

CRYP2NOVA

Vibestarter

// WHITEPAPER · DRAFT v1

Vibestarter Whitepaper

Design treatment of the vibecoin funding primitive — mechanism, contract system, trust model.

// CONTENTS

Introduction1. Executive Summary2. The Funding Mismatch3. Vibecoins as a Primitive4. Design Goals5. Time-Released Funding6. Challenge Windows7. Liquidity & LP Lock8. Reputation9. Economics: Worked Examples10. Contract Architecture11. Trust & Safety12. Governance13. Risks & Disclosures14. Appendix

Archived copy · cryp2nova.com

▶ // CONTENTS

Vibestarter Whitepaper

> Authored by Vibestarter Labs. An expansion of the original thesis at vibestarter.xyz/thesis into a full treatment of the funding primitive, its mechanism design, and the contract system that implements it.

Audience: mixed. Layered structure — executive summary for skimmers, mechanism and economics sections for technical readers, appendices for spec-level detail.

Scope: the funding primitive itself, treated as protocol design. Vibestarter is the reference implementation, but the design choices are presented as portable. Per-raise parameters — token supply, targets, and allocation — are set by each individual raise and are out of scope here.

Language: US-cautious throughout. "Backer," "contribution," "tranche release," "secondary liquidity" — not "investor," "yield," or "returns."

Sections

#	File	Section	Status
1	`01-executive-summary.md`	Executive summary	**Draft v1**
2	`02-funding-mismatch.md`	The funding mismatch	**Draft v1**
3	`03-vibecoins-as-primitive.md`	Vibecoins as a funding primitive	**Draft v1**
4	`04-design-goals.md`	Design goals & failure modes we're avoiding	**Draft v1**
5	`05-time-released-funding.md`	Time-released funding	**Draft v1**
6	`06-challenge-windows.md`	Challenge windows	**Draft v1**
7	`07-liquidity-lp-lock.md`	Liquidity and the indefinite LP lock	**Draft v1**
8	`08-reputation.md`	Reputation: Ethos and on-chain signals	**Draft v1**
9	`09-economics-examples.md`	Economics: worked examples	**Draft v1**
10	`10-contract-architecture.md`	Contract architecture overview	**Draft v1**
11	`11-trust-safety.md`	Trust and safety	**Draft v1**
12	`12-governance.md`	Governance and decentralization roadmap	**Draft v1**
13	`13-risks-disclosures.md`	Risks and disclosures	**Draft v1**
14	`14-appendix.md`	Appendix: glossary, parameters, references	**Draft v1**

Reading order

The paper is structured for both linear reading and section-by-section reference.

Archived copy · cryp2nova.com

- **Linear read:** §1 → §14 in order. Roughly 15 pages of prose.
- **Skim read:** §1 only, then dip into specific sections of interest via the table in §1's "How to read the rest of this paper."
- **Mechanism only:** §4 through §9.
- **System / trust review:** §10 through §13.

Current version: v1.0 draft (all 14 sections drafted) **Last updated:** 2026-05-24

1. Executive Summary

> A funding primitive for vibecoded projects. Capital releases over time. Backers can pause it. Liquidity is permanent. The protocol is honest about what is and is not on-chain.

The thesis, in five claims

1. **The cost of building software has collapsed.** AI coding agents have crossed a threshold where a non-technical founder with clear product vision can ship a working MVP in days. The original thesis at vibestarter.xyz/thesis makes the full argument.
2. **The cost of funding software has not collapsed.** Venture capital, ICOs, grant programs, and reward-based crowdfunding all fail for distinct reasons — gatekeeping, lack of accountability, illiquidity, or absence of upside for backers. None are adequate for the population of builders who can now ship.
3. **Vibecoins — tokens issued at the moment a vibecoded project launches — are the funding primitive that matches the new execution speed.** The token launch and the project launch are the same event.
4. **A vibecoin is only useful if the funding mechanism enforces continued accountability.** A token issued at launch with no recall mechanism degenerates into an ICO. The mechanism this paper describes is the minimum structure required to prevent that degeneration.
5. **Vibestarter is one implementation of that mechanism.** The design principles are intended to be portable. The implementation runs on Base.

The mechanism, in five primitives

Each primitive maps to a design goal introduced in Section 4 and is specified in detail in the section indicated.

Primitive	Parameter	Section
Time-released funding. Capital releases on a fixed schedule. No milestones, no approvals on the success path.	10% at finalization + 15% every 30 days × 6 tranches = 6 months total	§5

Archived copy · cryp2nova.com

Primitive	Parameter	Section
Challenge windows. Each tranche request opens a 72-hour window during which token holders can pause the schedule.	72-hour window; graduated holder thresholds (0.25 / 0.50 / 1.00% of supply); 20% slash on rejected challenges; 7-day cooldown	§6
Indefinite LP lock. 15% of every raise creates an Aerodrome liquidity pool; the LP receipt is locked forever in a per-campaign fee claimer with no withdraw function.	Permanent. No unlock event. Trading fees are captured, not burned.	§7
Reputation as a signal, not a gate. Anyone can back any raise — no reputation minimum. Launch admission is curated in the current phase against a published rubric (§12.1), not by reputation scores. Vibestarter surfaces each founder's and backer's reputation so participants can judge for themselves.	Ethos scores + on-chain history	§8
Two-tier admin separation. Master admin (Gnosis Safe multi-sig, M-1) controls infrastructure, fees, rescue, and treasury powers. Operations/escrow admin adjudicates challenges + freezes and cannot extract user funds.	Master = M-1 Safe (2-of-2 today, moving to 2-of-3). Operations = M-3, a revocable single-key hot EOA.	§11

What the contract guarantees, and what it does not

The protocol's contract guarantees are:

- The tranche schedule is encoded in `VibesTranchEscrow`. No party can accelerate, delay, or alter it.
- The challenge slash percentage, window length, and cooldown are encoded. The operator cannot adjust them per-raise.
- The LP receipt is locked in a soulbound per-campaign fee claimer at finalization. No party, including the operator, can recover it.
- Refund paths (contributor refund, holder refund, excess refund) are encoded. The operator cannot redirect them.
- The master admin's rescue capability is bounded by deposit reserves and active-claim checks. The operator cannot drain user escrow.

The protocol's centralized surfaces — surfaces that require continued operator involvement — are:

- **Challenge adjudication.** The operations admin decides whether to uphold, reject, or let expire each challenge. The published challenge standards constrain how. (§6.6, §11, §12.2)
- **Refund merkle root publication.** After a campaign is frozen, the admin publishes the snapshot root. The 24-hour commit-reveal delay lets holders verify the root before claims open. (§6.4, §11.3)

Archived copy · cryp2nova.com

- **Reputation display.** Which reputation signals (Ethos, on-chain history) are surfaced for founders and backers is curated off-chain by the platform; the underlying scores are externally sourced and independently verifiable. (§8)
- **Parameter and infrastructure changes.** Platform fee adjustments, escrow factory upgrades, and operations admin appointment require master admin action. (§11.1, §12.2)

Section 12 covers the path on which the centralized surfaces decentralize.

What this paper does not claim

The protocol does not claim:

- That backers will profit. Tokens are exposed to standard market risk. (§13.3)
- That fraud is impossible. A dishonest founder can claim early tranches before a challenge is upheld. The protocol minimizes this loss; it does not eliminate it. (§13.4)
- That displayed reputation guarantees a founder is trustworthy. Ethos and on-chain signals inform judgment; they do not guarantee a founder will deliver. (§13.7)
- That the audits guarantee bug-free contracts. One external audit (ZXVC LLC) plus internal review cycles have been completed; all High findings were remediated, while specific centralization findings (no admin timelocks on the treasury burn / infra setters) are accepted/deferred and disclosed as residual risk. Unknown bugs may still exist. (§13.1)
- That the LP lock guarantees value. The pool exists permanently. Whether anyone trades against it depends on what the founder builds. (§7.5)

Status

- **Live on Base mainnet** (chain ID 8453). Contract addresses are queryable on-chain from `VibesRegistry` events.
- **Two audit cycles complete.** April 2026 and May 2026. Remediation summary at app.vibestarter.xyz/audit.
- **Reference frontend** at app.vibestarter.xyz. Any client conforming to the on-chain interfaces can interact with the protocol.

How to read the rest of this paper

If you want...	Read
The thesis it is built on	§2, §3 (or the original at vibestarter.xyz/thesis)
Why the mechanism is shaped the way it is	§4
How the mechanism works in detail	§5, §6, §7, §8
Worked examples with real numbers	§9
The contract architecture	§10

Archived copy · cryp2nova.com

If you want...	Read
The operator surfaces and the audit history	§11
The decentralization roadmap	§12
What can go wrong	§13
Glossary, parameters, references	§14

The whitepaper synthesizes the protocol's behavior; the deployed contracts are authoritative for specific implementation details.

2. The Funding Mismatch

> Execution collapsed. Funding did not. The whitepaper begins where the thesis ended — with the observation that the two halves of the early-stage stack are now operating at different speeds.

The full case for vibecoins as a category lives in the original thesis at vibestarter.xyz/thesis. This section compresses that argument to the load-bearing claims, in service of setting up the mechanism design problem that the rest of this paper solves. A reader who wants the full rhetorical case should read the thesis directly.

2.1 What changed in execution

The capability of AI coding agents crossed a quality threshold during 2025–2026 that materially altered the economics of starting a software project. Models with long-horizon reasoning and context windows that can hold entire codebases in memory can now act as a project's first technical lead. A founder with a clear product vision can ship a real, functional MVP in days rather than months.

This is not a claim that AI eliminates the need for technical excellence. It is the narrower claim that **the gating role of the technical co-founder at day zero has been removed.** Deep engineering expertise still matters at scale, when optimizing, when building moats. It is no longer required to prove an idea works.

The consequence: the population of people who can ship an MVP without first finding (and convincing) a technical co-founder has expanded by orders of magnitude. Surface area for software experimentation increases. Some of the ideas that previously died in the graveyard now ship.

2.2 What did not change in funding

The funding rails available to early-stage software did not move at the same speed.

Archived copy · cryp2nova.com

Venture capital still operates on multi-month timelines, requires accredited capital, and uses equity structures that exclude non-investors. Cap table management, term sheets, and partner judgment are necessary functions when investing $5M in a single bet — they are structural overhead when funding a $50k experiment that took a weekend to build.

Grant programs route capital to builders without the all-or-nothing dynamic of equity rounds, but they are discretionary by design. Decisions take months. Recipients hold no liquid claim on what they helped create.

ICOs and their successors demonstrated that you can route significant capital to anonymous teams in days, but they also demonstrated what happens when capital arrives without continued accountability. The pattern was not a fluke of one cycle — it is what happens when funding mechanism and execution mechanism are misaligned in a specific direction.

Memecoin launchpads proved that coordinating speculative capital is now trivially cheap. They also demonstrated that speculation disconnected from any project being built becomes its own end. Builders trying to use these rails are drowned out by tokens optimized purely for liquidity events.

Reward-based crowdfunding (Kickstarter and its descendants) solved permissionless access but left backers with no claim on success, no governance, and no recourse if the project quietly dies after the funding window.

None of these are adequate for a population of builders who can ship in days, on budgets in the tens of thousands, with potential audiences in the millions. The shape of capital that fits is faster, smaller, more numerous, and more permissioning-free than any of the above.

2.3 The mismatch as a design problem

The thesis argues that this mismatch is *the* relevant design problem for funding infrastructure in this period. The whitepaper takes that argument as setup.

What follows from accepting it: a new funding primitive must be **fast** (days, not quarters), **permissionless** (anyone can launch, anyone can back), **denominated in liquid units** (a transferable claim, not an illiquid equity stake), **accountable** (continued performance is required, not assumed), and **honest about its operator surfaces** (which, as Section 12 makes clear, every protocol has).

The rest of this paper is the specification of one such primitive. Section 3 introduces the term we use for it (the *vibecoin*) and the broader category. Sections 4 through 11 specify the mechanism. Sections 12 and 13 cover the operator surfaces and the residual risks. None of this is the only possible specification — but the design constraints are sharp enough that most reasonable specifications would converge on similar choices, and we make those choices explicit in Section 4.

The thesis ended with a question: *how quickly do new funding norms form?* This paper is one attempt to answer it by building one.

Archived copy · cryp2nova.com

3. Vibecoins as a Funding Primitive

> A vibecoin is a token associated with a vibecoded project, designed to function as both community coordination and funding mechanism from day zero. The category is defined by what it is *not* as much as by what it is.

This section establishes the term, distinguishes it from adjacent categories that share surface features, and frames the design problem that the rest of the paper solves. As with Section 2, this compresses material from the original thesis — readers wanting the full argument should consult vibestarter.xyz/thesis.

3.1 The definition

A **vibecoin** is a token issued at the launch of a vibecoded project, where the token serves as both:

1. **A coordination instrument.** Holders are the project's earliest community. They have a transferable claim on the project's success and on its on-chain governance surfaces (where those exist).
2. **The funding instrument.** The capital that the project receives to build comes from backers contributing in exchange for vibecoins.

The two functions are inseparable. A token that does only the first (community without funding) is a community token. A token that does only the second (funding without community) is a security in most jurisdictions. The vibecoin sits between them: a transferable claim, tied to a specific project being built, issued at the moment of the project's founding.

3.2 What vibecoins are not

The category is most usefully defined against its neighbors.

A vibecoin is not an AI vapourware token. AI tokens issued by projects with no working software, no shipping cadence, and no demonstrable connection between the token's claimed utility and any product the team is building are the failure case. They take the surface features (token, AI branding, white-paper) and discard the load-bearing structure (something is actually being built).

A vibecoin is not a memecoin. Memecoins are coordination instruments without product. The launching team has no obligation to build, no schedule against which to be measured, and no accountability mechanism. Memecoins serve a legitimate function (purely speculative coordination is its own primitive) but are not what this paper is describing.

A vibecoin is not a launchpad token. Tokens launched on permissionless launchpads (pump.fun, bags.fm, similar) inherit the launchpad's mechanism — usually a bonding curve, often a fixed supply,

Archived copy · cryp2nova.com

sometimes a graduation event. They are not tied to a specific project's build timeline and are not gated by any accountability mechanism. Many launchpad tokens are memecoins; some happen to be associated with builders. The launchpad does not enforce the association.

A vibecoin is not a post-launch token. When a crypto project ships a working protocol and then tokenizes — typically to provide exit liquidity for early backers — the resulting token is an exit instrument, not a funding instrument. The product was built without the token. The token's launch is a financing event for the team that already built the product. This is the standard model for serious crypto projects today and produces good outcomes for the teams that use it, but it does not address the population of builders who need capital *before* they have built anything.

A vibecoin is not an equity stake. Backers are not buying shares in a company. There is no cap table, no preferred / common distinction, no liquidation preference, no voting rights at a board level. The vibecoin is a claim on the project's specific token-denominated success, with the contract-defined refund paths described in Section 6.4 as the closest analog to a downside protection.

3.3 What makes vibecoins specifically suited

The defining feature of the vibecoin is that the **token launch and the project launch are the same event**. This is enabled by the collapse in execution cost (Section 2). It is what distinguishes vibecoins from the prior categories.

When execution was expensive, no rational party would issue a token before the product existed — the token would be priced against vapor. With execution costs near zero, the token can be issued at *the same time as the MVP is being shipped*. The first backers are participating in something that is being built in front of them, not something that was pitched in a slide deck six months ago.

This temporal alignment enables three properties that prior categories cannot have:

1. **The funding window matches the building window.** A vibecoded MVP ships in days. The funding window can also close in days. The mismatch between weeks-to-build and quarters-to-fund disappears.
2. **Backers select projects by their actual emerging shape.** A backer in a vibecoin raise can see the project's repo, the current build state, the founder's recent commits. They are not pricing a roadmap; they are pricing what already exists plus the founder's demonstrated velocity.
3. **The token's value is empirically grounded.** The token is tied to a specific project that is observably being worked on. A backer can verify that the founder is building. The token's existence is contingent on the project's existence in a way that detached tokens are not.

These properties are *necessary but not sufficient*. A token issued at MVP launch can still fail in all the ways Section 4 describes — front-loaded capital, no accountability, no holder recourse, no permanent liquidity. The properties above describe what the vibecoin *can be*. The mechanism described in

Archived copy · cryp2nova.com

Sections 5 through 8 is what makes a vibecoin *actually work* — what prevents it from degenerating into one of the failure modes the thesis was reacting against.

3.4 The design problem

The remaining sections of this paper are the answer to a specific design question:

> Given that the vibecoin category is now economically viable (Section 2), and given that none of the existing funding rails (Section 2.2) or existing token-launch mechanisms (Section 3.2) are adequate, what is the narrowest set of on-chain mechanics that:
>
> 1. Permits anyone to issue a vibecoin without operator approval (G3);
> 2. Routes capital to the founder over time, contingent on continued execution (G1, G2);
> 3. Gives backers a transferable claim that retains meaning in success, stall, and failure paths (G4);
> 4. Provides permanent secondary-market liquidity (G5);
> 5. Includes a credible adversarial action when something is going wrong (G6);
> 6. Does not depend on its operator's continued cooperation for the contract-level guarantees to hold (G7);
> 7. Is honest about the operator surfaces that do remain (Section 12)?

The design goals are introduced formally in Section 4. The mechanism is specified in Sections 5 through 8. The system that ties them together is in Sections 10 through 12. The unsolved residual is in Section 13.

The vibecoin is the *category*. Vibestarter is one specific implementation of the category. The rest of this paper is Vibestarter's specific implementation — but the design principles are intended to be portable to any other implementation that wants to address the same problem.

4. Design Goals and Failure Modes We Are Not Willing to Inherit

> Every existing path from "early idea" to "funded product" has failure modes that get worse as the path gets more accessible. This section names them, then derives the constraints that the rest of the paper satisfies.

The mechanism described in Sections 5 through 11 is not a feature list. It is the narrowest design that survives once each of the failure modes below is ruled out. We are deliberately explicit about which prior model produced which lesson, so that the design choices that follow read as forced moves rather than preferences.

Archived copy · cryp2nova.com

4.1 Failure modes

The ICO collapse: capital arrives before accountability

The 2017–2018 token sale wave proved that you can route billions of dollars to anonymous teams in days. It also proved that founders who receive the full proceeds at launch have no remaining economic reason to ship. The failure rate of that cycle was not a fluke. It was the mechanism working exactly as designed. Once capital has been delivered, the contract between founder and backer is over.

Lesson: capital cannot arrive in a single lump. The founder must remain economically dependent on continued execution for the duration of the build.

Milestone gates: gameable, slow, centralizing

The reaction to ICO failure was to gate capital behind milestones. This works inside venture rounds, where partners apply judgment, but it fails as a permissionless primitive. Milestones become performances: a founder ships a "v2 mobile launch" because the contract pays for it, not because users wanted it. Whoever judges the milestone — a multisig, a DAO, a foundation — becomes a single point of capture, dispute, and delay. Disputes are slow. Delays compound. Every milestone-gated system we have studied either centralizes the judgment or collapses under the weight of arbitration.

Lesson: the schedule must be objective and adversarially un-gameable. Time is the only variable a founder cannot perform their way around.

Memecoin launchpads: speculation decoupled from building

Platforms in the pump.fun lineage proved that coordinating speculative capital is now trivially cheap. The token is rarely tied to anything being built. Most launches are speculation vehicles whose entire lifecycle is exit-driven, and builders trying to use these rails get drowned out by tokens optimized purely for liquidity events. Speculation is not the problem — speculation *disconnected from any project being built* is.

Lesson: the funding contract must enforce that capital is allocated to a project, not just a ticker, and that the founder remains the on-chain counterparty over a multi-month period.

Venture capital: gatekept, all-or-nothing, slow

Traditional venture funding requires accredited capital, multi-month timelines, equity structures that exclude non-investors, and centralized judgment about which ideas deserve to exist. When vibecoded MVPs ship in days, six-month rounds are not a feature; they are a structural mismatch. The cost of being wrong about which ideas to fund used to justify the gatekeeping. With execution costs near zero, the gatekeeping outlives its justification.

Archived copy · cryp2nova.com

Lesson: funding must be permissionless to access, denominated in a unit anyone can hold, and clear in days — not quarters.

Kickstarter and the no-upside backer

Reward-based crowdfunding solved permissionless access but left backers with no claim on success, no governance, and no recourse if the project quietly dies after the funding window. The most a backer can do is post a comment. This is fine for board games. It is not adequate for software intended to compound.

Lesson: backers must hold a transferable claim on the project's outcome that retains its meaning whether the project ships, fails, or stalls. They must also have a credible action — not just a comment thread — when something goes wrong.

Grant programs: discretionary and illiquid

Ecosystem grants and retroactive funding programs route capital to builders without the all-or-nothing dynamic of an equity round. But they are discretionary by design. A small committee decides; decisions take months; recipients hold no liquid claim on what they helped create, and supporters hold no claim at all. Grants are an important complement to other funding rails. They are not a substitute for one.

Lesson: funding decisions should be aggregated from many small bets, settled in a transferable instrument, and closed in days.

4.2 Design goals

The failure modes above do not leave much design space. The goals below are what survives once each one is ruled out. They are the contract this paper has with the reader: every mechanism described in Sections 5 through 11 satisfies one or more of them. If a mechanism does not map to a goal, it should not exist.

G1 — Time, not milestones

The capital release schedule is determined by elapsed time alone. No discretionary approval exists on the success path. No party can speed up or slow down the schedule by performing a milestone. The schedule in Vibestarter is **10% at finalization, then 15% every thirty days for six months.** It is encoded in the escrow contract and is not adjustable per-raise. *(Section 5.)*

G2 — Continued accountability

No tranche after the kickstart release pays out without a window during which token holders can pause the schedule. A founder who has abandoned a project does not collect the back half by

Archived copy · cryp2nova.com

waiting. The window is **72 hours** and opens when the founder requests a tranche. *(Section 6.)*

G3 — Permissionless on both sides (end-state; launch side currently curated)

Anyone can back a raise; there is no accredited-backer gate and no minimum reputation. Launching is the stated end-state of this goal but is **curated in the current phase**: mainnet raise launches require a founder application reviewed against the published rubric (Section 12.1 / the Launch Admission & Moderation Policy). Reputation systems gate features — they do not gate backing access. *(Sections 8, 12.)*

G4 — Backer upside is a transferable claim

Backers receive a token that represents their share of the project, transferable on secondary markets. The claim retains meaning in each of three outcomes:

- **Ship:** the token is the holder's link to the project's secondary market and to any community rights the founder defines.
- **Stall:** the holder can join a successful challenge to pause the schedule; remaining escrow becomes claimable pro-rata against burned tokens.
- **Fail to fund:** contributions are fully refundable in the original asset.

(Mechanics in Sections 5, 6, and 9.)

G5 — Liquidity is permanent, not promotional

The liquidity pool created at finalization holds **15% of the raised capital paired against tokens** and is locked **indefinitely**. The LP receipt is held in a soulbound per-campaign fee claimer with no withdraw function. It is not a marketing tool that can be pulled. This is a load-bearing choice for secondary-market integrity and is described in Section 7.

G6 — Adversarial pressure is part of the design

A token holder who believes a project is dead or fraudulent has a contract-level action: the **challenge**. The action is costly enough to discourage griefing — a rejected challenger forfeits **20% of staked tokens** to the burn address — and decisive enough to matter: an upheld challenge **freezes the campaign** and opens a holder-refund path. The challenge stake threshold scales with tranche number (0.25% / 0.50% / 1.00% of supply) and challengers are subject to a **7-day cooldown** after raising one. *(Section 6.)*

G7 — The protocol does not depend on its operator

Every load-bearing decision — release schedule, challenge window length, refund path, LP lock — is encoded in the contract. The operator can be replaced, captured, or disappear without the schedule changing for an in-flight raise. Centralized surfaces remain (challenge arbitration, refund-merkle

Archived copy · cryp2nova.com

publication, parameter governance) and are described honestly in Section 12, along with the path on which each one decentralizes.

4.3 What is intentionally not a goal

It is as useful to name what we are not optimizing for.

- **We are not optimizing for the largest possible raises.** A primitive that works for $50k-pre-MVP ideas is more valuable than one that works for $50M Series-B-replacements. The former is structurally underserved.
- **We are not optimizing for backer "yield."** A token whose value is the project's success is not a yield instrument. Holders accept the same outcome distribution as the project itself.
- **We are not optimizing for full decentralization on day one.** Some surfaces — challenge arbitration, in particular — start with operator involvement and decentralize on a schedule we make public. Pretending otherwise would be the same dishonesty we are reacting against in 4.1.
- **We are not building a memecoin launchpad.** A founder who is not building anything has nothing to claim tranches against and no defense against a challenge. The mechanism is hostile to that use case by construction.

The rest of this paper describes how each goal is implemented and what is still imperfect about each implementation.

5. Time-Released Funding

> Capital releases on a fixed time schedule. No milestone reviews. No approvals on the success path. Time is the only variable a founder cannot perform around.

This section describes how capital flows from a successful raise to the founder over the six months after funding closes. It is the implementation of design goal G1 (*time, not milestones*) and the structural counterweight to G2 (*continued accountability*) which is implemented by the challenge window described in Section 6.

5.1 The schedule

At the moment a raise finalizes, the raised capital splits:

Slice	Destination	Released when
15%	Liquidity pool — paired against tokens, LP receipt locked in a per-campaign fee claimer	At finalization, permanent (Section 7)

Archived copy · cryp2nova.com

Slice	Destination	Released when
85%	Escrow — `VibesTranchEscrow` clone for this raise	In seven tranches over six months

The escrowed 85% releases on the following schedule:

Tranche	Trigger time	Tranche size	Cumulative
Kickstart (T0)	Immediately on finalization	10% of escrow	10%
T1	+30 days	15% of escrow	25%
T2	+60 days	15% of escrow	40%
T3	+90 days	15% of escrow	55%
T4	+120 days	15% of escrow	70%
T5	+150 days	15% of escrow	85%
T6	+180 days	15% of escrow	100%

Each tranche is subject to a **2.5% platform fee**, deducted from the tranche at claim time. The founder receives 97.5% of each tranche; the platform receives 2.5%.

The schedule constants are encoded in `VibesTranchEscrow`:

- `KICKSTART_BPS = 1000` (10%)
- `MONTHLY_BPS = 1500` (15%)
- `NUM_MONTHLY_TRANCHES = 6`
- `TRANCHE_DURATION = 30 days`
- `PLATFORM_FEE_BPS = 250` (2.5%)

These are protocol-level constants. They are not configurable per-raise. A founder who wants different terms cannot get them on Vibestarter — they would have to build on a different platform.

5.2 Why these specific numbers

The 10/15/6 schedule is the answer to three constraints solved simultaneously.

The kickstart has to be enough to start. A founder receiving zero capital at finalization cannot pay for servers, contractors, or their own time during the first month while waiting for tranche one. Ten percent of escrow is enough to fund the first thirty days of a vibecoded project's burn for the vast majority of raises we have modeled. It is also small enough that a founder who absconds with only the kickstart has not done meaningful damage to the backer pool.

Tranche size has to be uniform after the kickstart. Front-loading subsequent tranches (e.g., 30% then declining) creates the same incentive problem as the ICO collapse on a slightly longer timeline. Back-loading (e.g., 5% then increasing) creates a perverse pressure to ship something dramatic right

Archived copy · cryp2nova.com

before the largest release, which is the milestone problem with extra steps. Equal tranches keep the founder's marginal incentive to continue building constant across the six months.

Six months is the longest schedule that does not become its own problem. Long schedules create stale-token problems (backers forget the project exists, secondary market liquidity dries up before final release), founder cash-flow problems (paying contractors who expect payment in fiat from a tranche schedule denominated in ETH), and operator-attention problems (challenge windows that span a year are harder to monitor than ones that span six months). Six months is empirically the point at which all three of these begin to bite.

The platform fee at 2.5% per tranche compounds to a total platform take of 2.5% of the escrowed amount — equivalent to **2.125% of the total raise** after the LP split. This is deliberately lower than ICO-era platform fees (commonly 5–10%) and lower than centralized crowdfunding (Kickstarter is ~5% + payment processing). The fee is taken at each tranche, not upfront, which keeps the platform's incentives aligned with the founder's continued performance.

5.3 The tranche request lifecycle

A tranche does not release automatically when its time arrives. The founder must explicitly request it. The full lifecycle of a single tranche:

1. **Time gate elapses.** Block timestamp ≥ tranche release time.
2. **Founder calls `requestTranche(trancheIndex)`.** This opens a **72-hour challenge window** during which token holders may raise a challenge. The challenge window is described in detail in Section 6.
3. **Window resolves.** One of three outcomes:
    - *No challenge raised:* after 72 hours, anyone can advance the state and the founder can claim.
    - *Challenge raised and rejected:* challenger is slashed, founder can claim.
    - *Challenge raised and upheld:* campaign is frozen, this tranche and all subsequent tranches are blocked, refund path opens (Section 6.4).
4. **Founder calls `claimTranche(trancheIndex)`.** The contract transfers `trancheAmount × 97.5%` to the founder's address and `trancheAmount × 2.5%` to the platform wallet.

The founder cannot batch tranches. Each tranche is requested, challenged-or-not, and claimed individually. This is deliberate: each tranche is a fresh opportunity for backers to act on new information.

5.4 What this does *not* do

It is worth being explicit about what time-based release does not solve.

It does not protect against founders who are dishonest from the start. A founder who launches a raise with no intention of building can still receive the kickstart and the first tranche or two before

Archived copy · cryp2nova.com

challenges accumulate. The protocol minimizes the size of this loss — it does not eliminate it. The reputation system (Section 8) and the founder deposit (refunded only on successful finalization) are the complementary defenses.

It does not enforce that the founder is actually building. No contract can. The challenge window is a *recall mechanism*, not a *progress oracle*. If holders are inattentive, a non-building founder will collect their full schedule. This is the price of refusing to gate capital behind centralized milestone judgment.

It does not adjust for project-type variance. A research project that needs 70% of capital in month one for compute reservations is poorly served by 10% kickstart + monthly tranches. The protocol's answer is that those projects should raise differently — through grants, through traditional venture, through pre-commit campaigns. Vibestarter is built for a specific shape of project (early-stage software, sustained burn, six-month build horizon to first meaningful milestone). The schedule is opinionated on that basis.

5.5 Pausing, force-refund, and the limits of automation

Two operator-controlled escape hatches exist, both restricted to the operations admin role described in Section 11.

`pauseCampaign()` / `resumeCampaign()` halt tranche releases during the active raise window. This is intended for cases where a known issue (a contract advisory, a suspected exploit) needs to be triaged before further capital flow.

`forceRefundDuringRaise()` kills an active raise before finalization and returns all contributed ETH to backers. This is the escape hatch for raises that need to be terminated before they finalize — for example, a discovered fraud pattern, a smart contract issue requiring redeployment, or a fundamental compliance problem.

Both functions affect contributors only during the active raise window. After finalization, neither is available — at that point the only path that returns capital to holders is a successful challenge followed by a holder refund (Section 6.4). This boundary is intentional: once capital has been routed to escrow and the LP has been created, the protocol treats the raise as committed, and any unwinding requires the adversarial challenge path rather than an operator action.

The honest framing: these are operator powers, not contract-level guarantees. They exist because no realistic launch protocol can ship without them, and because pretending otherwise would be the kind of dishonesty Section 4 rules out. Section 12 covers how each of them decentralizes over time.

6. Challenge Windows

Archived copy · cryp2nova.com

> A 72-hour window opens whenever a founder requests a tranche. Token holders can pause the schedule. The cost of being wrong is high enough to deter griefing; the cost of being right is low enough to make it worth doing.

The challenge mechanism is the implementation of design goal G2 (*continued accountability*) and G6 (*adversarial pressure is part of the design*). It is the counterweight to Section 5's time-based release: without it, time-only release degenerates into a long-form ICO. With it, the schedule becomes contestable in a structured way.

6.1 The window

Each tranche request opens a **72-hour challenge window** before the tranche becomes claimable. The window is encoded as `CHALLENGE_WINDOW = 72 hours` in `VibesTranchEscrow`.

72 hours is a deliberate choice. Shorter windows (24 hours) systematically advantage the founder: holders in different timezones, with day jobs, or simply not watching their notifications may miss the window. Longer windows (one week) systematically advantage griefers and stall projects by accumulating a stale-data risk during slow review. Three days covers a full weekend and a working day on either side, which empirically captures the long tail of holder attention without indefinitely deferring the schedule.

The kickstart tranche (T0) does not have a challenge window. It releases immediately on finalization. This is a concession to the *kickstart* design constraint described in Section 5.2 — a founder cannot start work if their first capital is gated by a 72-hour delay. The trade-off is that T0 is 10% of escrow, the smallest tranche.

Every subsequent tranche (T1 through T6) is gated by a fresh challenge window.

6.2 Who can challenge

Any address holding at least a threshold amount of the project token can raise a challenge. The threshold is **graduated** by tranche number:

Tranches	Required holdings (% of supply)
T0–T2 (early)	0.25%
T3–T4 (mid)	0.50%
T5–T6 (late)	1.00%

The graduation reflects information asymmetry. Early in a project's life, holders have very little data on whether the founder is actually building. A low threshold lets more holders raise concerns before too much capital has been released. Later in the schedule, the founder has accumulated a track record (delivered features, missed deadlines, public communication), and the threshold rises to prevent a small fraction of holders from blocking tranches against a clear pattern of execution.

Archived copy · cryp2nova.com

The threshold is checked at challenge time against the challenger's current balance. The full stake is locked in the escrow contract for the duration of the challenge.

A single challenger cannot serially challenge every tranche. Each address is subject to a **7-day cooldown** after raising a challenge (`CHALLENGE_COOLDOWN = 7 days`). Different addresses can independently challenge different tranches, but a single actor cannot grief a project by burning through their token balance one challenge at a time.

6.3 The slash

When a challenge is rejected, the challenger loses **20% of their staked tokens** (`CHALLENGE_SLASH_BPS = 2000`), burned to `0xdead`. The remaining 80% is returned.

The slash exists to make challenges costly. Without it, an actor with a small balance could raise repeated challenges (across many wallets) to grief a founder at zero cost. With it, raising a frivolous challenge has a real, calculable token cost. The 20% number is the result of two constraints:

- It must be large enough to make griefing uneconomic. At 20%, a griefer with 0.25% of supply burns 0.05% of supply on each rejected challenge. Five rejected challenges costs 0.25% of total supply — a meaningful position. This is enough to deter sustained griefing campaigns.
- It must be small enough that a *good-faith but ultimately wrong* challenge is recoverable. A 100% slash would prevent honest holders from ever raising a concern unless they were sure they would be upheld, which defeats the purpose of having a recall mechanism for ambiguous cases.

20% threads this: griefers can't afford to be systematically wrong, but honest concerns aren't punished out of existence.

A *successful* challenge — one that is upheld by the admin review described in Section 6.4 — returns the challenger's full stake. No slash, no reward. Vibestarter does not pay challengers for being correct. Paying for successful challenges would create a meta-game in which prolific challengers (correctly identifying issues, but also farming the bounty) drive up the rate of challenge requests. Returning stake at par keeps the incentive purely defensive: you challenge to protect the value of your remaining position, not to earn a fee.

6.4 Outcomes

A challenge resolves in one of three ways.

Uphold

The admin reviews the challenge and finds it warranted. The contract action is `upholdChallenge()`, which:

Archived copy · cryp2nova.com

- Freezes the campaign. All remaining tranches are permanently blocked.
- Returns the challenger's full stake.
- Captures `frozenEthBalance` (the ETH remaining in escrow at freeze time) and `frozenTotalSupply` (the total token supply at freeze time, optionally excluding admin-specified addresses such as the founder's own holdings).
- Opens the **holder refund** path.

Holder refund works on a merkle-proof basis. The admin publishes a refund merkle root after a 24-hour commit-reveal delay (`MERKLE_ROOT_DELAY`). Token holders then burn tokens to `0xdead` and claim a proportional ETH refund: `(frozenEthBalance × burnedTokens) / frozenTotalSupply`.

The commit-reveal delay exists to give holders a chance to verify the published root before claims start. A 24-hour window after the root commitment allows the holder community to check that no addresses were unfairly excluded from the denominator and no addresses were unfairly included. After 24 hours, claims open.

Reject

The admin reviews the challenge and finds it unwarranted. The contract action is `rejectChallenge()`, which:

- Slashes 20% of the challenger's stake to `0xdead`.
- Returns 80% to the challenger.
- Releases the tranche to the founder via the normal claim path.

Expire

The admin takes no action within the adjudication window — 7 days from filing (`ADJUDICATION_WINDOW`) on raises launched after the PC-09 cut-over; 72 hours on earlier escrows, including $VIBES, whose constants are immutable. Anyone can then call `expireChallengeIfNeeded()`. The challenger's full stake is returned (no slash). The tranche releases to the founder.

This is the *quiet failure* path: a challenge that the admin neither upholds nor explicitly rejects is treated as not warranting a slash, but also not warranting a freeze. In practice this should be rare — the admin is expected to respond to every challenge within the window — but the path exists so that a delinquent admin cannot indefinitely hold tranches hostage by simply not responding.

6.5 Challenge support

Other token holders can call `supportChallenge()` to add their voice to a pending challenge. This is **event-only** — no state change, no stake required beyond holding some non-zero balance of the token. It exists so that the admin reviewing a challenge has signal beyond the one challenger's claim. If twenty holders representing a meaningful fraction of supply support a challenge, that is informationally distinct from a single 0.25% holder acting alone.

Archived copy · cryp2nova.com

Support events are surfaced in the admin's review tooling and are part of the public record of the challenge.

6.6 The honest reality of admin review

The challenge review — up to 7 days from filing on post-PC-09 escrows (72 hours on earlier ones) — is performed by the **operations admin**, an account separate from the platform's master multi-sig (described in detail in Section 11). The operations admin can uphold or reject a challenge, but cannot move user funds to arbitrary destinations and cannot release escrow ahead of schedule.

The set of decisions the admin is empowered to make is small and adversarially constrained:

- Uphold: triggers contract-defined freeze and refund path. The admin cannot redirect the refund.
- Reject: triggers contract-defined slash. The admin cannot adjust the slash percentage.
- Take no action: tranche releases automatically after window expiry.

What the admin cannot do: extract escrow, alter the schedule, change a tranche size, exclude themselves from a refund denominator without that exclusion being publicly visible. The `_excludeAddresses` parameter on `upholdChallenge` is the highest-trust surface — it controls which addresses are excluded from the refund denominator, and a malicious admin could in principle manipulate it to over-pay specific holders. The challenge standards (Section 6.6) define how this parameter is used, and Section 11 covers the cosigner separation that constrains it on the production multi-sig.

This is the centralized surface that *we will not pretend is not centralized*. It exists because contract code cannot read the world: a contract cannot determine whether a founder has actually disappeared, whether claimed progress is real, whether a "missed milestone" is a missed deadline or a strategic pivot. A human (or a multi-sig) has to make that call, in the same way that arbitration exists in every other contractual system humans have built. Section 12 covers the path on which this surface decentralizes — challenge standards documents, multi-sig adjudication panels, and eventually community-elected reviewers.

The contract guarantees the *consequences* of the admin's decision (slash percentages, freeze mechanics, refund proportionality). It does not guarantee the decision itself. That is the boundary, and we mark it clearly rather than hiding it.

7. Liquidity and the Indefinite LP Lock

> 15% of every raise creates a liquidity pool that is locked forever. The LP receipt is held by a per-campaign fee claimer with no withdraw function — no party can ever pull it. There is no unlock event. There is no governance vote that releases it. It is permanent.

Archived copy · cryp2nova.com

This section describes the liquidity primitive that pairs every Vibestarter raise. It is the implementation of design goal G5 (*liquidity is permanent, not promotional*) and is also a load-bearing piece of G4 (*backer upside is a transferable claim*) — a claim is only meaningfully transferable if there is somewhere to transfer it.

7.1 What gets created

At finalization, the protocol creates an **Aerodrome volatile-pair liquidity pool** on Base, pairing:

- **15% of the raised ETH** (taken from the contributor pool before escrow)
- **Project tokens sized to open the pool at the backer entry price minus 5%.** Up to 15% of supply is reserved for this (taken from the project's token allocation); the locker pairs only the amount needed to hit that opening price and **returns any unused portion to backers**. The token and ETH sides are therefore no longer a symmetric 15%/15%.

The LP receipt token — the on-chain claim on that pool's liquidity — is transferred to a **per-campaign `VibesLPFeeClaimer`** clone. The claimer is *soulbound*: it has no transfer, withdraw, rescue, pause, or admin function, so the LP receipt can never leave it. No party, including Vibestarter, can move the position or reclaim the liquidity. The lock is irreversible and recorded on-chain.

This is implemented in `VibesLPLockerV2.createAndLockLP()` (the discount-aware locker active on mainnet since the 2026-06-14 cut-over; `VibesLPLockerV2` is byte-identical to `VibesLPLocker` except for this discount-aware sizing). It is invoked exactly once per raise, at the moment of finalization.

The pool itself remains active. Trades against it continue to happen. Unlike a burned LP — where the trading fees the position earns would be stranded forever — the claimer **captures those fees**: the ETH-side fees route to the platform, and the project-token-side fees to the project's treasury (or are burned if the raise has no treasury). Settlement is permissionless, so anyone can trigger it, and the principal liquidity never moves. The pool is a permanent secondary market that nobody controls.

7.2 Why indefinite, not time-locked

Most token launch platforms time-lock LP for a period (six months, one year, two years). Vibestarter does not. The LP is locked in a claimer from which it cannot be recovered, at any time, by any party.

Three reasons.

A time-locked LP is a future event that backers must price in. A holder of project tokens who can see "LP unlocks in 73 days" knows there is a future moment at which the founder may withdraw all paired liquidity and convert it to ETH. This creates a predictable cliff in secondary market depth, which rational holders price into the token before the cliff. The closer the unlock approaches, the worse the depth. The "lock" provides a window of confidence that systematically narrows.

Archived copy · cryp2nova.com

A permanently locked LP has no such future event. Depth at month one looks identical to depth at month thirty-six. There is no countdown.

Time locks are a marketing tool. Permanence is a property. A founder who can describe their LP as "locked for two years" is doing the launchpad equivalent of describing a vesting schedule as a commitment. It is a commitment with a known expiry. The interesting question is what happens after the expiry, and the honest answer for most platforms is *whatever the founder wants*. Permanence forecloses the question.

It changes founder selection. Founders unwilling to permanently abandon their paired LP self-select away from the platform. This is the desired effect. A founder who needs the option to pull liquidity at month thirteen is signaling a relationship to the project's secondary market that is incompatible with what backers are buying when they receive a transferable token.

The cost of this choice is that some legitimate use cases do not fit. A founder running a treasury-management strategy that involves rebalancing paired liquidity has no path on Vibestarter. They should use a different launchpad. The opinionation is the point.

7.3 What the holder actually gets

A backer who contributes ETH to a Vibestarter raise receives tokens denominated against three sources of value:

1. **Pro-rata claim on the project's success.** If the token gains value in the open market, the holder's position appreciates. Standard secondary-market exposure.
2. **Pro-rata claim on remaining escrow in the event of a successful challenge.** Described in Section 6.4. The holder can burn tokens for ETH at the freeze price.
3. **Pro-rata exit liquidity against the permanent pool.** At any time, the holder can sell into the Aerodrome pair created at finalization. The depth is whatever the trading activity since launch has accumulated to.

The third leg is what the indefinite lock guarantees. A holder selling at month nine of a raise faces the same on-chain depth profile as a holder selling at month two. The protocol does not promise that depth will be high — that depends on trading interest, founder execution, and broader market conditions. The protocol promises only that the depth will not be artificially yanked by the founder reclaiming the paired position.

7.4 LP creation failure modes

Locking an LP is not always trivially successful. The Aerodrome router can fail for reasons unrelated to the protocol (gas pricing, ordering, slippage at the moment of finalization). The contract handles this by routing failed LP creations to a **rescue mapping**, from which the operations admin can manually resolve and re-create the LP.

Archived copy · cryp2nova.com

When a rescue happens, the ETH and tokens intended for LP creation are held by the LP locker contract until the admin completes the manual resolution. The admin calls `resolveRescuedFunds(campaign, to)` to forward the rescued ETH and tokens to a chosen recipient, who then manually creates the LP on Aerodrome and locks the receipt in a `VibesLPFeeClaimer` (or, as a legacy fallback, burns it to `0xdead`). The admin then proves the lock by calling `recordManualLPLock(...)`, which requires on-chain evidence that the claimer — or the dead address, for a legacy burn — holds at least the locked LP amount before the rescued state transitions to verified-locked. The May 2026 audit (finding VIB-09) hardened this path further: the supplied claimer must be a genuine `VibesLPFeeClaimer` clone for the campaign, so a non-canonical or withdrawable holder cannot be passed off as a permanent lock.

Without the `recordManualLPLock` step, the rescued campaign cannot complete its LP gate on the router, and the founder's tranche claims are permanently blocked. This is the audit fix originally described as H-4 (2026-04-15). It closes the trust gap where a manual rescue might be claimed to have happened without on-chain proof.

This is a clear example of where the protocol's centralized surface intersects with the contract's enforced guarantees. The admin can choose where to send the rescued funds. The admin cannot bypass the on-chain proof requirement. The trust boundary is the multi-sig that holds the LP locker owner role, and the published cosigner separation (Section 11) constrains what that multi-sig can do unilaterally.

7.5 What the indefinite lock does not solve

Impermanent loss continues to exist. The fee claimer holds an LP position; that position is subject to the same impermanent loss as any other LP position when the token price moves against the paired ETH. Holders selling against the pool effectively receive less than they would in an idealized setting where price had not diverged. This is a property of the AMM, not of the lock mechanism, and no protocol-level intervention can change it.

Depth is not guaranteed to be useful. A raise of 0.5 ETH produces an LP with 0.075 ETH on one side. The pool exists, but the depth is too thin for meaningful secondary trading. The protocol does not solve this — small raises produce shallow pools. The mitigation is that small raises are also low-stakes for backers, so a thin pool is appropriate.

The pool is not a price oracle. Trades against a low-volume permanent pool can move the price substantially with small orders. Holders should not treat the AMM mid-price as a fair-value indicator, particularly for raises with small total liquidity.

Permanent does not mean *valuable forever*. A pool with no trading activity is not a useful exit. If a project fails completely and trading dries up, the holder's exit liquidity is the same on-chain mechanism but functionally worthless. The lock guarantees the depth was not artificially removed. It does not guarantee anyone wants to trade.

Archived copy · cryp2nova.com

The honest summary: the indefinite LP lock is one of the strongest commitments the protocol makes, but the commitment is *about the founder's future actions* (they cannot pull the LP), not *about the project's future value* (which depends on what the founder builds and how the market receives it).

8. Reputation: Ethos and On-Chain Signals

> Anyone can back a raise, and reputation never decides who can: Vibestarter does not gate participation on reputation scores. (Launch admission is curated in the current phase against the published application rubric — Section 12.1 — not by Ethos or Starter scores.) What the platform does with reputation is surface it, so backers and founders can judge who they are dealing with for themselves.

This section describes the reputation layer that sits beside the on-chain protocol. It is part of how Vibestarter delivers design goal G3 (*permissionless on both sides — end-state; launch side currently curated*) honestly: backing is permissionless, launch admission in the current phase is decided by the published application rubric (Section 12.1) rather than by reputation, and the answer to the trust and Sybil problems that openness brings is *transparency*, not a reputation gate. The reputation layer surfaces credibility signals on every participant; it does not grant or deny access.

8.1 Reputation is a signal, not a gate

In an open market the relevant question is rarely "is this person allowed to participate" — it is "who is this person." The reputation layer answers the second question only (launch admission in the current phase is the application rubric's job — Section 12.1; backing has no admission step at all):

- **Reputation does not decide launch admission.** In the current curated phase, launches are admitted by the published application rubric (Section 12.1), not by Ethos or Starter scores; an admitted launch's success is determined by its own funding outcome, not by the founder's standing.
- **Anyone can back any raise**, regardless of reputation. There is no minimum score, no tier, and no allowlist requirement to contribute.
- **Anyone can hold tokens, raise challenges, claim refunds, and call any contract function.** The contract layer does not read reputation at all.

Reputation informs judgment; it never grants or denies access. A founder deciding whether to trust a backer, or a backer deciding whether to support a founder, gets the signals to make that call — and makes it themselves.

(Founder-configurable requirements — a founder choosing, for example, to limit their own raise to backers above some reputation threshold — are a possible future option. They are not implemented today: at present no raise gates participation on reputation.)

Archived copy · cryp2nova.com

8.2 What Vibestarter surfaces

Two families of signal, both chosen because they are expensive to fabricate at scale and independently verifiable.

Ethos. Ethos (ethos.network) is a cross-platform reputation network in which participants vouch for, review, and — where warranted — slash one another. A wallet's Ethos score aggregates that activity into a single credibility figure. Because Ethos reputation is built across platforms and over time, it is costly to manufacture for a throwaway identity. Vibestarter displays a founder's and a backer's Ethos score and standing wherever they appear.

On-chain history. Everything a wallet has done on Base is public, and Vibestarter surfaces the parts relevant to credibility:

- **Wallet age and transaction history** — how long the wallet has existed and how active it has been.
- **Vibestarter track record** — raises a founder has previously launched, raises a backer has supported, and how those played out.
- **Identity and activity signals** — ENS, Farcaster, and similar public web3 identity that ties a wallet to a persistent presence.

None of this requires the platform's permission to verify. Anyone can check a wallet's Ethos score or its on-chain history directly; Vibestarter's role is to collect and present it in context — on profiles and on each raise page.

8.3 What reputation does not do

The honest framing of the reputation layer:

- **It does not verify a founder is honest.** A wallet with a high Ethos score and a long history can still fail to deliver or act in bad faith. The on-chain challenge system (Section 6) is the enforcement mechanism; reputation is a first-glance signal, not a guarantee.
- **It does not gate the protocol.** Reputation is presentational. Every contract function is callable regardless of a participant's standing.
- **It can be cultivated.** Expensive-to-fake is not impossible-to-fake. A determined actor can build a credible-looking profile over time, so reputation should be one input among several — read alongside the project itself and the founder's specific track record.
- **It depends on third parties.** Ethos and similar sources are external services that can change methodology or become unavailable. The underlying data stays verifiable on-chain and at the source; the *display* depends on those integrations.

Which signals are surfaced, and how they are presented, is a product decision Vibestarter makes off-chain. The decentralization path for that surface is covered in Section 12.

Archived copy · cryp2nova.com

9. Economics: Worked Examples

> Three scenarios, real numbers. The mechanism is easier to evaluate against concrete cases than against parameter tables.

The examples below trace capital flow through three representative outcomes: a successful raise that completes its full schedule, a raise that is challenged mid-schedule and frozen, and a frivolous challenge that is rejected and slashed. All numbers use the protocol's actual constants and would be reproducible on-chain.

9.1 Example A: A successful raise

Setup:

- Project: "Loomwright" — a vibecoded design tool.
- Token: `$LOOM`, total supply 1,000,000,000.
- Raise target: 100 ETH, fixed-goal.
- Backers: 240, contributing amounts from 0.01 to 5 ETH.
- Raise duration: 14 days. Funded on day 11.
- Backer token allocation (per the standard split): 65% of supply = 650,000,000 $LOOM.

Flow at finalization

The 100 ETH raised splits immediately:

Slice	ETH	Notes
LP (paired into Aerodrome, receipt locked in fee claimer)	15	Paired against $LOOM to open the pool at the backer entry price −5%. Up to 150,000,000 $LOOM (15% of supply) is reserved; the locker pairs only what's needed at that price and returns the remainder to backers.
Escrow (`VibesTranchEscrow`)	85	Released across seven tranches

The 1,000,000,000 $LOOM also distributes (assuming standard configuration with 5% founder, 15% treasury, 0% community rewards, 2.5% staker rewards):

Slice	Tokens	Destination
Backers	625,000,000 (+ any unused LP reserve)	Token Distributor — claimed by backers pro-rata
LP	up to 150,000,000 reserved	Paired with 15 ETH to open the pool at the backer entry price −5%; only the amount needed at that price is deposited (LP receipt locked in the fee claimer), and the remainder is returned to backers

Archived copy · cryp2nova.com

Slice	Tokens	Destination
Treasury	150,000,000	`VibesTreasuryEscrow` — quarterly releases via challengeable proposals
Founder	50,000,000	`VibesVesting` — 6-month cliff + 12-month linear
Staker rewards	25,000,000	`VibesStakerRewards` — distributed to $VIBES stakers

These five slices are the complete allocation — they sum to the full 1,000,000,000 supply, and the platform takes no share of token supply. The platform's only per-raise fee is the 2.5% ETH tranche fee shown in the schedule below.

Tranche schedule (escrowed 85 ETH)

Tranche	Day	Tranche size (ETH)	Platform fee (2.5%)	To founder (97.5%)
T0 (kickstart)	0	8.5	0.2125	8.2875
T1	30	12.75	0.31875	12.43125
T2	60	12.75	0.31875	12.43125
T3	90	12.75	0.31875	12.43125
T4	120	12.75	0.31875	12.43125
T5	150	12.75	0.31875	12.43125
T6	180	12.75	0.31875	12.43125
Total		**85**	**2.125**	**82.875**

The founder receives **82.875 ETH** across six months, in increments. The platform receives **2.125 ETH** in tranche fees. The remaining 15 ETH is permanently in the LP, where it backs secondary market trading for the lifetime of the project.

No challenges are raised during the six months. Each tranche requests through the standard path: `requestTranche()` → 72-hour window expires → `claimTranche()`.

What backers hold at the end

Each backer holds their pro-rata share of 650,000,000 $LOOM. The LP at month six contains whatever the trading activity has accumulated to — the initial 15 ETH paired with 150M $LOOM, plus or minus the net of all trades. Backers can sell into this pool at any time. The pool will exist as long as Aerodrome does.

9.2 Example B: A challenged tranche, upheld, holder refund

Setup: Same Loomwright raise. Tranches T0, T1, and T2 release normally. The founder posts decreasing public activity from month two onward. By the day-90 mark, the public release schedule

Archived copy · cryp2nova.com

promised on the raise page has not materialized; the founder has stopped responding to community questions.

Day 90: T3 request and challenge

On day 90, the founder calls `requestTranche(3)`. The 72-hour challenge window opens. The cumulative position at this moment:

- Tranches paid: T0 + T1 + T2 = 8.5 + 12.75 + 12.75 = **34 ETH** out of escrow.
- Escrow remaining: 85 − 34 = **51 ETH**.

A backer holding 0.6% of supply (6,000,000 $LOOM — they cleared the T3–T4 threshold of 0.50%) raises a challenge, staking their 6,000,000 $LOOM in the escrow contract. Other holders representing 8% of supply call `supportChallenge()` to add their voice.

Admin review and uphold

Within the 72-hour window, the operations admin reviews the challenge. The evidence: public communication has stopped; the GitHub repo has had no commits in 40 days; the public release schedule from the raise page has missed two stated dates by 30+ days each. The admin calls `upholdChallenge(_excludeAddresses)`.

The action:

- Campaign is frozen. All future tranches (T3, T4, T5, T6) are permanently blocked.
- The challenger's 6,000,000 $LOOM stake is returned in full.
- `frozenEthBalance` is set to **51 ETH** (the remaining escrow).
- `frozenTotalSupply` is set to the current circulating supply, optionally excluding the founder's own vested holdings.

CRYP2NOVA

Holder refund

The admin publishes a refund merkle root after the 24-hour `MERKLE_ROOT_DELAY` commit-reveal window. The root commits to the snapshot of holder balances at the freeze time.

Holders then burn tokens for refunds. The formula: `refund = (frozenEthBalance × burnedTokens) / frozenTotalSupply`.

For a holder with 1,000,000 $LOOM (0.1% of supply, having contributed roughly 0.1 ETH originally):

- If `frozenTotalSupply` is 800,000,000 (assuming founder + treasury exclusions): refund = (51 × 1,000,000) / 800,000,000 = **0.0637 ETH**.
- They burn their 1,000,000 $LOOM. They receive 0.0637 ETH.

The holder's contributed 0.1 ETH split into three legs: ~0.015 ETH to the permanent LP (still in the pool; not refunded), ~0.034 ETH already paid to the founder across T0–T2 (not recoverable), and

Archived copy · cryp2nova.com

~0.051 ETH that was in escrow at freeze (now refunded pro-rata).

What this scenario does *not* recover

- The 34 ETH already paid to the founder across T0, T1, and T2 is gone. The protocol does not claw back tranches that have been claimed.
- The 15 ETH in the permanent LP is still there but is not refunded — it remains as trading liquidity against whatever the token now trades for.
- Holders who do not burn for refund retain their tokens and their share of the LP-backed market, which post-freeze will reflect the residual value of the project.

The honest summary: a successful challenge protects future capital. It does not undo past capital. This is the mathematical reality of any time-based release with a halt mechanism, and it is the reason the early-tranche challenge threshold (0.25%) is deliberately lower than the late-tranche threshold (1.00%) — the cost of being slow to catch a bad founder is highest at the start of the schedule when the most capital is still in escrow.

9.3 Example C: A rejected challenge, slash

Setup: Same Loomwright raise, alternate timeline. The founder is executing on schedule. Public commits are regular; community communication is active; the project has shipped its claimed features.

On day 60, an actor holding 0.30% of supply (3,000,000 $LOOM — just over the T0–T2 threshold of 0.25%) raises a challenge against T2. The actor has a history of speculative short positions in related tokens and stands to profit from a freeze.

The challenge stake

The actor stakes their full 3,000,000 $LOOM in the escrow contract. The 72-hour review window opens.

Admin review and reject

The operations admin reviews. The evidence is overwhelmingly positive: continuous shipping, active community, on-track public roadmap. No reasonable evidence of fraud or abandonment exists. The admin calls `rejectChallenge()`.

The action:

- 20% of the challenger's stake — **600,000 $LOOM** — is burned to `0xdead`.
- The remaining 80% — **2,400,000 $LOOM** — is returned to the challenger.
- Tranche T2 releases as normal: 12.75 ETH out of escrow, 12.43125 ETH to founder, 0.31875 ETH to platform.

Archived copy · cryp2nova.com

The challenger's loss

The actor loses 600,000 $LOOM permanently. At the prevailing market price, this is a real cost. Additionally, the actor is now subject to a 7-day `CHALLENGE_COOLDOWN`, so they cannot immediately re-challenge T3.

For the actor's strategy to have been worthwhile, the expected value of the challenge — the probability of being upheld times the gain from a freeze, minus the probability of being rejected times the slash — would have needed to be positive. With the actor's evidence base, that calculation is negative. The slash mechanism is the contract-level enforcement of that disincentive.

9.4 What these examples illustrate together

The three scenarios trace the protocol's core asymmetry:

- **Honest path:** the founder receives 82.875 ETH across six months, plus a vesting allocation of 50M $LOOM after the 18-month cliff/linear schedule. Backers hold tokens against an executing project and a permanent LP.
- **Caught dishonesty:** the founder captures at most 34 ETH before a successful challenge halts the schedule and opens holder refunds. The cost to backers is bounded by *when* the dishonesty is caught.
- **Failed griefing:** the challenger pays a 20% token cost for being wrong. The founder is not delayed beyond the 72-hour window.

The numbers are real. The constants (10%, 15%, 72 hours, 20% slash, 0.25/0.50/1.00% thresholds) are protocol-level and not configurable per-raise. The same calculations apply to any raise on the platform; the only variables are the total amount raised and the token supply allocations chosen by the founder.

A backer evaluating whether to participate in a raise can compute their own version of these scenarios for any specific raise size and any specific assumption about when (or whether) the founder might fail. The protocol's transparency is that the math is the same for everyone, the parameters are the same for every raise, and the admin's actions are bounded by contract code.

10. Contract Architecture

> A short tour of the contract topology that implements the mechanism described in Sections 5 through 8. This is an overview, not a specification — the deployed contracts are the authoritative technical reference.

The implementation comprises fourteen Solidity contracts deployed on Base (chain ID 8453), of which twelve are core protocol and infrastructure and two are testnet-only variants. All contracts

Archived copy · cryp2nova.com

target Solidity ^0.8.20 and use OpenZeppelin v5 primitives (ReentrancyGuard, SafeERC20, two-step Ownable patterns).

10.1 The topology

```
VibesLaunchRouterV2  ──  main entry point
│
├── VibesTokenFactory          (deploys per-raise ERC20 tokens)
├── VibesRegistry              (immutable provenance registry — Origin Capsules)
├── VibesTranchEscrowFactory   (EIP-1167 minimal-proxy factory)
│   └── VibesTranchEscrow      (per-raise escrow clone)
│       ├── Contributions      (ETH in from backers)
│       ├── Tranches           (ETH out to founder, time-gated)
│       ├── Challenges         (backer disputes per Section 6)
│       └── Refunds            (ETH back to holders if frozen)
├── VibesLPLocker              (creates and permanently locks Aerodrome LP per Section 7)
├── VibesVesting               (per-raise founder vesting clone)
├── VibesTreasuryEscrow        (per-raise treasury with challengeable proposals)
├── VibesTokenDistributorV2    (merkle-based backer claim + ETH refund)
└── VibesStakerRewards         (merkle-based staker reward distribution)

VibesStaking                   (independent — $VIBES staking with 7-day cooldown)
VibesIdentityRegistry          (independent — ERC-8004 agent identity registry)
```

The **router** is the single point of contact for a launch. A founder calling `launchWithCampaign()` triggers, in one transaction, the deployment of a fresh token, the cloning of a fresh escrow, the cloning of a fresh vesting contract, the LP creation and lock, the treasury setup, and the registration of the project's Origin Capsule. After the launch transaction completes, the per-raise contracts operate autonomously according to their encoded rules — the router is not consulted again except for the final tranche-completion gate.

This shape — *single entry, per-raise clones* — is the result of two constraints. First, the per-raise state (escrow balance, tranche schedule, challenge history) is the kind of thing that must not be commingled across projects, so each raise needs its own contract instance. Second, deploying a full new contract per raise would be gas-prohibitive, so the per-raise contracts are EIP-1167 minimal proxies pointing to a shared implementation. The implementation logic is shared; the storage is isolated.

10.2 The core protocol contracts

VibesLaunchRouterV2 (1,092 lines)

The entry point. Implements `launchWithCampaign()` (atomic launch + raise creation), administers the platform-level fee configuration, holds the master admin role (`owner`) and the operations admin role (`operationsAdmin`), and gates the final LP completion check that allows tranche claims to proceed.

Archived copy · cryp2nova.com

The router is the only contract with `pause`/`unpause` capability. A paused router halts new launches, new contributions, and new tranche claims across the entire platform. This is the protocol's emergency-stop primitive and is restricted to the master admin (a multi-sig).

VibesTranchEscrow (1,220 lines)

Per-raise. Holds the 85% of raised ETH that is not paired against the LP. Implements the tranche schedule (`requestTranche`, `claimTranche`, `expireChallengeIfNeeded`), the challenge mechanism (`raiseChallenge`, `upholdChallenge`, `rejectChallenge`, `supportChallenge`), the refund paths (contributor refund for failed raises, holder refund for frozen raises, excess refund for pro-rata oversubscription), and the operator escape hatches (`pauseCampaign`, `forceRefundDuringRaise`, `emergencyRefundFunded`).

The escrow is the contract that most directly implements the design goals from Section 4. Its constants (`KICKSTART_BPS = 1000`, `MONTHLY_BPS = 1500`, `PLATFORM_FEE_BPS = 250`, `CHALLENGE_WINDOW = 72 hours`, `CHALLENGE_SLASH_BPS = 2000`, `CHALLENGE_COOLDOWN = 7 days`, `MAX_TIME_DRIFT = 1 hours`, `MERKLE_ROOT_DELAY = 24 hours`) are protocol-level and not adjustable per-raise.

VibesLPLocker (256 lines)

Creates the Aerodrome volatile-pair pool at finalization, transfers the resulting LP receipt to a per-campaign `VibesLPFeeClaimer` (soulbound — it captures Aerodrome trading fees while the principal liquidity stays locked), and records the lock on-chain. Includes the `resolveRescuedFunds` / `recordManualLPLock` path (Section 7.4) for handling LP creation failures.

VibesVesting (256 lines)

Per-raise. Implements the founder's 18-month vesting schedule: 180-day cliff (0% vested), then 365 days of linear vesting (0% → 100%). The freezer role can burn all unvested founder tokens to `0xdead`, used when a treasury challenge is upheld as malicious (Section 11).

VibesTreasuryEscrow (459 lines)

Per-raise. Holds the treasury portion of the token allocation. Operates a proposal-based withdrawal model: the treasury admin proposes a withdrawal, the proposal is challengeable on the same general pattern as tranche challenges, and upheld challenges can either block the proposal (rework) or burn the entire treasury (malicious — the nuclear option that also freezes founder vesting).

VibesTokenDistributorV2 (401 lines)

Per-raise. Handles the merkle-based distribution of backer tokens after finalization, plus any ETH refunds from failed or frozen raises. Backers claim with a merkle proof of their allocation. Unclaimed allocations can be swept by the admin after a six-month window, with `totalPendingEthRefunds` protected from the sweep.

Archived copy · cryp2nova.com

10.3 The infrastructure contracts

VibesTranchEscrowFactory (252 lines)

Deploys escrow clones using EIP-1167. Holds the maximum-raise-duration constant (`MAX_RAISE_DURATION = 30 days`) and the maximum scheduling window. The authorized router and the LP locker address are set here and used by every escrow clone.

VibesTokenFactory (50 lines)

Deploys fixed-supply ERC20 tokens. No access controls — anyone can call it. The minimalism is intentional: the factory has no authority, makes no claims about the deployed tokens, and is interchangeable with any other token factory.

VibesRegistry (213 lines)

The provenance layer. When a raise launches through the authorized router, the registry emits a `VibesCertified` event committing the project's Origin Capsule on-chain. The capsule includes the launch metadata, the founder's wallet, the project description hash, and other provenance fields. The registry is append-only.

VibesIdentityRegistry (205 lines)

ERC-8004 identity registry for AI agents associated with the platform. Independent of the launch path; allows agents acting on behalf of users (autonomous backers, automated challenge monitoring) to maintain on-chain identity.

10.4 The staking contracts

VibesStaking (301 lines)

Independent of the launch path. Allows $VIBES holders to stake their tokens with a 7-day unstaking cooldown. Implements lazy per-user balance snapshots (audit fix F4) so that the reward-distribution contract can compute accurate stake-weighted allocations without requiring a global snapshot.

VibesStakerRewards (500 lines)

Distributes Vibetokens (the per-raise tokens) to $VIBES stakers on a snapshot-and-accumulator pattern. Each new raise allocates 2.5% of its token supply to the rewards contract, distributed pro-rata against staked $VIBES at the time of the raise.

10.5 Security patterns

Archived copy · cryp2nova.com

The contracts use a small, opinionated set of security primitives. The list is not exhaustive — it is the set that recurs across multiple contracts:

- **OpenZeppelin ReentrancyGuard** on every ETH-handling contract (eight contracts).
- **OpenZeppelin SafeERC20** for all token transfers (seven contracts).
- **Checks-Effects-Interactions** for all ETH transfers. State updates precede `.call{value:}`.
- **Two-step ownership/admin transfer** for all privileged-role contracts. A transfer is initiated by the current holder and must be accepted by the new holder before taking effect. Prevents accidental transfer to an inaccessible address.
- **Two-tier admin separation.** A master admin (multi-sig) controls infrastructure and rescue. An operations admin (EOA or smaller multi-sig) controls day-to-day operations and cannot extract user funds.
- **Commit-reveal timelock** on refund merkle root publication (`MERKLE_ROOT_DELAY = 24 hours`). Holders can verify the committed root before claims open.
- **Oracle time-drift guard.** Time oracle reads are bounded by `MAX_TIME_DRIFT = 1 hours` to prevent manipulated time from shortening challenge windows or accelerating tranche releases.
- **Proof-based LP lock recording.** The LP locker's manual-resolution path requires on-chain proof that the campaign's fee claimer (or `0xdead`, for a legacy burn) holds at least `lpAmount` of the pool token before the rescued state is marked verified-locked.
- **Cross-contract finalization guards.** Token claims hard-require `finalizationPhase == FullyComplete`. Emergency refunds query the router's finalization state via try/catch. Prevents the token-plus-ETH double-dip that state drift could otherwise enable.
- **EIP-1167 minimal proxies** for per-raise contracts to keep gas costs reasonable across many launches.
- **Custom errors** instead of revert strings for gas-efficient reverts.
- **Dead-address locking** for irrecoverable burns (slashed challenge stakes, malicious-treasury burns, holder-refund token burns). The primary LP lock instead uses a soulbound fee claimer (Section 7).

CRYP2NOVA

This list has been shaped by the two audit cycles described in Section 11. Each of the named guards (commit-reveal delay, time-drift guard, proof-based recording, cross-contract finalization checks) corresponds to a specific finding from the audit history.

10.6 What this section deliberately omits

This is a tour, not a specification. Specific details intentionally outside the scope of this section:

- **Function signatures and parameter types.** Verifiable on the deployed contracts via a Base block explorer.
- **Storage layout.** Contracts that use upgradeable patterns or shared storage have specific layouts documented in their respective sources.
- **Gas budgets and cost analysis.** See the deployment notes for gas profiles of representative operations.

Archived copy · cryp2nova.com

- **Test coverage.** The contracts ship with a comprehensive test suite covering the mechanism paths described above.
- **Audit findings in detail.** See the remediation summary at app.vibestarter.xyz/audit.

The intent of this section is to give a reader enough to understand *which contract enforces which design goal* and to provide the entry points for deeper investigation. The protocol's correctness claims live in the source and the audits, not in this paper.

11. Trust and Safety

> The contracts cannot reason about the world. Some surfaces must be operated by humans. This section names them and describes the constraints under which they operate.

The honest position: every permissionless protocol that handles user funds has an operator layer of some shape. The question is not *whether* one exists — it always does, somewhere, even if it is implicit. The question is whether the operator layer is named and bounded by the contract, or whether it is hidden behind language like *decentralized* and *trustless* that the protocol cannot actually deliver.

Vibestarter's operator layer is explicit. It is structured around two principles: that no single party can act unilaterally on user funds, and that every operator action has a contract-enforced consequence the operator cannot override.

11.1 The two-tier admin model

The protocol distinguishes between two privileged roles. The separation is enforced by contract logic, not by convention.

CRYP2NOVA

Master Admin (`VibesLaunchRouterV2.owner`)

The master admin is the holder of the router's `owner` role. It is expected to be a **Gnosis Safe multi-signature wallet** with at least a 2-of-3 threshold (3-of-5 recommended at scale).

Its powers — the ones that can move user funds or alter infrastructure — include:

- `pause()` / `unpause()` — global emergency stop
- `setEscrowFactory()` / `setLPLocker()` — replace infrastructure contracts
- `setFeeConfig()` — adjust platform fee parameters
- `setOperationsAdmin()` — appoint or revoke the operations admin
- `rescueETH()` — withdraw ETH from the router contract above deposit reserves
- `rescueERC20()` — rescue stuck ERC20 tokens, guarded by active-claim/escrow/LP checks
- `forfeitDeposit()` — seize a founder's deposit and send it to the fee recipient
- `transferOwnership()` — initiate two-step ownership transfer

Archived copy · cryp2nova.com

The master admin is the only role that can extract ETH from the router. This is the highest-trust surface in the system. The mitigations:

- **Multi-sig only.** A single-EOA master admin is incompatible with operating the platform. The deployment configuration enforces transfer to a Safe after launch.
- **Two-step transfer.** Ownership changes require acceptance by the new holder, preventing transfers to inaccessible addresses.
- **Bounded rescue scope.** `rescueETH` is bounded by the deposit reserves tracked on the router; founder deposits cannot be drained via rescue.
- **Public on-chain visibility.** Every action by the master admin is visible on-chain. Multi-sig transactions show their signers; rescue events are emitted; pause/unpause events are emitted.

Operations Admin (`VibesLaunchRouterV2.operationsAdmin`)

The operations/escrow admin is a separately-held role, revocable at any time. In the deployed configuration it is **M-3, a dedicated single-key hot EOA** (not a multisig), kept hot for minute-level freeze response. **NOTE (admin split, 2026-06-07):** the escrow/challenge admin is sourced from `factory.admin` = M-3 (single key) and is passed to every new escrow at creation time as its `admin`; the router `operationsAdmin` — which feeds the **treasury** admin, including the irreversible burn — is the **M-1 Safe**, not M-3.

The escrow/challenge admin's powers (M-3), exercised across all per-raise escrows (the two treasury-challenge functions below are exercised by the **M-1 treasury admin**, not M-3, since the 2026-06-07 split):

- `upholdChallenge()` / `rejectChallenge()` — adjudicate challenges
- `freezeCampaign()` — freeze a funded campaign for holder refunds
- `setRefundMerkleRoot()` — publish refund merkle root (commit-reveal gated)
- `pauseCampaign()` / `resumeCampaign()` — pause specific raises
- `forceRefundDuringRaise()` — terminate an active raise pre-finalization
- `emergencyRefundFunded()` — emergency refund path for funded raises
- `upholdChallengeRework()` / `upholdChallengeMalicious()` — treasury challenge adjudication

What the operations admin **cannot** do:

- Move user funds to arbitrary addresses. Every action above either returns funds to identified holders, slashes to `0xdead`, or freezes funds in place.
- Alter the tranche schedule. The schedule is hard-coded in the escrow.
- Change the LP lock. The LP receipt is permanently held in a soulbound per-campaign fee claimer with no withdraw path.
- Bypass the commit-reveal delay on refund merkle roots.
- Adjust the slash percentage or the challenge window length.
- Call any function on the master admin's surface (rescue, pause, infrastructure changes).

Archived copy · cryp2nova.com

If the operations admin is compromised, the worst-case scenario is bounded: an attacker can freeze campaigns and burn individual treasuries, but cannot exfiltrate ETH. The master admin can revoke a compromised operations admin in a single transaction via `setOperationsAdmin(newAddress)`. The operations admin is, by design, a *fast-revocable* role.

11.2 Off-chain key topology (Safes + EOAs)

Beyond the on-chain `owner` and `operationsAdmin` roles, the operational roles sit at the organizational level across **two Gnosis Safes (M-1, M-4) plus two single-key EOAs (M-2 cold, M-3 hot)** on Base. The two Safes use a 2-of-2 threshold today (moving to 2-of-3); **M-2 and M-3 are single keys, not multisigs.**

Safe	Role	Purpose
M-1	Platform Operations	Receives 2.5% ETH tranche fees (no token-side launch fee exists). Pays platform infrastructure costs.
M-2	$VIBES Raise Founder Wallet	Receives Vibestarter's own raise tranche payouts (pre-entity custody).
M-3	Ops/escrow admin (single-key hot EOA — not a multisig)	Holds the escrow/factory `admin` role (from `factory.admin`): adjudicates raise/tranche challenges, freezes, force-refunds. The router `operationsAdmin` (treasury admin) is M-1, not M-3, since the 2026-06-07 split.
M-4	Community Rewards	Administers the `VibesCommunityRewards` contract (post-cliff distribution batches).

Separation of concerns is enforced by cosigner policy:

- M-2 (founder wallet) and M-3 (escrow/challenge admin) are **distinct single-key EOAs** — separate keys, no shared control.
- The two Safes (M-1, M-4) limit cosigner overlap to at most one shared seat to avoid over-concentration.
- For challenges against the $VIBES raise itself, the founder's adjudication conflict is mitigated by **off-chain transparency** (the published challenge standards), since M-3 is a single key with no on-chain cosigner; M-3 only executes the publicly-reasoned decision.

The cosigner separation prevents the obvious failure modes — the founder adjudicating challenges against themselves, the same parties controlling both the platform fee wallet and the protocol admin — without requiring perfect role isolation, which is impractical at small operating scale.

11.3 The `_excludeAddresses` parameter

One specific operator surface deserves explicit discussion because it is the highest-trust function in the system that does not move funds.

Archived copy · cryp2nova.com

When `upholdChallenge()` or `freezeCampaign()` is called, the admin supplies an `_excludeAddresses` array. Addresses in this array are excluded from the `frozenTotalSupply` denominator that determines the proportional refund amount for each holder. The math: a holder burning X tokens receives `(frozenEthBalance × X) / frozenTotalSupply` ETH back. A smaller `frozenTotalSupply` means each remaining token redeems for more ETH.

The reason this parameter exists: at freeze time, some addresses holding the project's token may be legitimately non-redeemable. Examples include the founder's vesting contract (already partially burnable via the malicious-challenge path), the treasury contract (subject to its own challenge mechanism), and the burn address itself. Including these in the refund denominator would over-dilute the refunds to actual backers, which is the opposite of what the freeze is trying to accomplish.

The trust risk: a malicious admin could in principle exclude addresses they should not, increasing the refund for specific holders (or for themselves, if they hold tokens via an address the admin chooses to exclude only conditionally).

The mitigations:

- **Publication and review.** The `_excludeAddresses` array is part of the on-chain transaction. Any holder can verify which addresses were excluded.
- **Challenge standards.** The challenge standards (Section 6.6) define the off-chain policy for which addresses are excluded under what circumstances. Deviations are publicly auditable.
- **On-chain publication (no cosigner consent).** M-3 is a single-key hot EOA, so the `_excludeAddresses` array is **not** gated by cosigner approval; the mitigations are its on-chain visibility, the 24-hour `MERKLE_ROOT_DELAY`, the published challenge standards, and — for the $VIBES raise — off-chain recused adjudication.
- **24-hour `MERKLE_ROOT_DELAY`.** After the freeze, the admin must commit a refund merkle root and then wait 24 hours before claims open. Holders can verify the root during this window and surface complaints before any ETH moves.

This is one of the surfaces where the protocol relies on transparency and review rather than pure on-chain enforcement. It is named here because pretending otherwise would be the kind of dishonesty Section 4.1 ruled out.

11.4 Audit history

The contracts have been through one external audit (ZXVC LLC, May 2026) plus internal review cycles (April and June 2026). All High findings were remediated and re-tested; certain centralization findings (no admin timelocks on the treasury burn / infrastructure setters) are accepted or deferred and disclosed as residual risk.

April 2026 audit

Issued findings across multiple severity levels. The fixes that landed on 2026-04-15 include:

Archived copy · cryp2nova.com

- **H-1: Testnet time-drift parity.** Testnet variant of escrow now enforces the same `MAX_TIME_DRIFT = 1 hours` as mainnet. Enforced in CI by `AuditTestnetParityGuard2026_04.t.sol`.
- **H-2: Testnet merkle-root-delay parity.** Testnet `MERKLE_ROOT_DELAY = 24 hours` identical to mainnet.
- **H-3: Cross-contract finalization guard.** Token claim path hard-requires `finalizationPhase == FullyComplete`. Emergency refund queries router finalization state via try/catch. Prevents token-plus-ETH double-dip.
- **H-4: Proof-based LP lock recording.** `recordManualLPLock` requires on-chain proof that the LP receipt is held by the campaign's fee claimer (or `0xdead`, for a legacy burn) before transitioning rescued state to verified-locked.
- **H-06 (earlier): Oracle time-drift guard.** Time oracle reads bounded by `MAX_TIME_DRIFT = 1 hours`.
- **L-1: Reentrancy guard on `resolveRescuedFunds`.** Added 2026-04-15.

May 2026 audit

Follow-up audit covering the staking and rewards contracts and the contracts modified post-April. Findings and remediations are summarized at app.vibestarter.xyz/audit.

The audit findings should be treated as the authoritative source for known issues — this paper does not substitute for them. A remediation summary is published at app.vibestarter.xyz/audit.

11.5 Incident response

The platform maintains a documented incident response procedure. The key components:

- **Pause as a first action.** The master admin can pause the router globally while an incident is triaged.
- **Per-campaign pause.** The operations admin can pause a specific raise without halting the entire platform.
- **Force refund.** The operations admin can terminate an active raise pre-finalization, returning all ETH to contributors.
- **Public communication.** Incident-class events trigger public communication via the platform's announced channels.

Incidents that have happened on-chain are publicly verifiable through the chain history. The platform commits to publishing post-mortems for material incidents, including the timeline of detection, the parties involved in response, and the contract-level actions taken.

11.6 What the safety model does *not* cover

A short list of risks that the trust and safety model explicitly does not address:

Archived copy · cryp2nova.com

- **Founder choices that are within their rights.** A founder who claims all tranches on schedule and then ships nothing valuable is not committing a contract violation. The protocol releases the schedule; the market values the result.
- **Off-chain regulatory action.** A jurisdiction may determine that a token was issued in violation of local law. The protocol cannot prevent that determination. Section 13 covers regulatory risk.
- **Smart contract bugs.** Audits reduce the probability of unknown bugs; they do not eliminate it. A previously-unknown bug could result in loss of funds. The audit history is the relevant disclosure.
- **Bridge or chain-level risk.** The contracts deploy on Base. Risks to Base itself — sequencer downtime, L2 finality issues, or systemic problems in the Ethereum L1 settlement layer — are outside the protocol's control.
- **Wallet compromise.** A backer whose wallet is compromised has lost their tokens via the standard ERC20 transfer path. The protocol has no recovery mechanism for individual user compromise.

These limits are the boundary of what an honest trust-and-safety section can claim. Everything else — the contract guarantees, the operator constraints, the audit findings — is on-chain or in the public audit record.

12. Governance and the Decentralization Path

> The protocol does not pretend to be fully decentralized today. This section names every centralized surface, explains why it is centralized at launch, and describes the path on which it decentralizes.

A protocol that ships honest decentralization claims must distinguish between three categories: surfaces that are already permissionless, surfaces that are centralized and will remain so, and surfaces that are centralized today and decentralize on a published path. Mislabeling any of these is the standard pattern of dishonesty in this space. The position here is to name each one.

12.1 What is already permissionless

The protocol layer is permissionless today. Without any operator action:

- **Anyone can create a token** through the router's simple `launch()` path — no signature, no approval, no whitelist at the contract level.
- **Raise launches are gated during the current curated phase.** `VibesLaunchRouterV2.launchWithCampaign()` requires a platform-issued EIP-712 launch signature (`trustedLaunchSigner`), and the platform issues it only to wallets meeting the published criteria — including, in the current phase, an approved founder application reviewed against a published rubric (see the Launch Admission & Moderation Policy). Permissionless raise admission remains the stated end-state of this document; until the curated phase ends, describing the

Archived copy · cryp2nova.com

launch path as sign-off-free would be false, so we don't. Backing, challenging, claiming, and refunds below are permissionless today.

- **Anyone can back a raise** by sending ETH to the escrow's `contribute()` function during the active window (subject to the platform terms signature that enforces ToS acceptance and sanctions screening). No accreditation, no minimum reputation.
- **Anyone can hold and transfer tokens** issued by raises. The tokens are standard fixed-supply ERC20s with no transfer restrictions.
- **Anyone holding a tranche-challenge threshold can raise a challenge** during a challenge window. The threshold is graduated (Section 6.2) but the action is permissionless.
- **Anyone can support a challenge** by calling `supportChallenge()` while holding any non-zero token balance.
- **Anyone can claim a tranche on behalf of the founder** after the challenge window expires uneventfully. The function is permissionless; the funds flow to the founder's pre-set address regardless of caller.
- **Anyone can claim a refund** they are entitled to (contributor refund for failed raises, holder refund for frozen raises via merkle proof, excess refund for pro-rata oversubscription).
- **Anyone can stake $VIBES** by calling `VibesStaking.stake()`. No allowlist.

These are the on-chain interactions a backer or founder will typically have with the system. Apart from the launch-signature gate named above (and the terms signature on gated actions), none require Vibestarter Labs' permission.

12.2 What is centralized today

The centralized surfaces. Each is named with the reason it currently requires operator involvement.

Launch admission (curated phase)

The backend launch signature is the enforcement point for sanctions screening, terms acceptance, and — in the current phase — the curated founder-application review. The review operates against the published rubric in the Launch Admission & Moderation Policy; every decision is logged with its reason.

Why centralized: the platform launched invite-only as a trust-model control; the application review is the per-launcher screen against fraud and impersonation while reputation infrastructure matures.

Path: the stated end-state is permissionless admission — the signature remains as a mechanical screen (sanctions, geo, terms) with no per-founder discretion. The curated phase ends on a published decision, not silently.

Challenge adjudication

The challenge review (Section 6) is performed by the operations admin (M-3, a single-key hot EOA). The admin decides whether to uphold, reject, or let expire each challenge. Backers have 72 hours to

Archived copy · cryp2nova.com

file after a tranche request; once filed, the admin has a 7-day adjudication window before the challenge auto-expires in the founder's favor (raises launched before the PC-09 cut-over, including $VIBES, run both clocks at 72 hours — the constants are immutable per escrow).

Why centralized: A contract cannot read the world. Determining whether a founder has actually abandoned a project, whether claimed progress is real, or whether a missed deadline reflects fraud or strategic pivot requires off-chain judgment. Until a decentralized adjudication mechanism (jury panels, prediction markets, community vote) has been built and stress-tested, the operations-admin key is the practical answer.

Constraints today: The admin can only choose between contract-defined outcomes (uphold, reject, expire). The admin cannot move funds to arbitrary destinations, cannot adjust slash percentages, and cannot skip the commit-reveal delay on refund roots. The challenge standards (Section 6.6) define the criteria the admin uses, and deviations are publicly auditable.

Refund merkle root publication

After a campaign is frozen, the admin publishes a refund merkle root that snapshots holder balances. Holders then claim against the root.

Why centralized: The snapshot must be taken at a specific block, must correctly capture all holder balances, and must apply the `_excludeAddresses` policy described in Section 11.3. This is bookkeeping that requires off-chain computation against on-chain state.

Constraints today: The 24-hour commit-reveal delay (`MERKLE_ROOT_DELAY`) gives holders time to verify the published root before claims open. The root is publicly inspectable. Holders who believe the root is incorrect can surface complaints during the delay window.

`_excludeAddresses` policy

The exclusion array passed to `upholdChallenge()` and `freezeCampaign()` is an admin decision.

Why centralized: The set of legitimately non-redeemable addresses (vesting contracts, treasuries, the burn address itself) varies per raise and is not derivable on-chain without additional infrastructure.

Constraints today: Governed by the challenge standards (Section 6.6). Visible on-chain in the transaction. Subject to operations-admin review (M-3, a single key) and the published challenge standards.

Reputation display

The reputation surfaced for founders and backers (Section 8) — Ethos scores and on-chain history — is fetched and presented off-chain by the platform.

Why centralized: The signals are sourced from third parties (Ethos) and from on-chain data that must be queried and rendered. Which signals are surfaced, and how they are presented in the

Archived copy · cryp2nova.com

interface, is a product decision.

Constraints today: The underlying data is externally sourced and independently verifiable — anyone can check a wallet's Ethos score or its on-chain history directly, without the platform's involvement. Vibestarter curates the presentation; it does not control or custody the data.

Infrastructure parameter changes

Adjustments to platform fee configuration, founder deposit amounts, the escrow factory address, the LP locker address, and similar infrastructure changes are master admin actions.

Why centralized: Some of these require coordinated upgrades (a new escrow factory must be deployed and authorized in a specific sequence). Others reflect operating-cost considerations that change over time (fee structures, deposit amounts).

Constraints today: All such changes require master admin (the M-1 Safe) action. Each emits public events. A timelock layer is recommended for infrastructure changes and is on the roadmap (see 12.4).

Operations admin appointment

The master admin appoints and can revoke the operations admin via `setOperationsAdmin()`.

Why centralized: Until the operations admin is held by a sufficiently large multi-sig with a community-elected composition, the master admin's appointment of the operations admin is the practical mechanism.

Constraints today: The operations admin cannot extract user funds. A compromised or malicious operations admin can be revoked in a single transaction.

12.3 What will remain centralized

A short list of surfaces that are not on a decentralization path, with reasons.

- **Emergency pause.** The router's pause capability is intentionally fast and unilateral (within the multi-sig). A pause is a defensive action that must be available in seconds during an incident; routing it through a community vote would defeat its purpose.
- **Fund rescue (`rescueETH` / `rescueERC20`).** Bounded by deposit reserves and active-claim checks, but the master admin's ability to recover stuck funds is essential to operating the platform. Decentralizing this fully would mean accepting that legitimately-stuck funds (failed transactions, mis-sent tokens) become unrecoverable.
- **The protocol's own raise (`$VIBES`) parameters.** The Vibestarter Labs raise itself uses admin overrides (the `disableStakerAllocation` flag, the community-rewards authorization) that are not available to other launches. This is permanent — these overrides exist specifically because the

Archived copy · cryp2nova.com

protocol cannot stake against itself at launch and because the community rewards path requires special atomic deployment.
- **The platform fee.** Fees are configurable by the master admin within bounds. The expectation is that fees move down over time, not up, but the protocol does not commit to a maximum.

These are not failures of decentralization — they are the load-bearing operator functions that any honest platform requires.

12.4 The decentralization path

For the surfaces in 12.2 that are on a decentralization path, the staged plan:

Stage 1 — Multi-sig consolidation (current)

The escrow/challenge admin role is held by M-3, a single-key hot EOA (kept hot for minute-level freeze response — **not a multisig**). The master + treasury admin is the M-1 Safe. For the protocol's own raise, the founder's adjudication conflict is mitigated by off-chain transparency (the published challenge standards), not by an on-chain cosigner; an on-chain guardian module to move M-3's non-freeze setters behind a multisig is planned, not yet deployed.

Status: in place.

Stage 2 — Challenge standards transparency

The criteria the operations admin uses to adjudicate challenges are published in this paper (Section 6.6). Deviations from published standards are publicly visible and grounds for community complaint.

Status: in place.

Stage 3 — Adjudicator diversification

M-3 (currently a single-key hot EOA) is migrated to a multi-sig that includes community members beyond Vibestarter Labs. The cosigner set is rotated periodically. The criteria for cosigner inclusion are published.

Status: roadmapped post-mainnet.

Stage 4 — Specialized adjudication panels

Challenges of different types (suspected fraud, missed deliverables, treasury proposal disputes) are routed to specialized panels with relevant expertise. Each panel is a sub-multisig with its own cosigners.

Status: roadmapped.

Archived copy · cryp2nova.com

Stage 5 — Community-elected adjudicators

Adjudicators are elected by $VIBES stakers or a similar credibly-decentralized constituency. Elections are time-bounded and the elected adjudicators have term limits.

Status: longer-term, contingent on a stable staking economy and meaningful participation.

Stage 6 — On-chain dispute resolution

Where possible, individual elements of adjudication are moved on-chain (e.g., automated verification of specific milestone-style claims, prediction-market-based resolution for ambiguous cases, jury panels with stake-weighted voting). The operations admin retains final authority for cases that cannot be resolved on-chain.

Status: research-stage, contingent on the maturity of decentralized dispute resolution primitives.

Timeline

The protocol does **not** publish a specific calendar for these stages. Committing to specific dates for decentralization is the kind of promise that gets broken loudly in this space, and we would rather make the path visible and the criteria explicit than commit to dates we may have to revise.

What we will commit to: each stage transition will be publicly announced, including the change to the on-chain configuration, the cosigner set (if relevant), and the new published criteria.

12.5 The honest reality

Decentralization is a property of *what the protocol enforces in code*, not of *what the operator promises in marketing*. The contract-enforced commitments described in Sections 5 through 10 are decentralized: they are guaranteed by deployed code, not by Vibestarter Labs' continued cooperation. A Vibestarter Labs that disappears tomorrow cannot retroactively shorten a challenge window, accelerate a tranche, or unlock the permanent LP.

What requires continued operator involvement: adjudicating challenges, publishing refund merkle roots, maintaining the reputation system. A Vibestarter Labs that disappears tomorrow without first transferring these roles to a successor would leave the protocol in a *degraded mode* — tranches continue to release on time, but no one can act on challenges and no one can publish refund roots for frozen campaigns.

This is the honest framing. The contract guarantees are robust to operator failure. The operational layer is not. The decentralization path in 12.4 is the work of making the operational layer also robust to operator failure, by distributing the operator role across an increasingly diverse and credibly committed set of parties.

Archived copy · cryp2nova.com

13. Risks and Disclosures

> Everything that can go wrong, named.

A whitepaper that does not name the failure modes of the protocol it describes is, at best, marketing. This section lists the categories of risk that participants — backers, founders, and operators — should understand before interacting with Vibestarter. None of these are theoretical. All of them have analogs in prior systems.

13.1 Smart contract risk

The contracts have been through one external audit (ZXVC LLC, May 2026) plus internal review cycles (April and June 2026). All High findings were remediated and re-tested; certain centralization findings (no admin timelocks on the treasury burn / infrastructure setters) are accepted or deferred and disclosed as residual risk. The external audit report is public.

The unmitigated residual: **previously-unknown bugs may exist**. No audit guarantees the absence of bugs; it raises the probability that critical issues have been identified. The mitigation paths the protocol has in place:

- **Pause capability.** A discovered exploit can be contained by halting new launches and new claims while a fix is prepared.
- **Bounty program.** The bug bounty (Section 8.2) provides material incentive for responsible disclosure.
- **Open-source code.** The contracts are public, and the audit reports are public. Third parties can and should review the source independently.

What the mitigation does not address: a bug exploited *before* it is discovered may result in irreversible loss of funds. This is a category of risk that no protocol has eliminated.

13.2 Regulatory risk

Tokens issued via Vibestarter are intended to function as **utility tokens** within their respective project ecosystems: they confer no equity, ownership, dividend, profit-share, or revenue-share rights. Whether any specific token is a security is a legal determination that turns on its particular facts and the jurisdiction in question. The legal disclaimers shown on raise pages and at app.vibestarter.xyz/terms set out what the tokens do and do not confer.

The unmitigated residual: **regulatory bodies in specific jurisdictions may reach different conclusions**. Possible outcomes include:

Archived copy · cryp2nova.com

- A specific token being determined by a regulator to be a security in their jurisdiction, with enforcement against the founder, the platform, or backers.
- New regulation in a relevant jurisdiction altering the legal status of platforms in this category.
- Restrictions on platform access from specific jurisdictions, imposed by Vibestarter Labs in response to regulatory guidance.

The protocol itself cannot make legal determinations. Participants are responsible for understanding the legal status of the activity in their own jurisdiction. The disclaimers and language standards reduce the surface area for confusion; they do not eliminate the underlying regulatory exposure.

13.3 Market risk

Tokens issued via Vibestarter are exposed to standard secondary market risks:

- **The token may go to zero.** A project that fails to ship, fails to attract users, or fails for any market reason will see its token trade toward zero. The permanent LP guarantees there is *somewhere to sell*, not that the price will be favorable.
- **Impermanent loss in the LP.** The Aerodrome pool is subject to standard AMM dynamics. Holders selling against a pool that has experienced significant price divergence receive less than they would in an idealized setting.
- **Slippage on thin pools.** Small raises produce shallow LPs. Trades against a thin pool can move the price substantially with small orders.
- **Correlated market risk.** A general crypto market downturn affects all tokens, including those on Vibestarter. Project-specific execution does not insulate against macro market movements.

The protocol does not promise that backers will profit. The legal disclaimers explicitly note that contributions should be amounts participants can afford to lose entirely. This is a literal statement of the risk profile.

CRYP2NOVA

13.4 Founder risk

Even with the mechanisms described in Sections 5 through 8, founder-related failure modes remain:

- **Founder may capture early tranches before being caught.** The kickstart (10%) and one or two monthly tranches (15% each) may be claimed before a successful challenge halts the schedule. In Example B (Section 9.2), a successful challenge at day 90 leaves 40% of escrow already claimed — that capital is not recoverable.
- **Founder may game challenges.** A founder with reputational standing may successfully argue against legitimate concerns, prolonging the schedule despite underperformance.
- **Founder may execute poorly but not fraudulently.** A founder who is genuinely trying to ship but making poor product decisions is not actionable through the challenge mechanism. Holders bear the cost of poor execution as a market outcome.

Archived copy · cryp2nova.com

- **Founder may abandon post-tranche.** A founder who completes the tranche schedule and then stops working on the project has fulfilled their contract obligation. Vesting on the 5% founder allocation provides some incentive to continue (the cliff is at 6 months, full vesting at 18 months), but does not enforce continued shipping.

The protocol minimizes the *unbounded* version of these risks. It does not eliminate them.

13.5 Operator risk

The operator surfaces described in Sections 11 and 12 introduce specific risks:

- **Operations admin compromise.** A compromised operations admin can freeze campaigns and burn individual treasuries before the master admin revokes the role. Funds cannot be exfiltrated, but in-flight campaigns can be disrupted.
- **Master admin compromise.** A compromised master admin holding the multi-sig threshold can pause the platform, alter infrastructure, or extract ETH above deposit reserves. The multi-sig configuration is the primary mitigation.
- **Operator disappearance.** If Vibestarter Labs ceases operating without transferring the operations admin role, the protocol enters a degraded mode: tranches continue to release on schedule, but challenges cannot be adjudicated and refund roots cannot be published.
- **Operator capture.** If the operations admin multi-sig becomes captured by interests aligned with founders against backers, challenges may be systematically rejected or never adjudicated. The cosigner separation policy and the published challenge standards are the mitigations.

These risks are the trade-off the protocol makes by having an operator layer at all. Section 12 covers the path on which they decrease over time.

13.6 Challenge mechanism risks

The challenge mechanism itself has failure modes:

- **Coordinated challenges to grief honest founders.** Even with the slash and cooldown, a well-funded coordinated attack with multiple distinct addresses can serially challenge a project. Each challenge requires a fresh address with the threshold balance, the slash on each rejected challenge is meaningful but not catastrophic, and the cumulative delay can be material. Mitigation: the slash percentage is calibrated against this risk; the operations admin can recognize coordinated patterns and reject them.
- **Insufficient challenges against dishonest founders.** A small holder base, an inattentive community, or a project whose holders are not technically capable of identifying issues may produce no challenges against a founder who should be challenged. The protocol cannot manufacture challenges.
- **Successful challenge but slow refund.** After a campaign is frozen, the admin must publish a refund merkle root. This requires off-chain computation; the 24-hour `MERKLE_ROOT_DELAY` adds

Archived copy · cryp2nova.com

further latency. Holders wait at least a day for their refund path to open.
- **Holder refunds rely on holders acting.** A holder who does not burn their tokens for refund does not receive their pro-rata share of the frozen escrow. The protocol does not push refunds.

13.7 Reputation display risks

The reputation surfaced on the platform (Section 8) has its own caveats:

- **Displayed reputation is not a guarantee of honesty.** A founder with a high Ethos score and a long on-chain history can still fail to deliver or act in bad faith. Reputation is a signal to inform judgment, not a substitute for it.
- **Signals can be cultivated.** Ethos standing and on-chain history are expensive to fabricate at scale, but a determined actor can build a credible-looking profile over time. Treat reputation as one input among several.
- **Third-party dependency.** Ethos and other reputation sources are external services that can change methodology, rate-limit, or become unavailable, which would degrade or interrupt the displayed signals.
- **Reputation is not access.** The platform does not gate participation on these signals: anyone can back any raise regardless of reputation, and launch admission (curated in the current phase — Section 12.1) is decided by the published application rubric, not by Ethos or Starter scores. Founder-configurable requirements may be offered in future but are not in place today.

13.8 LP-specific risks

The permanent LP introduces specific dynamics:

- **Impermanent loss is permanent.** A normal LP holder can withdraw their position when divergence has reverted. The permanent LP cannot — the position is locked in whatever state divergence has produced.
- **Trading fees go to the protocol, not to holders.** Aerodrome routes a portion of trading fees to the LP position. For the locked LP, the fee claimer captures those fees — the ETH side to the platform, the token side to the project treasury (or burned if there is no treasury) — so token holders do not receive them.
- **No rebalancing.** A liquidity pool that becomes too imbalanced (e.g., one side fully drained) cannot be rebalanced. The pool may end up effectively non-functional even though it nominally exists.
- **AMM contract risk.** The pool is on Aerodrome. Risks to the Aerodrome protocol — including its own contract risk, governance changes, or fee structure changes — affect the permanent LP.

13.9 Platform-level risks

Risks at the level of the platform rather than individual raises:

Archived copy · cryp2nova.com

- **Chain risk.** The protocol deploys on Base. Risks to Base (sequencer downtime, L2 finality issues, bridge security) affect all activity on the platform.
- **External dependency risk.** Aerodrome (LP creation), Ethos and other reputation signals (reputation display), and third-party services (X for identity linkage) are dependencies that could change behavior or become unavailable.
- **Founder deposit erosion.** The founder deposit (0.05 ETH at baseline) is intended to deter spam launches. As ETH prices change, the real cost of the deposit varies. The deposit amount is operator-adjustable, but adjustments lag.
- **Fee structure changes.** Platform fees are configurable by the master admin. The protocol does not commit to a maximum fee.

13.10 What the disclosures do not cover

This list is not exhaustive. New failure modes may emerge that no one has yet identified. Specifically:

- New attack patterns against the challenge mechanism, the reputation system, or the LP lock.
- Emergent market behaviors that the protocol's mechanism produces unintentionally.
- Legal or regulatory developments in jurisdictions not currently being modeled.
- Interactions between the protocol and other on-chain systems that produce unexpected outcomes.

The protocol commits to ongoing review and to public disclosure of material risks as they are identified.

13.11 The summary disclosure

> Vibestarter is a crowdfunding platform. Tokens obtained through Vibestarter are utility tokens intended for use within their respective project ecosystems. Contributing to a raise is not an investment and does not constitute purchasing a security. Past performance of projects on this platform does not guarantee future results. Contributors should only contribute amounts they can afford to lose entirely.

This statement appears on all public-facing raise pages and is the load-bearing summary disclosure for participants. The detailed risks in this section expand on it. Participants are responsible for reading and understanding both.

14. Appendix

> Glossary, parameter table, contract inventory, and references.

14.1 Glossary

Archived copy · cryp2nova.com

Backer. A person who contributes ETH to a raise. The protocol uses *backer*, not *investor*, because contributions are not investments in the regulatory sense.

Challenge. A formal on-chain action by a token holder, raised against a pending tranche request, that places the holder's staked tokens into escrow pending admin review.

Challenge window. The 72-hour period that opens when a founder requests a tranche, during which holders meeting the graduated threshold may raise a challenge.

Contribution. ETH sent by a backer to a raise's escrow during the active raise window.

Escrow (`VibesTranchEscrow`). The per-raise contract that holds 85% of raised ETH and releases it according to the tranche schedule. One escrow clone per raise, deployed via EIP-1167.

Finalization. The moment a raise transitions from active to funded. Triggers LP creation, escrow setup, and token distribution to the appropriate contracts.

Founder. The person who launched a raise. The protocol uses *founder*, not *issuer*, because no securities are being issued.

Frozen. State of a campaign after a successful challenge or admin freeze. All future tranches are permanently blocked; the holder refund path is open.

Holder refund. The path by which a token holder claims a pro-rata share of frozen escrow ETH. Requires burning tokens to `0xdead` and submitting a merkle proof.

Indefinite LP lock. The mechanism by which 15% of a raise is paired against tokens, deposited in an Aerodrome pool, and the LP receipt locked in a soulbound per-campaign fee claimer. Permanent and irrecoverable by any party; the claimer captures trading fees while the principal stays locked.

Kickstart (T0). The first tranche, 10% of escrow, released immediately on finalization. The only tranche without a challenge window. **Master admin.** The holder of the router's `owner` role. Expected to be a Gnosis Safe multi-sig. Can extract ETH above deposit reserves and alter infrastructure.

Operations admin. A separately-held role appointed by the master admin. Adjudicates challenges, publishes refund roots, and operates per-campaign actions. Cannot extract user funds.

Origin Capsule. The on-chain provenance record for a project, emitted via the `VibesCertified` event by `VibesRegistry`.

Permanent LP. See *indefinite LP lock*.

Raise. A specific crowdfunding campaign on Vibestarter. The protocol uses *raise*, not *ICO* or *token sale*, because the term is descriptive and not a securities term.

Slash. The 20% burn of a rejected challenger's staked tokens, sent to `0xdead`. **Tranche.** A time-based release of capital from escrow to the founder. Seven tranches per raise: T0 (kickstart) plus T1 through

Archived copy · cryp2nova.com

T6 (monthly).

Treasury (`VibesTreasuryEscrow`). The per-raise contract holding the project's treasury token allocation. Withdrawals are proposal-based and challengeable.

Vibecoded. Built primarily with the assistance of AI coding agents. The platform's term for AI-assisted development.

Vibecoin. The token associated with a vibecoded project. Per-raise; not a single protocol token.

14.2 Parameter table

Parameter	Value	Where set	Adjustable?
`KICKSTART_BPS`	1000 (10%)	`VibesTranchEscrow`	No
`MONTHLY_BPS`	1500 (15%)	`VibesTranchEscrow`	No
`NUM_MONTHLY_TRANCHES`	6	`VibesTranchEscrow`	No
`TRANCHE_DURATION`	30 days	`VibesTranchEscrow`	No
`CHALLENGE_WINDOW` (filing + payout delay)	72 hours	`VibesTranchEscrow`	No
`ADJUDICATION_WINDOW` (filed challenge → auto-expiry; PC-09, post-cut-over raises — $VIBES remains 72h)	7 days	`VibesTranchEscrow`	No
`ADJUDICATION_WINDOW` — treasury (filed treasury-proposal challenge → auto-expiry; PC-10, treasuries created by a post-PC-10 router — $VIBES treasury remains 72h)	7 days	`VibesTreasuryEscrow`	No
`CHALLENGE_SLASH_BPS`	2000 (20%)	`VibesTranchEscrow`	No
`CHALLENGE_COOLDOWN`	7 days	`VibesTranchEscrow`	No
`CHALLENGE_THRESHOLD` (T0–T2)	0.25% of supply	`VibesTranchEscrow.getChallengeThreshold()`	No

CRYP2NOVA

Archived copy · cryp2nova.com

Parameter	Value	Where set	Adjustable?
`CHALLENGE_THRESHOLD` (T3–T4)	0.50% of supply	`VibesTranchEscrow.getChallengeThreshold()`	No
`CHALLENGE_THRESHOLD` (T5–T6)	1.00% of supply	`VibesTranchEscrow.getChallengeThreshold()`	No
`platformFeeBps`	250 (2.5%), hard cap 1000 (10%)	`VibesTranchEscrow`	**No** — set once at raise creation, no setter; the factory default for *future* raises is adjustable by the master admin (`setPlatformFeeBps`, PC-07)
`MIN_CONTRIBUTION`	0.01 ETH	`VibesTranchEscrow`	No
`MERKLE_ROOT_DELAY`	24 hours	`VibesTranchEscrow`	No
`MAX_TIME_DRIFT`	1 hour	`VibesTranchEscrow`	No
`LP_BPS`	1500 (15% of raise → LP)	Protocol	No
`ESCROW_BPS`	8500 (85% of raise → escrow)	Protocol	No
`CLIFF` (founder vesting)	180 days	`VibesVesting`	No
`VESTING_DURATION` (founder)	365 days post-cliff	`VibesVesting`	No
`UNSTAKE_COOLDOWN` ($VIBES)	7 days	`VibesStaking`	No
`MAX_RAISE_DURATION`	30 days	`VibesTranchEscrowFactory`	No
`MAX_SCHEDULE_WINDOW`	30 days	`VibesTranchEscrowFactory`	No
`Founder deposit`	0.01 ETH (configurable; refunded on successful finalization)	Router	Master admin (via `setFounderDepositWei`)
`Launch fee (flat ETH, optional)`	0 (off)	Router	Master admin (via `setFeeConfig`)

Parameters marked *No* are protocol-level constants and not adjustable per-raise or by any admin action without a contract upgrade. `setFeeConfig` controls only the router-level flat launch fee; the

CRYP2NOVA

Archived copy · cryp2nova.com

escrow's `platformFeeBps` is fixed per raise at creation, with `setPlatformFeeBps` adjusting the factory default for future raises only.

14.3 Contract inventory

The deployed contract set on Base mainnet (chain ID 8453) at v0.1 of this paper:

Contract	Address	Role
VibesLaunchRouterV2	[TBD]	Main entry point
VibesTranchEscrowFactory	[TBD]	Escrow factory
VibesTokenFactory	[TBD]	Token factory
VibesLPLocker	[TBD]	LP creation and lock
VibesRegistry	[TBD]	Origin Capsule registry
VibesIdentityRegistry	[TBD]	ERC-8004 agent identity
VibesStaking	[TBD]	$VIBES staking
VibesStakerRewards	[TBD]	Staker reward distribution
VibesToken ($VIBES)	[TBD — TGE address]	Protocol token

Per-raise contracts (`VibesTranchEscrow` clones, `VibesVesting` clones, `VibesTreasuryEscrow` instances, `VibesTokenDistributorV2` instances) have one deployment per raise. The full list is queryable from `VibesRegistry` events.

Canonical addresses are queryable on-chain from `VibesRegistry` events and are recorded at each deployment.

14.4 Audit reports

The contracts have had one external audit (ZXVC LLC, May 2026) plus internal review cycles (April and June 2026). All High findings were remediated and re-tested; certain centralization findings (no admin timelocks on the treasury burn / infrastructure setters) are accepted or deferred and disclosed as residual risk. A remediation summary is published at app.vibestarter.xyz/audit.

14.5 Source and verification

The deployed contracts are verifiable on-chain on Base. Contract addresses are queryable from `VibesRegistry` events; the protocol token and per-raise contracts can be inspected on a Base block explorer.

The whitepaper synthesizes; the referenced documents are authoritative for specific details.

Archived copy · cryp2nova.com

14.7 Version history

Version	Date	Notes
v0.1	2026-05-24	Initial draft. Mechanism sections (4–9), system sections (10–13), and this appendix complete. Part 1 (sections 1–3) compressed from the original thesis.

Future revisions will be tracked here. Material changes that affect the mechanism (parameter changes, decentralization-path stage transitions, new failure modes identified) will be reflected in version bumps.

← Back to home

CRYP2NOVA

Archived copy · cryp2nova.com