Triffle — Decentralized Raffle Marketplace on Base

Whitepaper v1.1.0 · 2025-10-21

Abstract

Triffle is a fully on-chain raffle marketplace on Base. Creators list assets from their own wallets, define ticket supply, price, and a countdown, and the protocol settles outcomes programmatically using Chainlink VRF for verifiable randomness and OnChainWin smart-contract modules. When the draw condition is met, a winner is selected on-chain and the prize is transferred automatically to the winner’s address.

Table of Contents

Executive Summary

Problem. Traditional digital raffles are opaque: an operator holds assets off-chain and claims to run a fair draw. Rules, randomness, and settlement are difficult to audit.

Solution. Triffle encodes raffle rules in smart contracts, escrows assets on-chain, and requests verifiable randomness from Chainlink VRF (Verifiable Random Function). The contract verifies the VRF proof, computes a winner deterministically, and transfers the prize automatically.

Why Base. Base provides low fees, EVM equivalence, and Ethereum-aligned security—ideal for frequent ticket purchases and verifiable on-chain settlements.

Key Differentiators.

• Fully on-chain listing, ticketing, and settlement.

• Verifiable randomness via Chainlink VRF with on-chain proof validation.

• Clear marketplace rules and fallbacks (20-minute cancel window; minimum 1 ticket to run).

• Modular contract set (NFT raffles, token/ETH raffles, quest/community flows) using OnChainWin technology.

TL;DR.

• List an asset from your Base wallet, set rules, and start.

• Participants buy any number of tickets; equal odds per ticket.

• At countdown end or sell-out, the contract requests VRF and pays the winner on-chain.

• Early cancel within 20 minutes refunds everyone; zero-sale refunds the owner.

Background & Rationale

Centralized raffles require trust in a third party for asset custody, draw integrity, and payouts. Triffle replaces trust with verification by keeping core logic and funds on-chain.

Benefits include self-custody listing from a wallet, public on-chain auditability of ticket purchases and randomness proofs, and programmatic settlement without human intervention.

System Overview

Roles

• **Owner (seller):** Proves asset ownership with a Base wallet; defines ticket count, price, and a countdown; initiates the raffle.

• **Participant (buyer):** Purchases one or more tickets; each ticket has equal winning odds.

• **Protocol/Contracts:** Escrow the prize, sell and account for tickets, orchestrate the draw, and settle payouts.

• **VRF Coordinator:** Chainlink service that returns randomness with a cryptographic proof verified on-chain.

• **Automation:** Chainlink Automation monitors conditions and triggers the draw.

High-Level Flow

```
flowchart LR
  U[User Wallet] -- list asset & params --> T{Triffle Contracts}
  T -- escrow & ticketing --> S[(On-chain State)]
  P[Participants] -- buy tickets --> T
  A[Chainlink Automation] -- check end or sell-out --> T
  T -- requestRandomWords --> V[Chainlink VRF]
  V -- random + proof --> T
  T -- compute winner --> S
  T -- transfer prize --> W[Winner Address]
```

Alt: User → Triffle Contract → Chainlink VRF → On-chain Winner & Payout.

Protocol Architecture (Detailed)

Contracts & Responsibilities

Triffle's contract set comprises five modules: Triffle (NFT raffles), TriffleERC20 (Token/ETH raffles), TriffleQuestNFT, TriffleQuest, and TriffleCommunityManager. These modules share security controls (ReentrancyGuard, CEI ordering, verified asset/token whitelists) and integrate Chainlink VRF and Automation.

Representative Data Structures & Events

Core storage includes raffle structs (creator, prize, timebox, fees, maximum and total entries, VRF status), verified token and collection lists, community registries, and accounting of creator balances. Key events include RaffleCreated, TicketPurchased, DrawRequested, RandomnessFulfilled, RaffleCanceled, and RefundProcessed.

Owner Listing & Escrow

```
sequenceDiagram
  participant Owner
  participant App
  participant Triffle
  Owner->>App: Connect Base wallet, select asset (NFT/ERC20)
  App->>Triffle: createRaffle(asset, idOrAmount, duration, entryFee, maxEntries)
  Triffle-->>Triffle: validate verified asset; compute commission; escrow asset
  Triffle-->>Owner: RaffleCreated(raffleId)
```

Caption: Owner lists an asset (NFT/ERC20-ETH/Token); contract validates, escrows the asset, and emits RaffleCreated.

Ticket Purchase & Accounting

```
sequenceDiagram
  participant Buyer
  participant Triffle
  Buyer->>Triffle: buyEntries(raffleId, n) with value = n * entryFee
  Triffle-->>Triffle: CEI update totals; credit creatorBalance; record buyer entries
  Triffle-->>Buyer: TicketPurchased(...)
```

Caption: Buyer sends ETH; contract updates counts before external calls, mitigating reentrancy.

Draw & VRF Callback

```
sequenceDiagram
  participant Automation
  participant Triffle
  participant VRF as Chainlink VRF
  Automation->>Triffle: performUpkeep() (end reached & ≥1 ticket)
  Triffle->>VRF: requestRandomWords(keyHash, subscr, conf)
  VRF-->>Triffle: fulfillRandomWords(requestId, randomWords, proof)
  Triffle-->>Triffle: verify proof; winnerIndex = randomWords[0] % totalEntries
  Triffle-->>Winner: transfer prize (NFT/ERC20-ETH/Token)
  Triffle-->>Creator: creatorBalance available for claim
```

Caption: Automation triggers VRF; contract verifies proof, selects winner, pays prize (NFT/ERC20-ETH/Token), and credits creator revenue.

Refunds & Cancellations (20-minute window)

```
sequenceDiagram
  participant Owner
  participant Triffle
  Owner->>Triffle: cancelRaffle(raffleId) within 20 minutes
  Triffle-->>Owner: return asset (NFT/ERC20-ETH/Token)
  Triffle-->>Participants: batch refunds (processRefunds)
```

Caption: Within 20 minutes, owner may cancel; the escrowed asset (NFT/ERC20-ETH/Token) returns and ticket funds are refunded in batches.

Raffle State Machine

```
stateDiagram-v2
  [*] --> Created
  Created --> Active: escrow complete
  Active --> Locked: t>=start+duration OR totalEntries==max
  Locked --> DrawRequested: Automation.performUpkeep()
  DrawRequested --> Awarded: VRF fulfill & winner computed
  Awarded --> Settled: prize transfer + creator balance ready
  Active --> Canceled: cancel within 20 min
  Active --> Refunded: countdown ends AND totalEntries==0
  Awarded --> Refunded: failure -> batch refunds (admin/emergency path)
```

Caption: Lifecycle from creation to settlement; early-cancel and zero-sale paths are explicit.

Randomness, Draw, and Settlement

Automation checks if the countdown elapsed, at least one ticket exists, and no VRF request was sent. If true, it calls performUpkeep, which requests randomness and marks the raffle accordingly.

On fulfillRandomWords, the contract verifies the callback source, calculates winnerIndex = randomWords[0] % totalEntries, selects the winning ticket’s owner, and transfers the prize via safe transfer (NFT) or safe value/erc20 transfer (ETH/ERC-20).

Edge cases: When zero tickets sell by the end, the asset returns to the owner and no draw occurs. Draws can proceed with any positive number of tickets. If a VRF response is delayed or fails, a guarded retry function is available.

Marketplace Rules & Fallbacks

Owner flow (checklist)

• Connect a Base wallet and prove asset ownership.

• Set ticket count, ticket price, and a countdown; confirm creation.

• Share the raffle link and monitor ticket sales.

• Optional: cancel within the first 20 minutes. After 20 minutes, the asset is locked until the countdown ends.

Participant flow (checklist)

• Connect a Base wallet; open the raffle page.

• Review terms; buy one or more tickets. Each ticket has equal odds. A single buyer may buy the full supply.

• Wait for the result; verify the outcome on BaseScan (VRF proof and winner transaction).

Hard rules

• Minimum to run: at least 1 ticket must be sold, otherwise the asset is refunded to the owner.

• Early cancel: permitted only within the first 20 minutes; refunds are automatic.

• Equal odds per ticket: enforced by uniform modulo selection on total entries.

Triffle OG NFTs & Community Raffles

OG NFTs introduce an access layer for community-led raffles on Triffle.

What OG holders can do

• Open a Community by escrowing the OG NFT; this anchors the Community to the OG’s address.

• Launch special community raffles that can be free or paid.

• (Optional) Add simple task gating to improve discovery and fairness.

Supported tasks

• Visit website

• Join Discord server

• Follow X account

How task-gated raffles work

• Tasks are verified in-app. Raffle rules, draws, and payouts are enforced on-chain.

• After verification, entrants can join free raffles or buy tickets for paid raffles. Each ticket has equal odds.

Regular users

• Can initiate paid raffles but cannot specify tasks.

Settlement

• At the end, the contract requests Chainlink VRF randomness with proof, selects a winner on-chain, and transfers the prize automatically.

Security Model

Principles & Controls

• Reentrancy protection via ReentrancyGuard and Checks-Effects-Interactions (CEI) ordering.

• Verified whitelists for NFTs and ERC-20s; ETH treated as a pre-verified prize type.

• Only the VRF Coordinator can call fulfillRandomWords; request IDs are tracked.

• Emergency functions (withdraw/unstick flows) exist for recovery and are restricted and logged.

• Batch refunds to avoid out-of-gas failures.

Threat Model

Attack Vector	Impact	Mitigation
Reentrancy during purchase/claim	Drain balances or double-spend entries	Apply nonReentrant and CEI; update state before interactions and isolate accounting per raffle.
Malicious token/NFT prize	Rug or transfer failure	Use verified NFT/Token whitelists; treat ETH as pre-verified; validate safe transfers and revert on failure.
VRF spoofing	Biased or fake randomness	Restrict fulfillment to the VRF Coordinator; verify request IDs and proofs; ignore unexpected callbacks.
Mass refunds gas griefing	Refund loop fails	Process refunds in bounded batches with per-item try/catch and progress tracking to avoid out-of-gas.
Admin abuse	Centralized control risk	No admin method to set winners; cancel window limited to 20 minutes; all admin actions emit events and are auditable.

Economics & Fees

The UI surfaces the entry fee and the platform commission before confirmation. Participants pay entryFee + commission; the creator receives the base entry fee times tickets sold; the platform collects the commission.

Parameter	Value
Base entry fee (creator)	0.01 ETH
Commission (5%)	0.0005 ETH
Ticket price shown to buyer	0.0105 ETH
Tickets sold	500
Creator revenue	500 × 0.01 = 5.00 ETH
Platform fee	500 × 0.0005 = 0.25 ETH

Why Base

Base is an OP Stack Layer-2 with low fees, high throughput, and EVM equivalence. It inherits Ethereum’s security assumptions while enabling fast, affordable transactions—ideal for frequent ticket purchases and on-chain draws.

Compliance & Responsible Use

Triffle provides on-chain tooling to run transparent raffles. Regulations vary by jurisdiction. Creators and participants are responsible for ensuring their raffles comply with local laws and platform policies. Nothing in this document constitutes legal, tax, or financial advice. If unsure, seek qualified counsel. Contact: contact@triffle.xyz.

Roadmap (2025–2027)

2025

• v1 mainnet launch on Base with NFT and ERC-20/ETH raffles.

• Public docs and explorer links; transparency dashboards for tickets and VRF proofs.

• Security hardening; initiate independent audit and bug bounty.

2026

• Discovery features: search, curation, and community raffles with Quest flows.

• Extended asset support with curated whitelists and partner integrations.

• Additional Automation monitors and service coverage.

2027

• Formal verification for critical modules where feasible.

• Cross-app embed SDK; analytics APIs for creators.

• Continuous audits and security improvements.

Glossary & FAQ

On-chain. Logic and state are stored/executed on a public blockchain.

VRF. Verifiable Random Function returning randomness plus a proof verified on-chain.

Automation. Chainlink’s scheduled/conditional calls that automatically trigger contract functions.

Escrow. Custody of an asset by a smart contract until conditions are met.

How is the winner chosen?

VRF returns random words with a proof; the contract computes winnerIndex = randomWords[0] % totalEntries and transfers the prize to that ticket’s owner.

Can one user buy all tickets?

Yes. Equal odds per ticket; a single buyer may buy the full supply.

What if no tickets are sold?

The asset returns to the owner automatically; no draw occurs.

Can I cancel after launch?

Within the first 20 minutes only. After that, the asset remains locked until the countdown ends.

Where can I verify the result?

On BaseScan: view the draw transaction, VRF proof, and winner transfer.

Which assets are supported?

Curated ERC-721/1155 NFTs and verified ERC-20 tokens; ETH is supported.

Are there admin controls over winners?

No. There is no admin method to select winners; selection is entirely VRF-driven.

Is Triffle available outside Base?

Triffle focuses on Base; other networks may be considered later.