RaffleHash Whitepaper

A Sui-Native On-Chain Game Protocol

---

1. Abstract

RaffleHash is a Sui-native on-chain game protocol built around a protocol asset, RaffleHash (RFLH). It replaces the custodial casino account model with wallet-approved transactions, on-chain settlement, protocol-controlled reserves, and smart contract enforced monetary rules.

The protocol uses RFLH as the asset for gameplay, accounting, minting, burning, routing, and capped distribution, so game settlement is expressed through code rather than a private operator ledger or a house bankroll.

RaffleHash begins with two core game structures. The first is the raffle, a pooled game where users enter with RFLH, receive probability proportional to their contribution, and compete for a prize supported by player entries and available Prize Reserve matching. The second is individual-settlement gameplay, where each user transaction resolves directly against protocol rules. Roulette is the initial individual-settlement game.

Loss-side value from individual-settlement games is not retained as house profit. It is burned under a supply-sensitive burn curve or routed to the Prize Reserve. The Prize Reserve can then match raffle entries 1:1 while funds are available. The result is a circular game economy where participation distributes supply, losses constrain or recycle value, and the raffle becomes the main destination for protocol-recycled value.

---

2. The Traditional Casino Model

Online casinos are usually built around three operator-controlled systems: custody, settlement, and bankroll management.

Custody means users deposit funds into a platform-controlled account. The user may see a balance, but the operator controls the ledger and the withdrawal process.

Settlement means outcomes are produced by private backend systems. Even with result verification, the user still depends on the operator's execution path for seed handling, display, accounting, and withdrawals.

Bankroll management means the operator must already hold enough external assets to pay winners. If the game pays in dollars, stablecoins, or SUI, ETH, BTC, or any other outside asset, the operator must possess that asset before it can honor a payout.

An on-chain casino using an external currency inherits the same constraint: the ledger may be public, but the payout asset cannot be minted by the protocol, so large payouts still depend on holding enough of someone else's asset.

3. RaffleHash Protocol Model

RaffleHash uses a protocol-native asset model. RFLH can be minted, burned, routed, capped, and distributed according to smart contract rules. This changes the role of the game asset from a passive payment token into the monetary engine of the protocol.

Users interact through external Sui wallets. A frontend may build transactions, display game state, and present results, but the wallet remains the approval point and the smart contracts remain the settlement authority.

The protocol model has four core components:

- RFLH, the transferable protocol asset used for gameplay and settlement.
- The Prize Reserve, which receives non-burned loss-side value and can match raffle entries.
- The Treasury, which receives raffle fees until its cap is reached and supports protocol operations.
- Game Credits, non-transferable gameplay credits used for participation campaigns and airdrops.

These components are connected. Individual-settlement games distribute RFLH through wins, burn or reserve losses, and feed the Prize Reserve. The raffle turns that recycled value into a public pooled game. Treasury and Game Credits support participation without changing the core settlement model.

4. RFLH Monetary Design

RaffleHash begins with an initial supply of 10,000,000 RFLH. This initial supply is intended for launch liquidity rather than a public presale. Users need a permissionless way to acquire RFLH before entering games, and DEX liquidity provides that path without requiring private allocation as the primary access route.

RFLH targets a soft cap of 100,000,000 RFLH. This is not a hard maximum on supply but the level at which the burn curve reaches 100% and loss-side value is fully burned. Roughly 90% of the intended supply is designed to be minted through protocol participation rather than allocated upfront. RFLH uses 8 decimals, so all on-chain amounts are denominated in base units, each one hundred-millionth of a token.

RFLH has three roles:

- It is the transferable protocol asset.
- It is the accounting unit for game settlement.
- It is the medium through which supply is progressively distributed through gameplay.

This design gives RaffleHash a participation-driven supply model. Wins can mint RFLH according to game rules. Losses can burn RFLH or route value to the Prize Reserve. The supply schedule is shaped by gameplay, the burn curve, reserve routing, and protocol-enforced distribution limits.

5. Initial Liquidity

The first 10,000,000 RFLH are intended to establish DEX liquidity. Without liquidity, RFLH may be deployed but difficult to use. Users need a market path to acquire RFLH before participation and an exit path if they no longer want exposure.

Initial liquidity is also part of the launch distribution model. A presale can concentrate supply before broad participation begins. A broad airdrop can miss intended users and immediately create sell pressure. DEX liquidity gives users a public market route into the protocol.

Liquidity positions can be made transparent through the chosen DEX structure. Depending on the pool design, positions can be timelocked with staggered expiry dates or burned where the DEX supports permanent liquidity.

---

6. Individual-Settlement Games

Individual-settlement games resolve a user's transaction directly against protocol-defined rules. This class of game is the economic engine of the protocol: it is where new RFLH is minted to winners and where loss-side value is burned or routed to the Prize Reserve. Roulette is the first implemented individual-settlement game, but the engine is the class as a whole, and every game added to it contributes to the loop in the same way.

A user commits RFLH to a game. The transaction settles on-chain. The result determines whether value is returned, minted, burned, or routed.

On a win, the original stake is returned and the payout is minted in RFLH. On a loss, the committed RFLH is split by the burn curve. The burned portion is removed from supply. The remaining portion, where applicable, is routed to the Prize Reserve.

This creates a direct link between individual gameplay and the raffle economy. Losses do not accumulate in a private house balance. They either reduce circulating supply or move into a reserve that can later support raffle prizes.

Because there is no house account, the mathematical edge that would normally become an operator's profit has nowhere to accrue. In a conventional casino, the gap between true odds and paid odds is kept by the operator as revenue. In RaffleHash, that same edge is expressed entirely as protocol behavior: across many settlements it becomes RFLH burned and value routed to the Prize Reserve. Each individual-settlement game therefore converts its built-in edge into supply contraction and raffle funding rather than private income.

Roulette is the first concrete example. It implements European single-zero roulette over the numbers 0 to 36 with standard payouts: a straight bet pays 35 to 1, split 17 to 1, street 11 to 1, corner 8 to 1, six line 5 to 1, and the even-money, dozen, and column bets pay their usual amounts. The single zero gives European roulette a house edge of one part in 37, approximately 2.70%, corresponding to a return to player of about 97.3%. Routed through the protocol, that 2.70% edge is the average share of staked RFLH that is burned or sent to the Prize Reserve over time.

Further individual-settlement games, such as Dice, Keno, Plinko, Limbo, and Slots, can be added later as independent game packages. Each would carry its own defined odds and return to player, and each would feed the same loop: wins mint RFLH, losses burn supply or fund the Prize Reserve. What powers the economy is the individual-settlement game class collectively, not any single title.

7. Burn Curve

The RaffleHash burn curve is supply-sensitive. It changes how individual-settlement losses are handled as total RFLH supply expands.

At lower supply levels, individual-settlement losses are partly burned and partly routed to the Prize Reserve. As supply approaches the soft cap, the burn percentage increases. At and above the soft cap, individual-settlement losses are fully burned.

The current curve is:

- At 20,000,000 RFLH supply or below: 20% of individual-settlement losses are burned.
- Between 20,000,000 and 100,000,000 RFLH supply: the burn percentage increases linearly.
- At 100,000,000 RFLH supply or above: 100% of individual-settlement losses are burned.

The formula is:

```
burn_rate = 20% when supply <= 20,000,000
burn_rate = 20% + ((supply - 20,000,000) / 80,000,000) x 80% when 20,000,000 < supply < 100,000,000
burn_rate = 100% when supply >= 100,000,000
```

The curve gives the protocol different behavior at different supply levels. Earlier in the lifecycle, more loss-side value can reach the Prize Reserve and support raffle activity. Later, as supply expands, burn pressure becomes the dominant response.

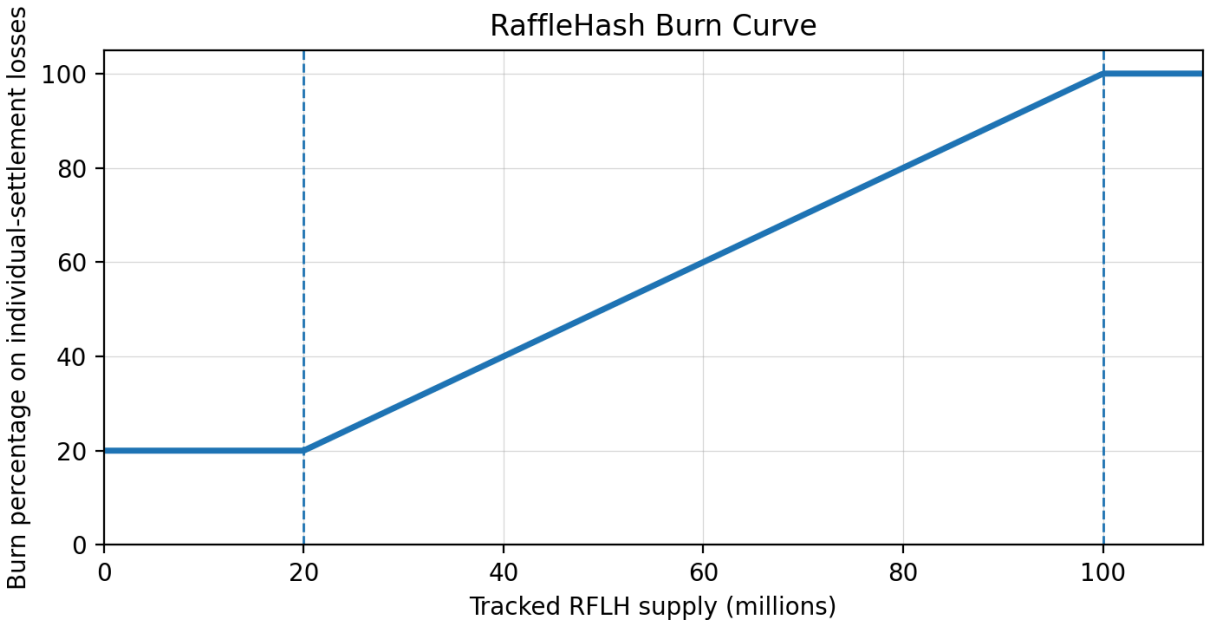

Figure 1. Burn percentage applied to individual-settlement losses as tracked RFLH supply increases.

8. Prize Reserve

The Prize Reserve is a protocol-controlled reserve that supports the raffle economy. It receives the non-burned portion of individual-settlement losses and can match raffle entries 1:1 while funds are available.

The Prize Reserve relationship is direct:

- Individual-settlement games generate wins and losses.
- Losses are split between burn and Prize Reserve routing.
- Prize Reserve funds can match raffle entries.
- Matched entries increase the effective prize available to participants.
- The raffle distributes the pooled prize according to protocol rules.

Loss-side contributions from individual-settlement games pass through an on-chain reserve contribution limit. While tracked RFLH supply is below 90,000,000, the amount added through this route is limited to 0.1% of current tracked supply in each two-hour window; any excess is burned. Once tracked supply reaches 90,000,000, this contribution limit is permanently disabled.

The Prize Reserve is not an operator bankroll. It exists to recycle protocol value into a player-facing pooled game. Its public framing is precise: the Prize Reserve matches raffle entries 1:1 while funds are available. If reserve funds are insufficient, matching is limited by the available amount.

---

9. The Raffle

The Raffle is RaffleHash's primary pooled game. Users enter with RFLH. Each entry increases the user-contributed pool and increases that user's chance of winning. The Prize Reserve can match user entries 1:1 while funds are available.

The raffle combines user participation with protocol-recycled value. It is not only a lottery funded by entrants. It is the destination for value accumulated through the wider RaffleHash economy.

If no Prize Reserve matching is available, the raffle is funded only by user entries. A 3% fee is taken from the total pool, leaving 97% of user-contributed value available to the winner. In aggregate the pool then returns 0.97 RFLH of prize value for every 1 RFLH entered.

If every user entry is fully matched 1:1 by the Prize Reserve, then each 1 RFLH entered by users creates 2 RFLH of gross pool value. After the 3% fee, 1.94 RFLH remains as post-fee prize value. This is the ceiling of the raffle payout ratio, 1.94 RFLH of prize value per 1 RFLH entered, reached only when the reserve fully matches every entry.

This 0.97 to 1.94 range is an aggregate measure of post-fee pool value relative to user-contributed entries, not an individual expected return. The raffle is winner-takes-prize, so any single participant's expected outcome depends on their share of the ticket range. The range also describes the pool before the winner payout cap is applied: at settlement, the winner payout is capped at 10% of current tracked RFLH supply, and any excess is deposited into the Prize Reserve. Individual outcomes depend on entry size, total pool size, timing, and the settlement result.

---

10. Raffle Timing and Anti-Sniping Design

The raffle begins with the first entry. The base duration is 2 hours, and only one raffle can be open at a time. Each entry receives a contiguous ticket range proportional to the amount entered.

During the final 20-minute window, an increase of at least 10% in the player-contributed pool extends the closing time by 20 minutes. The extension baseline resets after each qualifying increase, and the raffle cannot remain open for more than 48 hours from its first entry.

Entries are stored on-chain in bounded buckets rather than in a single growing vector. After the closing time, any user can submit the settlement transaction. The contract draws a weighted winning index, identifies the corresponding entry, deducts the raffle fee, and transfers the payout directly to the winner in the same transaction. There are no claim receipts or separate claim step. Settled entry buckets can later be pruned and the completed round object removed through permissionless cleanup calls.

---

11. Treasury

The Treasury receives raffle fees up to its 10,000,000 RFLH balance cap. Treasury funds support protocol operations such as liquidity operations, airdrops, prizes, and other protocol-supporting functions.

Each 3% raffle fee is routed according to the Treasury's remaining capacity. The amount required to reach the cap is deposited into Treasury, and any overflow from the same fee is deposited into the Prize Reserve. Once the cap is full, subsequent raffle fees are routed entirely to the Prize Reserve.

This creates a self-limiting administrative reserve. Treasury cannot exceed the configured cap through raffle fees, and fee flow continues through the Prize Reserve after the cap is filled.

---

12. Game Credits

Game Credits begin with an active supply of zero. They are created only when Treasury-held RFLH is burned: one base unit of RFLH burned creates one base unit of Game Credits in the protocol distribution pool. Active Game Credit supply is capped at 10,000,000, but the cap is not an initial allocation or a lifetime creation limit.

Game Credits are non-transferable ledger balances rather than Sui coins or wallet-held token objects. They cannot be traded or sold. Each recipient holds a single Game Credits balance with a rolling expiry. When new credits are distributed, the recipient's entire unexpired balance is reset to one fresh expiry, 30 days from that distribution and rounded up to the next UTC midnight, so a later distribution extends the life of the balance the user already holds.

Game Credits can be used in supported individual-settlement games but cannot be used to enter the raffle. Losing credit stake is removed from the user ledger and burned. Winning credit stake remains available, while the profit is paid in newly minted RFLH.

This structure supports airdrops, campaigns, prizes, and giveaways without distributing liquid RFLH directly. Expired balances are removed from active supply, allowing capacity under the active-supply cap to be used again.

---

13. Admin Powers and Path to Immutability

RaffleHash separates the monetary core from its games. The core package contains RFLH authority and supply tracking, Treasury, Prize Reserve, the Game Credits ledger, the burn curve, AdminCap, and the Game Registry. Raffle and Roulette are published as separate game packages that call the core through a restricted interface.

Each game package defines a witness type that can only be created inside that game module. Core functions that mint or burn RFLH, move Prize Reserve value, or debit Game Credits require this

witness and verify that its type is registered and enabled in the Game Registry. A new game can therefore be published and authorized without upgrading the core package or exposing a general minting interface.

Core administrative powers cover Treasury withdrawals and the creation and distribution of Game Credits. Game activation is controlled per registered game in the Game Registry, while entry and bet limits are stored in each game package configuration. These controls are visible through shared objects, flags, and emitted events.

The powers can be removed in stages. Treasury withdrawals, Game Credit creation, and Game Credit distribution can be permanently renounced individually or together. Limit updates can be permanently frozen for each game. A game's enabled or disabled state can also be locked permanently.

Upgrade authority is package-specific. The core UpgradeCap can be consumed to make the monetary core immutable. Each game package has its own UpgradeCap and can be made immutable independently, including before it is registered in the core Game Registry. This allows an immutable core to authorize additional game packages without changing core code.

The Game Registry has a separate finalization path. New game registration can remain open while the protocol expands, then be renounced permanently. Once registration is closed, registered game states are locked, game limits are frozen, core administrative powers are renounced, and the relevant UpgradeCaps are consumed, the resulting state is verifiable on-chain and cannot be restored through the retired authority paths.

---

14. Game Structures and Extensibility

RaffleHash is designed to grow as a catalogue of games in two structural families: pooled games, where many entries resolve into one shared prize event, and individual-settlement games, where each transaction resolves on its own. The Raffle is the current pooled game; Roulette is the current individual-settlement game.

New games are added as independent Move packages rather than by changing the monetary core. A game package interacts with core value only through the witness-gated interface described in the previous section, so the protocol can extend its catalogue while the RFLH authority, Treasury, Prize Reserve, and Game Credits ledger remain in a single, separately governed core.

This separation lets the game catalogue expand without expanding trust. Planned individual-settlement games such as Dice, Keno, Plinko, Limbo, and Slots would each ship as their own package with their own odds and limits, be registered and made individually freezable, and feed the same economic loop as Roulette. The two families define how a game settles; the core defines what value a game can touch.

---

15. On-Chain Settlement on Sui

RaffleHash is built on Sui's object-centric smart contract environment. Protocol state is represented through Sui objects, separately deployed Move packages, wallet-approved transactions, and emitted events. SUI is used for network gas. RFLH is the protocol asset used for game economics.

A RaffleHash transaction follows this model:

- The user connects a Sui wallet.
- The user chooses a protocol action.
- The frontend builds a transaction.
- The user approves the transaction in the wallet.
- Sui executes the transaction.
- The RaffleHash contract updates state and emits events.
- The frontend reads state and displays the result.

Where randomness is required, RaffleHash uses Sui's native randomness beacon. At the beginning of each epoch, the validator committee runs a Distributed Key Generation process that creates shares of a distributed secret key. No individual validator holds the complete key. During the epoch, validator shares are combined once the required stake threshold participates, producing fresh beacon output through the consensus system. Provided less than one-third of Sui voting power is Byzantine, no single validator can choose the output, predict it in advance, or generate it alone.

Move contracts access the beacon through Sui's shared Random object at reserved address 0x8. A game creates a transaction-bound random generator from the Random object and the transaction context, then derives the required value, such as a roulette number or a weighted raffle index. The Random object cannot be mutably controlled by an application or user.

RaffleHash consumes randomness inside the settlement entry point and completes the corresponding accounting in the same transaction. Roulette draws the spin result and settles the submitted bet set atomically. The raffle draws an index across the player-contributed ticket range, resolves the winning address, and pays the winner before the transaction completes. The frontend reads the resulting transaction effects and events; it does not generate or select the outcome.

---

16. Trust Model

RaffleHash changes the trust model by moving custody and settlement away from a private operator system.

Users do not deposit into a platform-controlled casino account to maintain an internal balance. They approve transactions from their own Sui wallets. The frontend can be replaced, replicated, or bypassed by other tools that build valid transactions.

Game rules and asset accounting are implemented in Move contracts. Randomness-dependent games receive their input from Sui's validator-operated beacon, and the contract maps that input to the outcome and settlement effects. Transaction effects, state changes, and protocol events provide the execution record.

The protocol state is visible on-chain: deployed package identifiers, package upgrade status, registered game types, locked or active game controls, configured limits, AdminCap renunciation flags, Treasury and Prize Reserve balances, and emitted events.

This makes the protocol independently verifiable. A user does not have to trust the frontend: they can confirm from the transaction which package settled a game, check that settlement drew from Sui's shared randomness object, read the emitted events for the exact amounts minted, burned, and routed, and inspect the on-chain flags to see which administrative powers and upgrade paths have already been renounced. The same records let anyone reconstruct the loop's behavior after the fact rather than relying on a private operator report.

17. Risks and Limitations

RaffleHash is experimental software that manages real on-chain value, and participants should understand its limits. Smart contract risk is inherent: although the monetary core is designed to be made immutable, immutability also means that a defect discovered after the core's UpgradeCap is consumed cannot be patched in place. Before renunciation and freezing, administrative powers exist and must be trusted not to be misused; after them, the trade-off shifts toward permanence.

The economics are participation-driven and are not a promise of return. The raffle's fully matched payout ratio is a ceiling that depends on Prize Reserve funds being available, not a rate any participant should expect; when the reserve is empty the raffle pays only from user entries after fee. Individual-settlement games carry a defined house edge that, by design, works against the staking player over time even though it accrues to the protocol rather than to an operator. RFLH's soft cap is a burn-curve threshold, not a claim about price, liquidity, or future supply.

The protocol also depends on components it does not control: the Sui network and its randomness beacon for settlement and fairness, DEX venues for liquidity, and the user's own wallet for custody and transaction approval. Regulatory treatment of on-chain games varies by jurisdiction and may restrict who can participate. Nothing in this document is investment advice, and RFLH is intended for protocol participation rather than as an expectation of profit.

---

18. The RaffleHash Loop

RaffleHash is a circular game economy:

- Users acquire or receive RFLH.
- Users play individual-settlement games or enter the raffle.
- Individual-settlement wins mint new RFLH.
- Individual-settlement losses are burned and/or routed to the Prize Reserve.
- The Prize Reserve matches raffle entries while funds are available.
- The raffle distributes pooled value to the winner.
- The 3% raffle fee funds Treasury until the Treasury cap is reached.
- After the cap, raffle fees route to the Prize Reserve.
- Treasury supports protocol operations and participation programs.
- Treasury RFLH can be burned to create non-transferable Game Credits for distribution.
- Distributed Game Credits expire after 30 days; wins mint RFLH profit and losses burn the losing credit stake.
- Supply expansion is constrained by the burn curve, soft-cap payout rules, and protocol-enforced distribution limits.

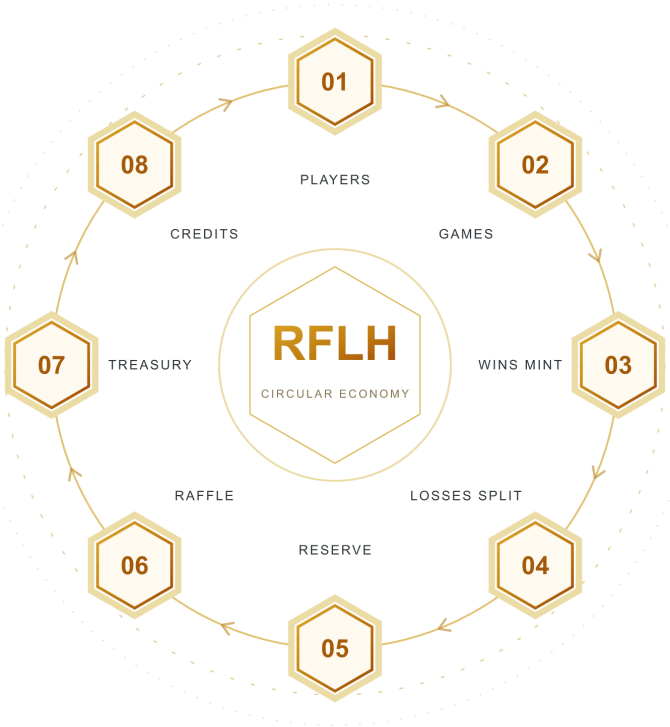

Figure 2. The RaffleHash loop: value moves clockwise from players and games through wins, the burn and reserve split, the raffle, Treasury, and Game Credits, and back.

The mechanisms operate as one system. Minting distributes supply through wins. Burning constrains supply. The Prize Reserve recycles value into the raffle. Treasury supports operations. Game Credits fund non-transferable participation. The immutable core and separately registered game packages allow the protocol to retain a stable monetary base while extending its game catalogue.

---

19. Conclusion

RaffleHash is a protocol-native alternative to the custodial casino model. Users interact through Sui wallets instead of platform-controlled balances. Game settlement is enforced by smart contracts instead of private backend accounting. The protocol uses RFLH as a native game asset that can be minted, burned, routed, capped, and distributed through gameplay.

The core economic design is the loop between individual-settlement games and the raffle, set out step by step in “The RaffleHash Loop” above: wins mint RFLH, losses burn supply or feed the Prize Reserve, and the reserve turns recycled value into a public pooled prize.

Game Credits extend the participation model without creating a second liquid asset. Treasury supports protocol operations but is capped. The monetary core is separated from independently published game packages: core powers can be renounced, the core package can be made immutable, game packages can be frozen before registration, per-game controls can be locked, and new registration can later be closed permanently.

RaffleHash should be understood as an on-chain game economy: non-custodial, permissionless, transparent, supply-aware, and designed around participation-driven distribution.