Rubik Pool — Information document

Version of 29 October 2026 · early capped beta · https://rubik.capital

Please read this document before you connect a wallet or deposit. It describes how the pool works, what can go wrong, how to verify the contract yourself and answers common questions. Where the text says “this page”, it means https://rubik.capital/connect. The legal terms are on https://rubik.capital/terms (draft).

In short: the pool is an early beta. The contract is not audited, the operator holds all the keys, the pool has a size limit, and you can lose all of your deposit.

1. How it works

Four steps, start to finish.

Deposit

Send USDC from your Hyperliquid spot balance to the pool, from the address that should receive your payouts; this page signs the transfer for you. It is a Hyperliquid transfer, not a contract call: no gas, minimum 10 USDC. Only this kind of transfer is credited; HyperEVM transfers, perp-balance or HIP-3 transfers and other tokens can get stuck. The keeper credits the deposit once it sees it on Hyperliquid's public ledger, usually within minutes.

The pool trades

All capital trades as one book on two separate Hyperliquid trading accounts, which split the money 50 % / 50 %. Until participants' capital reaches 1,000 USDC, deposits wait in the vault; then the keeper funds both accounts and trading starts. Later withdrawals below that amount do not stop it. Each account keeps at most 60 % of its capital in open positions. The start amount and that cap are rules of the operator's keeper and bot, not of the contract. A trade belongs to the participants whose capital was in the pool when it opened; a new deposit joins new trades only.

Weekly settlement

Results are settled once a week (Mondays at 04:00 UTC). The performance fee, 50 %, is charged only on net profit above your own high-water mark, and is zero in a week that ends below it. The rate is fixed at deployment and is the same for everyone; referral rewards come out of the operator's share, never on top of yours.

Withdraw and get paid

Request a withdrawal any time (a HyperEVM transaction; you pay the gas in HYPE). Requests are served in order. The part of your capital not tied up in open trades is paid first; the rest follows as trades close: up to 17 days for the bot to close positions, plus up to 7 days to the next settlement. Payouts are automatic, only to the address you deposited from, and accumulate until they reach 10 USDC.

2. Risks

Total loss is possible

The pool trades leveraged perpetual futures. Prices can move through a protective stop; in the worst case you lose your whole deposit.

You may pay for someone else's loss

If a trade closes through its protective stop (a flash crash, a gap, a delisting), its loss can exceed one participant's money. The other participants in that trade cover the difference, each at most their own share and their own money; anything still uncovered is shared by the whole pool. The participant who caused it repays from future profit, if there is any. The operator does not cover losses and there is no reserve fund.

You are trusting the operator

A program run by the operator (the keeper) records deposits, reports prices and trade results, and runs the weekly settlement; the operator alone holds the admin, guardian and keeper keys. The contract cannot see who sent a deposit and does not cross-check what the keeper reports. A mistake or dishonesty could credit a deposit to the wrong address or report fake results that move pool money to the operator or others, up to the whole pool; in effect, the operator can technically take the pool's money. The size limit only caps new deposits, and the operator can change it. Only deposit what you can afford to lose entirely.

Early stage, no audit

The contract is unaudited and the deposit path is new. Hyperliquid, its bridges or the contract code can fail in ways that lose money. There is no legal wrapper yet: entity, jurisdiction and terms are not final.

A refused deposit may not come back

A deposit that arrives while the pool cannot credit it (in debt, paused, full, or during settlement) is not credited. The keeper sends it back to the sending address through the contract's return function, but that path has not yet been used with a real refusal on mainnet, and some cases (another token, a perp-side transfer) are returned by hand, so a return may come late. This page checks first, but only as a snapshot; sending from anywhere else skips those checks.

Getting money out can take time

Money tied up in open trades is paid as they close: up to 17 days for the bot to close a position (a rule for the bot, not the contract) plus up to 7 days to the next settlement, and payouts pause while a settlement runs. If the keeper stops or a settlement stalls, anyone can trigger an emergency freeze after a waiting period (from 4 days; terms, section 11). A freeze cannot be undone, and only the operator's bot can close open positions.

A trading account's mode can change for good

Hyperliquid lets a trading key switch its account into unified mode, and that cannot be undone. In that mode the contract's balance checks for that account read differently. The keeper checks both accounts' mode at every step and, if it changes, stops its money operations until the operator resolves it, so crediting, payouts and settlements can be delayed. Only the operator's bot holds the trading keys, and the guardian can revoke them.

Not available to everyone

US persons are not accepted. Participants confirm their own status; no identity checks are performed.

3. Verify the contract yourself

You do not need to trust this website: every step works from a public explorer or a terminal. Do them before your first deposit and whenever the operator announces a change.

1 · The address is the same everywhere

Compare the pool address on this page, in the contract manifest the operator publishes (X and Telegram, linked in the footer), and in the destination shown in your wallet's signing window. They must match character by character; if any one differs, stop.

```
cast call  "ledger()(address)" --rpc-url $RPC
```

2 · The source code is public and verified, for every contract

The pool is several contracts: the vault (PoolVault), the ledger (PoolLedger), two trading accounts (TradingLeg) and logic contracts the ledger calls for read-only views and the weekly settlement. The published manifest lists every address. Open each on Sourcify and on HyperEVMScan: each must show verified source, and Sourcify must report an exact match. Any contract without verified source is a stop.

```
curl -s https://sourcify.dev/server/v2/contract//
```

3 · The code on chain is the published code

Build the published source (the commit and hash are published with the address) with the pinned compiler, or let Sourcify do it: an exact match means identical bytecode and metadata. The vault stores its ledger and trading-account addresses in its code, so its hash differs per deployment; compare it with the hash published for this one, which the live data block also shows.

```
cast codehash  --rpc-url $RPC
```

4 · Nobody can swap the code

The contracts have no proxy, no upgrade function and no self-destruct: the vault's EIP-1967 implementation slot must be empty. The ledger delegates read-only views and the settlement to logic contracts whose addresses are fixed at construction (see the manifest) and cannot be changed.

```
cast storage  0x360894a13ba1a3210667c828492db98dca3e2076cc3735a920a3ca505d382bbc --rpc-url $RPC
```

5 · Who holds which key, and what each can do

Read admin(), keeper() and guardian() on the vault, and the role-change delay DELAY(). The admin is a single-signature Safe held by the operator; the guardian is a separate hot key, also the operator's. A role change is announced by a Proposed event at least DELAY before it applies; watch Proposed, AdminChanged, KeeperChanged and GuardianChanged. What each role can do is listed below.

```
cast call  "admin()(address)" --rpc-url $RPC   # also keeper(), guardian(), DELAY()(uint256)
```

6 · The numbers fixed at deployment

limits() returns the minimum deposit and the participant limit; params() returns the treasury address and the fee rate in basis points (its fourth value). The ledger's fallback serves both, so explorers may not list them. They must match this page, and the fee rate is fixed: no function changes it.

```
cast call  "limits()(int256,uint256)" --rpc-url $RPC   # also params()
(address,address,int256,int256,int256,int256,int256,int256)
```

7 · The money is on HyperCore

Open the pool address and the two trading accounts in the Hyperliquid explorer. The vault's spot USDC must be at least what the ledger owes; the contract refuses any keeper action that would break this. reconcile() returns the vault's balance, what the ledger holds for payouts and the trading accounts' value. legShareBps() is the first account's share of every transfer to trading in basis points (5000 means half); it must match the split on this page, 50 % / 50 %.

```
cast call  "reconcile()(uint256,uint256,int256)" --rpc-url $RPC   # also legShareBps()(uint256)
```

8 · projectedShortfall() is zero

Call ledger() on the vault, then projectedShortfall() on that address (the ledger's fallback serves it, so explorers may not list it). Above zero means a deposit would be refused: do not deposit until it is zero.

```
cast call  "projectedShortfall()(int256)" --rpc-url $RPC
```

9 · Your own deposit, after it is credited

The vault emits DepositCredited(investor, amount, ledgerRef); ledgerRef is the hash of your Hyperliquid transfer, and your position is readable from the ledger. Code cannot check who sent a transfer (the keeper reads that from Hyperliquid's public ledger) or the prices the keeper reports; compare both with the public ledger and the venue's own prices.

```
cast call  "investorView(address)(int256[11])"  --rpc-url $RPC
```

What each role can do

Keeper. An operator-run service: credits deposits, reports prices and trade results, processes withdrawals, runs the settlement and moves money between the vault and the trading accounts. Money booked to participants can only be sent to its holder; the one exception is returning a transfer the pool did not credit (see Risks), where the keeper names the recipient and can only reach money the books do not hold. The contract does not cross-check the results the keeper reports: this is the main trust assumption. The size limit caps only new deposits, and the only brake is the guardian, also the operator's key.

Admin. Replaces the admin, keeper, guardian and trading keys and changes how money sent to trading is split between the two trading accounts, only after a public delay (DELAY(); watch the events). Immediately: sets the size limit, lifts a deposit pause and restores a keeper or trading key the guardian revoked. Approves and revokes partners.

Guardian. Can only take away: pause deposits, revoke the keeper or a trading key, cancel a pending role change (all but its own replacement), cancel a trading account's open orders (which also removes its protective stops). It cannot move money. Revoking the keeper stops the service until the admin restores it.

Trading keys. Trade the trading accounts. They cannot withdraw to another address, but trades can lose money. A trading key can also switch its account into Hyperliquid's unified mode, which cannot be undone (see Risks).

Nobody. Pays a participant's money to any address other than the one that deposited. The other outgoing payments are the operator's fee to its treasury, referral rewards to the referrer's own address and, once built, returns of refused transfers.

Quick start

Read-only commands with Foundry (they cost nothing). Set RPC to the network you are checking:

```
# RPC: Mainnet https://rpc.hyperliquid.xyz/evm
export RPC=https://rpc.hyperliquid.xyz/evm
cast call  "ledger()(address)" --rpc-url $RPC
cast call  "projectedShortfall()(int256)" --rpc-url $RPC
cast codehash  --rpc-url $RPC
```

4. Questions about the pool

Who runs the pool?

The operator, Rubik Capital: it runs the trading system and the keeper, and holds the admin, guardian and keeper keys. The legal entity and jurisdiction are not final yet; see the draft terms.

What do I need to join?

A Hyperliquid account with spot USDC and a wallet that can sign for it. A deposit needs no gas; withdrawing and settings are HyperEVM transactions and need a little HYPE. US persons are not accepted.

How do I deposit, and can I send from somewhere else?

Use this page: it runs the checks before you sign. Only USDC sent from your Hyperliquid spot balance to the pool's spot balance is credited, from the address that should receive your payouts, so not from an exchange. Transfers from HyperEVM or a bridge, to the pool's perp balance or a HIP-3 dex, and other tokens are not credited and can get stuck. The minimum is 10 USDC, and the keeper credits usually within minutes.

What are the minimum and the limit?

The minimum deposit is 10 USDC. The pool has a size limit, measured against participants' capital plus the new deposit, and a limit on participant addresses (10 in the beta; only the operator can raise it, only after a public 48-hour delay and never above 43). The size limit starts at 1,000 USDC and rises in steps (3,000, then 10,000 USDC), each only after two clean weekly settlements in a row. Both limits are read from the contract and shown under Pool status; the operator sets them and can change them.

Can I join trades that are already open?

No. There are no units bought at a price. Each trade belongs to the participants whose capital was in the pool when it opened, and its result is split among them. A new deposit joins new trades only.

Can the contract be changed?

The code cannot: there is no proxy, no upgrade function and no self-destruct. What can change is who holds the roles (after a public delay) and, immediately, the size limit, the pause switch, partner approvals and the restoring of a keeper or trading key the guardian revoked. Everything is readable on chain.

Is it audited or regulated?

No. The contract is unaudited, and the legal structure is not final. The source code is public and verified so you can read it, but that is not an audit, and its licence is source-available, not open source.

Can I deposit today?

Yes, on mainnet, within the beta limits.

What if the keeper or the operator disappears?

Anyone can freeze the pool once a condition holds: no keeper operation for 4 days, a withdrawal request unprocessed for 4 days, a settlement open for 4 days, no settlement for 15 days, the oldest queued request older than 28 days, or an announcement by the admin plus 48 hours. A freeze cannot be undone. Then each participant can claim a share of what is in the vault and of the free margin of the trading accounts, only to their own address. The contract does not close open positions: only the operator's bot can. See the terms, section 11.