hold sufficient LJT for the stake and ETH on Base for transaction gas and the randomness request.
3. Select a stake and target multiplier.
4. Grant an LJT allowance if the game contract needs spending permission.
5. Review and submit the game transaction.
6. Wait for the randomness callback and settlement.
7. Inspect the result and any payout through the app or BaseScan.

The current game interface uses a minimum stake of 0.1 LJT and a target range of 1.1x to 5,000x. The underlying contract accepts a wider lower range: its minimum target is 1.01x, and its minimum stake is 1,000 of the token's smallest units. With 18 decimal places, that contract minimum is 0.000000000000001 LJT. The interface intentionally presents different practical limits.

The contract permits one unresolved game per player wallet. Other wallets can have their own requests in progress. A confirmed opening transaction does not itself mean the random outcome has settled.

Token approvals are separate from bets. The current main game interface requests a very large allowance when permission is insufficient. Readers should inspect the spender and amount shown by their wallet; connecting to the website alone is not that spending permission.

4. Game mathematics

4.1 Stake, multiplier and payout

Let `b` be the stake measured in the smallest units of LJT. Let `c` be the integer coefficient stored by the contract; the displayed multiplier is `m = c / 100`.

```
101 <= c <= 500000
winning payout = floor(b * c / 100)
potential net winnings = winning payout - b
```

The winning payout includes the stake. A 10 LJT prediction at 2x therefore returns 20 LJT on a win and zero on a loss. Native-token expenses are separate from this LJT calculation.

4.2 Random roll and winning threshold

The contract reduces the supplied random word to a discrete roll and compares it with the threshold for the committed coefficient:

```
R = 100000000
threshold = floor(R * 9900 * 100 / (10000 * c))
roll = (randomWord mod R) + 1
win when roll <= threshold
```

On the uniform discrete-roll model, the probability of winning is `threshold / R`, approximately `0.99 / m`.

Target	Winning probability on that model	Total payout for a winning 10 LJT stake
2x	49.5%	20 LJT
10x	9.9%	100 LJT
100x	0.99%	1,000 LJT
5,000x	0.0198%	50,000 LJT

The payout column illustrates the arithmetic. It does not establish that a 10 LJT stake at every listed target can be accepted by the current pool; the available risk capacity must also permit the bet.

Reducing a uniform 256-bit word modulo `R` introduces a negligible discrete bias. For readers reproducing the exact calculation, let `N = 2^256`, `q = floor(N / R)`, `r = N mod R` and `t = threshold`. Then the probability under a uniform 256-bit word is `(t * q + min(t, r)) / N`. This expression is a mathematical consequence of the modulo operation.

4.3 House edge and return to player

The multiplier schedule uses a nominal 1% house-edge parameter, corresponding to approximately 99% theoretical gross return to player. Integer rounding of thresholds and payouts affects the exact result for a given input.

RTP is an expectation under the probability model across many outcomes. It does not describe a guaranteed refund, the result of one session or a limit on a player's losses. Gas and VRF costs are outside the gross RTP calculation.

Each round uses its own randomness request. A losing streak does not create a contractual entitlement to a win, and selecting a different stake does not improve the next random value.

5. Randomness, funding and settlement

Lemon Jet uses Chainlink VRF v2.5 direct funding with payment in native ETH. The game asks the configured wrapper for the current request price and requires the submitted ETH value to cover it. Base gas is an additional transaction cost.

The inspected configuration requests one random word, uses a callback gas limit of 100,000 and sets request confirmations to zero on its configured Base wrapper. That setting does not imply that reorganizations or delivery failures are impossible. Oracle configuration is part of the system's risk model.

Only the configured wrapper may deliver the accepted callback. The contract matches the response to the pending game, releases its reserved exposure and transfers the winning payout when applicable.

The game attempts to return ETH submitted above the request cost to the player. If that refund call fails, the excess remains in the contract. Residual native ETH can subsequently be sent to the fixed reserve fund by the public `claimNativeBalance` function; calling that function does not pay the caller or create a player-specific refund claim.

Settlement depends on network execution and a successful oracle callback. The verified implementation has no player-accessible function to cancel, manually settle or re-request a stalled round. A pending state should not be presented as a promise of an automatic refund after a fixed interval.

The Chainlink direct-funding documentation and VRF security guidance describe the external mechanism and its operating considerations.

6. Liquidity vault and payout capacity

6.1 Deposits and shares

Depositing LJT into the vault issues ERC-4626-style shares. The LJT value of those shares depends on the pool's assets and shares outstanding. A deposit adds assets and issues shares; it is not a fixed-interest loan or an automatic source of profit.

The conversion uses the implementation's rounding and virtual-unit rules, so an independent calculation should use the contract's conversion methods. A simple assets-divided-by-shares ratio is an intuition, not a replacement for those methods.

The current interface labels the vault shares eLJT, while the deployed share contract reports the symbol lpLJT. Both refer to the configured game vault; the underlying betting token is LJT. Contract identity matters more than the display label when inspecting a wallet or explorer.

6.2 Where returns come from

Bets enter the pool. Winning payouts leave it, while losing stakes remain. The results change the assets backing the shares. Referral and reserve share issuance also changes each holder's proportional ownership.

The house edge describes expected gaming margin, not the depositor's APY. Actual returns depend on game outcomes, volume, capital, share allocations and the period held. The pool can lose LJT, and a depositor can receive less than the amount originally contributed.

6.3 Risk capacity

The contract tracks the assets in the vault, full pending payouts and pending net winnings. Let `A`, `P` and `W` represent those values respectively.

```
settledBankroll = A + W - P
riskCap = floor(settledBankroll * 50 / 10000)
maxWinAmount = max(riskCap - W, 0)
```

A new game is accepted only if its potential net winnings fit within `maxWinAmount`. The 0.5% parameter therefore constrains aggregate unresolved net winnings relative to settled bankroll. It is not a rule that a stake or a full winning payout can always equal 0.5% of the visible pool balance.

The bankroll calculation excludes stakes belonging to unresolved games. As games settle and the vault changes, new capacity becomes available or existing capacity contracts.

6.4 Available withdrawals

```
withdrawableAssets = max(A - P, 0)
```

Full potential payouts are reserved while games remain unresolved. Those obligations constrain withdrawals and redemptions. Ownership of shares does not guarantee that all of their underlying assets can be withdrawn at once.

7. Fees and economic allocations

7.1 Allocations when a bet is accepted

The current game issues reserve-fund shares using 0.2% of each accepted stake as the calculation basis. If the player has a recorded referrer, it also issues referral shares using 0.3% of the stake as the basis.

These operations mint vault shares using the conversion rate at issuance; they do not directly transfer the corresponding LJT out of the pool. New shares affect the proportional ownership of existing holders.

Mechanism	Current parameter	Form
Nominal game edge	1%	Embedded in the probability schedule.

Mechanism	Current parameter	Form
Referral allocation	0.3% of a referred bet	Newly issued vault shares.
Reserve allocation on play	0.2% of a bet	Newly issued vault shares.
Exit-fee parameter	200 basis points	Applied through the withdrawal or redemption formula.

The nominal edge is therefore not paid in full to existing liquidity providers. It is also not a fixed deduction taken from every round's LJT stake: actual pool results depend on the outcomes that occur.

7.2 Withdrawing assets and redeeming shares

The two exit methods specify different quantities. `withdraw` requests an LJT amount to receive, while `redeem` supplies the number of shares to consume. The current fee implementation rounds the fee down in token units.

```
withdraw, with requested net assets a:
  fee = floor(a * 200 / 10000)
  burn shares covering a + fee, with conversion rounded up

redeem, with gross assets g represented by the shares:
  fee = floor(g * 200 / 10200)
  assets received = g - fee
```

As a result, the 2% fee parameter should not be read as a universal multiplication of every displayed gross amount by 0.98. The operation's inputs, conversion and rounding determine the outcome.

On exit, the implementation also issues reserve shares using a 0.1% factor on the derived pre-exit-fee share quantity. The rest of the retained value remains in the pool and affects remaining holders. This is not a transfer of the entire exit fee to an operator's wallet.

8. Referral program

A referrer creates an invitation through the application using a connected wallet and, where required, a wallet sign-in signature. Creating the link does not require a bet, vault deposit or on-chain gas payment.

The invitation flow preserves first-touch attribution for 30 days before confirmation. Opening a link is distinct from an on-chain referral: the relationship is recorded when an eligible game transaction successfully establishes the referrer for the player wallet.

Once recorded, the current contract preserves that relationship without a built-in expiry or normal reassignment function. Self-referral by the same wallet is rejected.

For a referred bet of 1,000 LJT, the reward calculation basis is 3 LJT. The contract issues the corresponding shares when the bet is accepted, before the random outcome is known. Winning and losing referred bets use the same reward calculation.

The referrer receives a vault position, not an immediate transfer of freely spendable LJT. Redeeming that position follows the pool's conversion, exit-fee and available-liquidity rules. Its value may change after issuance.

The application reports link activity separately from contract balances. Link opens and active invitation claims must not be interpreted as the number of paying players or the value of rewards earned.

The referral program guide provides the step-by-step user flow.

9. LJT token economics

9.1 Current deployed token

Property	Current implementation
Network	Base mainnet.
Token standard	ERC-20 with ERC-20 Permit support.
Symbol	LJT.
Decimal places	18.
Initial issuance	1,000,000,000 LJT, minted at construction.
Additional issuance function	None exposed in the verified implementation.
Core uses	Game stakes, winning payouts and deposits backing the vault.

The current game contract's underlying asset is this deployed LJT token. Its observed total supply matches the initial issuance at the time of this edition. Total supply does not establish circulating supply, holder distribution, liquid market float or the absence of concentration.

The current token implementation does not expose an owner mint, blacklist or pause function. This statement concerns the inspected deployed token; it does not imply that every external market, service or future token deployment shares the same properties.

9.2 Utility and value

The game creates a use for LJT when someone places a prediction or provides liquidity. That use is observable through activity and contract balances.

Market value also depends on distribution, selling, liquidity and demand outside the game. The same tokens can circulate through repeated bets, payouts and transfers, so betting volume must not be treated as an equal amount of new token purchases.

Vault returns are a third measure. They describe the changing claim on pool assets, not a guarantee about LJT's exchange price. A vault position can gain in token units while losing in market value, or vice versa.

9.3 Allocation, vesting and sales

This edition establishes the current deployed issuance and utility. It does not designate an approved team, treasury, liquidity or community allocation, or certify the custody and lockup of any allocation. Those claims require an allocation schedule and supporting contracts or disclosures.

The project's public launchpad materials describe a token-sale programme. At the time of this edition, the sale page does not identify a definitive token contract. Readers must therefore not infer that a sale entitlement automatically maps to the current game token, or that an exchange or migration ratio has been established by this whitepaper.

Sale prices, vesting, eligibility, refunds and listing arrangements are governed by the definitive documents for that sale. This whitepaper makes no offer and adds no sale terms.

10. Adoption and development direction

The existing foundation consists of the Base multiplier game, the LJT vault, wallet interaction, transaction history and the referral flow. This gives the project a product whose use and economics can be evaluated today.

The published roadmap describes the following development direction:

Published period	Direction	Status in this edition
Q3 2026	Mainnet and token-launch activity, launchpad and listing work, marketing and Base DeFi integrations.	The game is deployed. Token events, listings and additional integrations require their own launch confirmation.
Q4 2026	Additional game modes, including named Box Game and Options Game concepts.	Planned direction; this edition does not establish their availability or final rules.

These entries report published intentions rather than introducing new delivery commitments. Releases may change in scope or timing. A future feature becomes part of LJT's demonstrated utility when it is delivered with clear rules and actual availability.

Useful measures of progress include sustained settled-game activity, returning players, successful onboarding, vault capacity and reliable settlement. Each should be measured over time and interpreted in context; incentivized traffic alone does not establish lasting demand.

The opportunity for Lemon Jet is to make an inspectable on-chain game easy to understand and use. Product development, education and community distribution support that aim when they improve the experience people actually receive.

11. Control, dependencies and verification

The verified game implementation exposes no owner role, pause function, parameter setters or implementation-upgrade entry point. The reserve fund and VRF wrapper are constructor-configured and have no exposed setters. It does not provide an administrator with a function to change an individual player's odds or replace their recorded referrer.

These are properties of this deployment, not a claim that the entire application stack is immutable. Website releases, off-chain services, external infrastructure and any future contracts must be considered separately. A future interface update does not rewrite the source of this deployed game contract.

Contract source verification lets readers inspect the code associated with a deployment. It is not an independent security audit or a guarantee that the code has no defects. This edition does not claim independent audit assurance for the current contracts.

Readers can verify the contract identities from the official application, inspect the sources and check public read methods such as `asset`, `totalSupply`, `totalAssets`, `maxWinAmount`, `maxWithdraw` and `getReferrer` where exposed by the relevant contract. These answer different questions; a token's supply and the vault's assets are not interchangeable.

12. Material risks and participation

Game outcomes. A prediction can lose its full stake. The edge and probability model do not guarantee a session result or recovery after losses.

Vault exposure. Player wins, share allocations and fees affect pool ownership and redemption value. Pending obligations can restrict liquidity.

Contract execution. Publicly readable code can still contain defects. Approvals grant spending authority to their named spender; users should understand the permission they authorize.

Oracle and network operation. Congestion, delivery failure, callback failure or reorganization can affect execution and settlement. The current game does not expose a general cancellation or rescue path for a stalled round.

Token markets and distribution. LJT's price can fall, market liquidity can be limited, and holder concentration or future unlocks may affect available supply. Utility does not remove these risks.

External services. Wallets, swaps, bridges and other third-party services have separate functionality and risks. A source-chain transaction does not guarantee completion of a cross-chain transfer.

Eligibility. Use is subject to the Terms of Service, including age and location requirements, and the Privacy Policy. The game is intended for eligible adults as entertainment. Participants remain responsible for their own decisions and applicable obligations.

13. Primary references and further reading

Contract figures in this edition describe the verified implementations configured by the application at the publication date. A change of deployment requires a fresh assessment.

- LJT token: verified source and public state.
- Game and vault: verified source and public state.
- Base transaction fee documentation.
- Chainlink VRF direct funding.
- Chainlink VRF supported networks.
- Chainlink VRF security guidance.
- ERC-20 token standard.
- ERC-4626 tokenized vault standard.
- Published development roadmap.
- Public launchpad information and its separate sale terms.

For a shorter introduction, read how the crash-style game works, what LJT is used for and the factors behind LJT's potential.