



Verdant

Programmable escrow for anything.

Non-custodial, condition-driven escrow — built on Robinhood Chain.

● On Robinhood Chain

verdant-peach.vercel.app · **Presale live**

THE PROBLEM

Trust between strangers is rented — expensively.

Every deal between strangers needs trust. Today that trust is fragmented, custodial, and one-size-fits-all.

UP TO

20%

Fiverr's cut on a freelance deal — charged whether the work is good or not.

Escrow agents stay slow, manual, and custodial — they hold your money to release it.



Fragmented

A different siloed product for every use case.



Custodial

A middleman holds your money to release it.



Slow & manual

Escrow agents settle by hand, on their timeline.



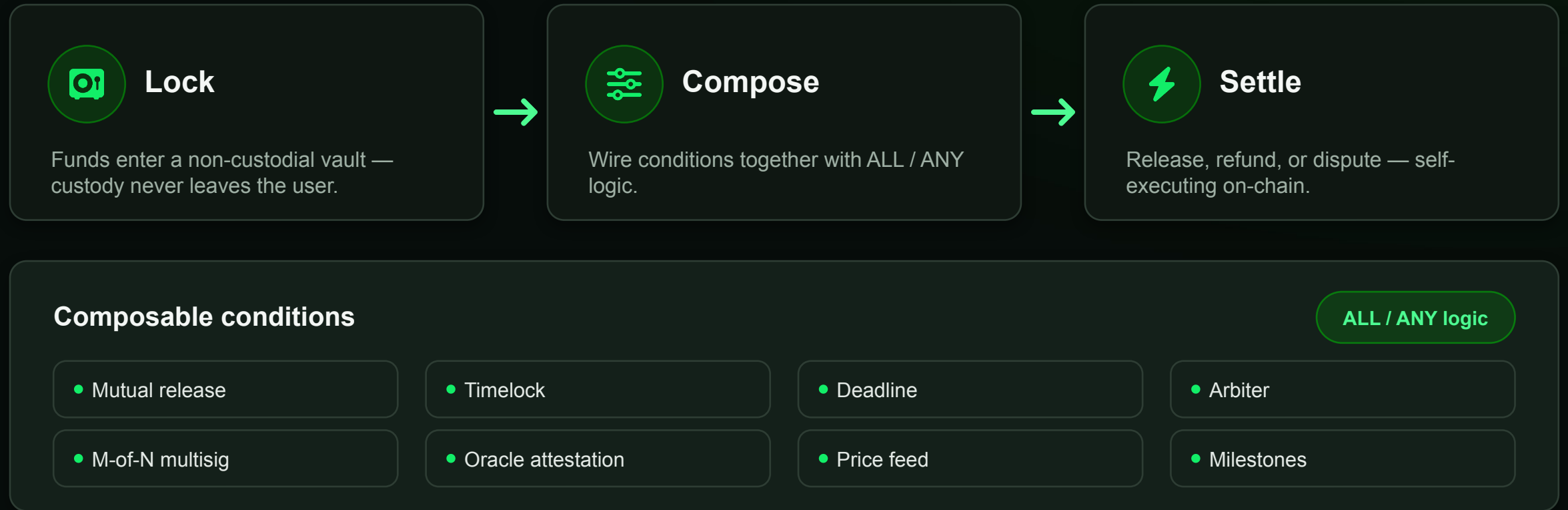
One-size-fits-all

No neutral, programmable layer for trades, wagers, or agreements.

THE SOLUTION

One programmable escrow primitive.

Assets lock in a non-custodial vault and release when a configurable set of on-chain conditions is met.



Four products. One primitive.

Templates are presets over the same vault. Drop to custom and wire any condition yourself.



Wager

Stake it. Winner takes the pot.

Oracle OR Deadline



Service / Gig

Fiverr-style milestone escrow — pay as work lands.

Milestone + Deadline + Arbiter



P2P Trade

Swap assets or goods with a safety net.

Mutual release + Deadline



Agreement

Put money behind any promise.

Mutual + Arbiter + Timelock

BUILT, END TO END

The whole stack is live today.

Not a concept deck — a working product. A live demo and presale are already running.



Web app

Create wizard, dashboard, and escrow detail — build, fund, track, and settle.



SDK + API

A typed TypeScript SDK for integrators, plus a REST API and indexer.



~150 automated tests

Fully typed, contracts-to-web, mirroring on-chain structs 1:1.

● Live demo + presale running

One spine, four layers

Web dApp

Next.js — build, fund, settle

SDK + API

Typed client · REST · indexer

packages/shared

Source of truth: types, fees, chains

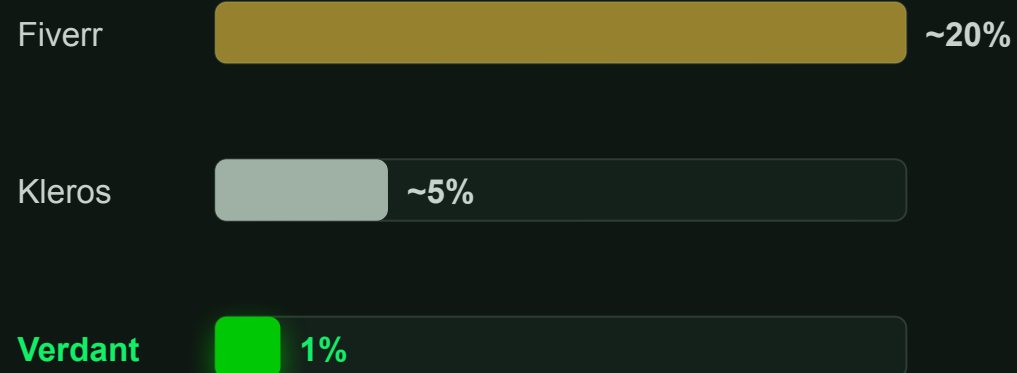
Contracts

Foundry · Solidity · Arbitrum Orbit

We earn only when the deal works.

1% on successful release — nothing on create, nothing on refund. Split 50/50, hard-capped on-chain.

Protocol take-rate vs. peers



1% base · 3% hard ceiling · \$500 per-deal cap — every parameter on-chain.

More ways it earns



Arbitration fee

20% protocol cut — refunded to the winner.



Idle-funds yield

Optional, principal-guaranteed; 50% of yield.



White-label rails

Builder fees for integrators (up to 2%).



Premium & subs

Power-user and template upsells.

We win on price **and** trust.

WHY NOW

Robinhood Chain just went live.

An EVM L2 for tokenized real-world assets across 120+ countries. That economy needs an on-chain settlement layer.



Global by default

Tokenized stocks and RWAs across 120+ countries on one EVM chain.



On-chain assets need on-chain deals

Tokenized value needs programmable agreements — escrow is that settlement layer.



The model that sticks

Non-custodial and programmable is what users and regulators will accept.

Chain live



RWAs tokenize



Agreements settle on Verdant

Built to be trusted with money.

Non-custodial by design — principal always returns to the rightful party.

INTERNAL AUDIT

0

open findings

33 issues fixed — every critical and high resolved and regression-tested.



Non-custodial

Funds live in the contract; principal returns to the rightful party.



Adversarial audit

6 independent lenses, 3-way verification, iterated to zero findings.



No predatory incentives

Success-only fees; winner-refunded arbitration; caps enforced on-chain.



External audit gated

A professional external audit is required before mainnet.

From testnet to the settlement layer.



NOW

- Live on Robinhood Chain testnet
- Presale open
- Internal audit complete



NEXT

- External audit → mainnet launch
- On-chain arbitration marketplace
- Idle-funds yield



THEN

- Cross-chain via bridges
- Institutional / OTC escrow
- White-label rails

Pre-launch — but everything is built.

Honest status: the protocol exists end to end and the presale is open.



Full stack shipped

Contracts + SDK + API + web, all typed.



Audit converged clean

Internal audit iterated to zero findings.



Live demo running

Build, fund, and settle escrows today.



Presale open

Back the protocol now.

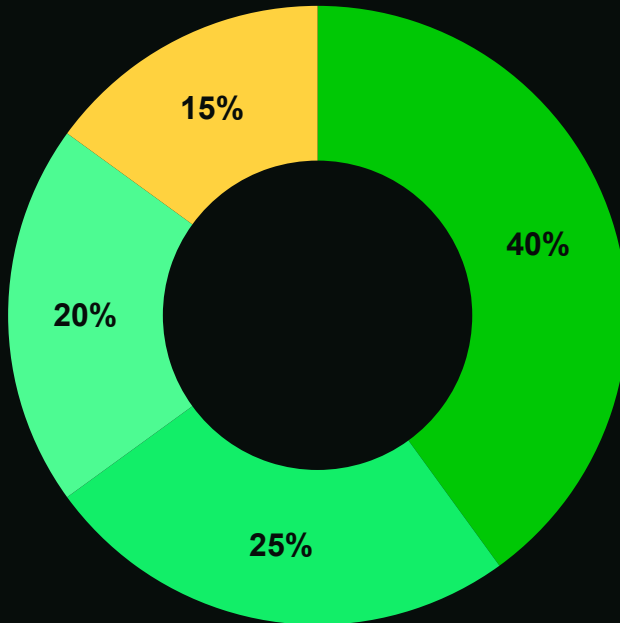


Kept honest

Pre-launch: no user counts or revenue to report yet. Those numbers come after mainnet — this deck states only what is true today.

Presale is live. Back the protocol.

Fund the path to mainnet. Allocation and raise target below are placeholders for the founder to finalize.



Use of funds

[PLACEHOLDER ALLOCATION]

● Dev	40%	Engineering to mainnet and beyond.
● Liquidity	25%	Protocol liquidity and market depth.
● Audit + bounty	20%	External audit and ongoing bug bounty.
● Growth	15%	Distribution, integrations, and community.

Raise target: [TBD]

Presale live at verdant-peach.vercel.app/presale



Verdant

Programmable escrow for anything.

verdant-peach.vercel.app/presale

Presale live · [contact / @handle — TBD]