

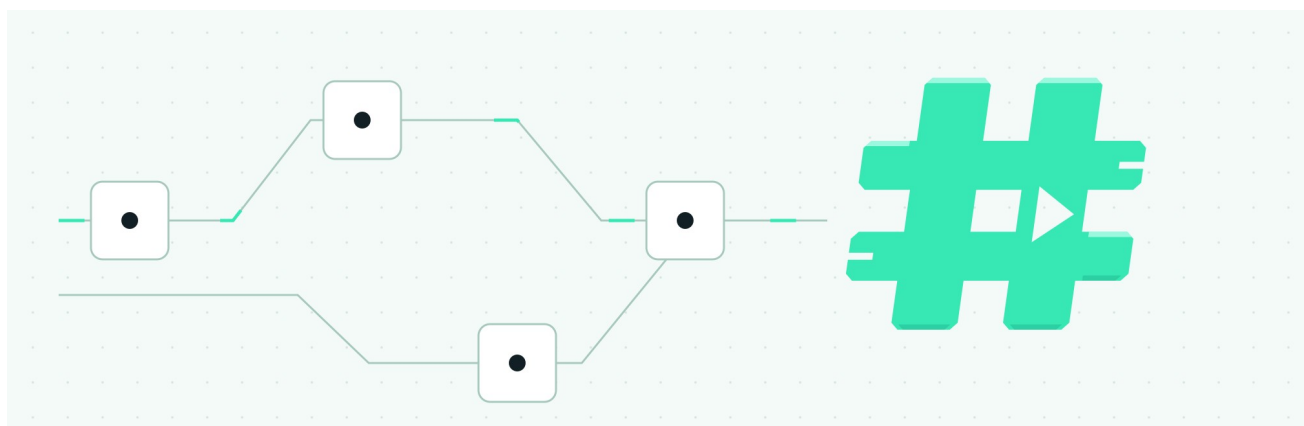


PROJECT WHITEPAPER

HashPlay Whitepaper

Platform mechanisms and fund flows

For users and partners



Version 1.0 / 30 October 2026

Contents

01	Project overview	02
02	Hash calculation and verification	03
03	Contracts and settlement	04
04	Fund flows and safeguards	05
05	Accounts and service rules	06

01 Project overview

HashPlay is an interactive entertainment platform built around block hashes. It brings wallet sign-in, round information, order records and account management into one interface. Users can recalculate published hashes and check results against order and transaction records.

The platform combines a seed commitment with a block hash. Before participation, it publishes the seed proof, target Ethereum block height and closing time. Once that block exists, users can recalculate the result using its actual hash and the revealed seed.

HashPlay contracts validate order authorizations, reserve payouts, process settlements and handle timeout refunds, with events recorded on-chain. The platform maintains individual account balances and determines outcomes under the applicable rules.

Platform features

Feature	Available information and actions
Wallets and accounts	Wallet sign-in, balances and account history
Rounds and orders	Round parameters, order submission and processing status
Hash verification	Seed proofs, block parameters and calculated hashes
Fund services	Settlement records, withdrawals and applicable refund procedures

Participation

Users choose a room and token, then review the round number, closing time, limits and fees. Before submitting an order, they save the seed proof and round parameters. After submission, they retain the order ID and amount.

When results are published, users obtain the seed and block parameters from the hash records, recalculate the hash and check the credited amount. Platform balance orders and direct wallet orders follow the funding paths configured for the relevant room.

This whitepaper describes platform features and protocol mechanisms. Available features, tokens, contract addresses and applicable rules are specified in the platform's published information.

02 Hash calculation and verification

Seed proof

Each room uses an independent 32-byte random seed for every round. The seed proof and target block height are published before participation. The original seed and actual block hash are disclosed with the result. Users compare the recalculated proof with their saved record to check the seed, room ID, round number and target height.

Published results use SHA-256 over the following UTF-8 text:

```
proofText = roomId + ":" + periodNo + ":" + blockNumber + ":" + seed
drawText = proofText + ":" + normalizedBlockHash
```

```
drawSeedProof = hex(SHA-256(UTF-8(proofText)))
drawHash = "0x" + hex(SHA-256(UTF-8(drawText)))
```

roomId, periodNo and blockNumber are the room ID, round number and target block height, expressed as decimal text without leading zeros. seed is 64 lowercase hexadecimal characters. normalizedBlockHash is the 64-character lowercase block hash with the 0x prefix removed. Fields are separated by a colon, with no spaces or line breaks. Hexadecimal text is hashed directly, without first decoding it to bytes. Digests use lowercase hexadecimal output.

Verification steps

1. Before participation, save the seed proof, room ID, round number, target height, closing time and applicable rules.
2. After publication, recalculate drawSeedProof using the revealed seed and compare it with the saved proof.
3. Use an independent node or block explorer to check the hash and confirmation status of the specified Ethereum mainnet block.
4. Recalculate drawHash, then check the order outcome and fund records under the applicable rules.

Closing time and business type are not included in this digest and should be saved with the round rules. A proof retrieved later does not replace the record saved before participation.

Contract result proof

HashPlay contracts use Keccak-256 and Solidity ABI encoding. The seed and block hash are decoded from hexadecimal text into 32-byte values, then encoded as bytes32:

```
seedCommitment = keccak256(abi.encode(seed))
resultHash = keccak256(abi.encode(seed, ethereumBlockHash))
```

drawHash serves public verification; resultHash serves the contract proof. Their encodings and digests are not interchangeable. Contracts validate authorizations and proof consistency. Block authenticity is checked separately; the platform determines outcomes under the rules and authorizes settlement.

03 Contracts and settlement

Contract roles

HashPlayFactory validates creation authorizations; HashPlayGameDeployer deploys game contracts. HashPlayRouter provides the common entry point. Each game has a dedicated HashPlayMargin. HashPlayVault holds pooled tokens for balance transactions.

The first accepted bet locks the round's seed commitment, target Ethereum block number and closing time. Later bets share that configuration. Orders must arrive before closing and within their authorization period. Acceptance requires a confirmed successful transaction and a matching BetPlaced event.

Reserved order funds

Let A be the stake and G the agreed gross payout, including the stake. At acceptance, the wallet or vault supplies A and the margin contract supplies G minus A . The game contract reserves G for that order. A failed funding or authorization check reverts the entire transaction.

Each pending order keeps its reserve until settlement or a timeout refund. The margin owner may withdraw available funds from the margin contract; reserves already in the game contract are released under the order rules.

Platform fees

The HashPlay contract's platform fee rate is 5%, calculated per order on the applicable base and rounded down to the token's smallest unit. A winning user's fee is based on gross payout minus stake and deducted from the payout. For a losing order, the fee is based on the stake earned by the margin provider. A draw incurs a stake-based fee paid from margin; the user's stake is returned in full. Timeout refunds carry no platform fee. Gas is separate.

The example below uses a stake of 100 and a gross payout of 200. The total reserve is 200, including 100 supplied as margin. All amounts use illustrative units of the same token.

Outcome	User or vault receives	Margin contract receives	Platform fee
Win	195	0	5
Loss	0	195	5
Draw	100	95	5
Timeout refund	100	100	0

Margin returns include returned contributions; net change deducts the original margin. Balance-order returns enter the vault and are credited to individual accounts by order.

These rules cover the relevant HashPlay contract version. The room rules and deployment specify funding paths, odds and token precision.

04 Fund flows and safeguards

Order fund flows

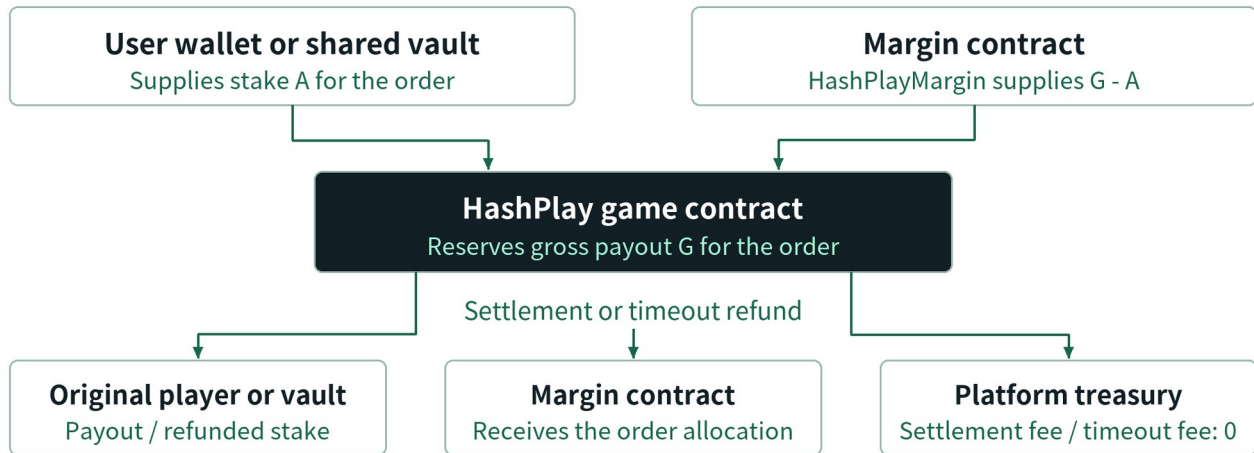
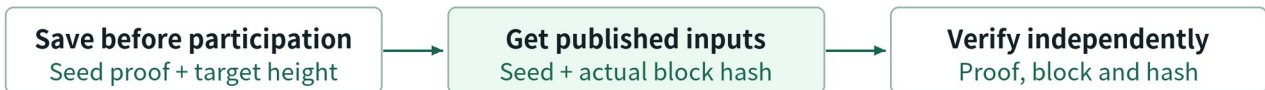


Figure 1 Funding reserves and distributing funds for contract orders

The original order determines the source of funds and the return path. Wallet orders pay the original player directly. Balance orders return funds to the shared vault and are linked to the individual ledger. Allocations depend on the outcome; zero-value transfers are skipped. Timeout refunds carry no platform fee.

Verification and fund processing

Public verification



Contract execution



Timeout recovery

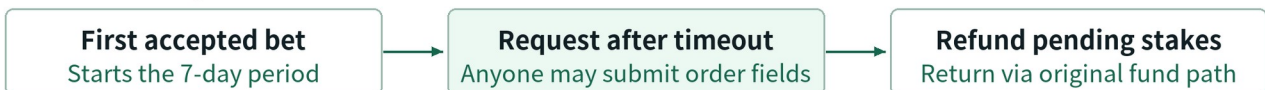


Figure 2 Public verification and contract processing checks

The target height is published before participation. The actual hash is obtained once the block exists and checked together with the revealed seed. Public verification and contract proofs follow their respective calculation rules. Block sources and business outcomes are checked separately. Vault administration is described in the next section.

05 Accounts and service rules

Accounts and withdrawals

The platform ledger maintains available and frozen balances and account history. HashPlayVault holds pooled tokens for balance transactions; it does not maintain individual balances.

To withdraw through HashPlayVault, the platform checks and freezes the withdrawable balance, then signs an authorization binding the user, recipient, token, amount, order ID and expiry. The user submits a transaction; the contract verifies the authorization and vault funds before transferring tokens. Successful withdrawal orders cannot be replayed. A valid authorization, sufficient funds and an executable token transfer are required.

Wallet orders pay the original player. Balance orders return funds to the vault and are linked to individual account records. The paths are reconciled separately to prevent duplicate credits.

Timeout refunds

A round's processing deadline is seven days after its first accepted bet, shared across all its tokens. Results can be published after closing and before that deadline. Settlement requires a published result and must also precede the deadline. At the deadline, pending contract orders become eligible for stake refunds using their original order fields.

Anyone may submit an eligible refund through the router without a new platform signature. A transaction must be submitted and gas paid. Wallet stakes return to the original player; balance stakes return to the vault for the platform to credit against each order. Margin returns to the margin contract. Settled orders remain final. User refunds cover the stake.

Refunds require accepted HashPlay contract orders matching their original records, and a chain and token that support transfers. Platform payment cancellations and other refunds follow their respective procedures.

Administrative permissions

The game owner manages business signers. The bet signer authorizes acceptance; the result signer authorizes publication and settlement. Contracts require valid authorizations. The vault administrator manages access, withdrawal signers and pooled funds, with administrative withdrawal authority. Administrative and authorized user withdrawals are recorded separately.

Margin providers bear the corresponding payouts. Users' stakes change with order outcomes and may be lost. Participants should understand the applicable rules, fees and permissions and meet the service's eligibility requirements.

Record lookup

Check results against the round number, order ID, amount and rules. For on-chain orders, retain the chain ID, contract address and transaction records. BetPlaced, BetSettled and BetRefunded identify acceptance, settlement and refunds. Balance orders also require reconciliation with individual account history.