GitLaunch

Open Source Markets

A market layer for public code, verified maintainers, and recurring creator fees.

Whitepaper · v1.0 · June 2026 · gitlaunch.app · Built on Base

GitLaunch turns public GitHub repositories into transparent markets. Maintainers launch a repo market for free, backers discover and support open-source projects early, and a fixed share of every trade routes back to the people maintaining the code.

Open source already runs the internet. The missing piece is a market structure that lets the people who believe in that work back it directly, while letting maintainers capture recurring fees from the attention, belief, and liquidity around their projects.

This is the category GitLaunch is building: **Open Source Markets**.

Important Notice

This document is informational only. It is not an offer, solicitation, recommendation, or invitation to buy, sell, or hold any token, security, or financial instrument. It is not investment, legal, tax, or financial advice.

Repo-tokens are speculative, volatile, and can lose all value. A repo-token is attached to public, auditable open-source work; it is not collateralized by that work, redeemable for that work, or a claim on that work. USDC refers only to the maintainer payout currency. It is not a promise about token value, token stability, or future returns.

GitLaunch is a market and fee-routing protocol for open-source projects. It does not promise returns to backers, maintainers, or any other participant.

Executive Summary

Open source is one of the largest economic foundations in the world, yet most of the people maintaining it are unpaid. Companies depend on it. AI accelerates it. The internet runs on it. But the default funding path is still donation buttons, grant committees, sponsorship budgets, or nothing at all.

Crypto solved a different problem: permissionless market rails. It created liquid markets anyone could enter, but most of those markets were attached to memes, vaporware, or projects with no inspectable substance.

GitLaunch joins the two.

The mechanism is simple:

1. A maintainer signs in with GitHub.
2. GitLaunch verifies that they control a public repository.

3. The maintainer launches a repo-token market for free.
4. Backers can buy or sell the repo-token through a transparent market.
5. A fixed share of trading fees routes to the maintainer in USDC.
6. The market and repo data are tracked through GitLaunch and gitmarketcap.

The result is a new primitive: a market attached to verifiable open-source work.

For maintainers, GitLaunch creates a recurring fee stream from market activity around their work.

For backers, GitLaunch creates a liquid way to support projects they believe could matter.

For the ecosystem, GitLaunch creates a discovery layer where open-source projects can be ranked, watched, backed, and funded in public.

1. The Problem: Open Source Runs Everything

Open source is the invisible infrastructure underneath modern software. Every bank, exchange, startup, AI lab, hospital, government, and consumer app depends on code that was published in the open.

The scale is no longer philosophical. It is measurable:

- GitHub has more than 180 million developers and more than 630 million repositories.
- 86% of commercial codebases contain open-source components.
- Replacing the open-source software firms already depend on has been estimated at $8.8 trillion.
- 60% of maintainers are unpaid.
- 46% of maintainers report being paid nothing at all.

That is the contradiction GitLaunch is built around: open source creates massive economic value, but the maintainers capture almost none of it.

The world already depends on open source. The market has not caught up.

2. The Old Funding Model Is Too Weak

Open-source funding today is fragmented:

- Donations are generous but small, private, and hard to scale.
- Sponsorships depend on corporate budgets and attention.
- Grants are slow, selective, and episodic.

- Foundations help, but only for a tiny slice of the ecosystem.
- Maintainers often receive attention only after something breaks.

The issue is not that people do not care. The issue is that the rails are weak.

A donation is a one-way transfer. It helps, but it ends immediately. A grant can fund work, but it depends on committees. A corporate sponsorship can be meaningful, but it usually follows adoption instead of helping projects become durable earlier.

Open source has never had a native market.

GitLaunch does not replace donations, grants, or foundations. It adds a missing layer: public, liquid markets tied to verifiable repositories, with fees routed directly to maintainers.

3. Crypto Built the Rails, But Pointed Them at the Wrong Thing

Crypto proved that global, permissionless markets can exist without banks, brokers, or private allocation lists. Automated market makers, public ledgers, token contracts, and programmable fee routing are real infrastructure.

The failure was not the machinery. The failure was what the machinery usually pointed at.

Too often, tokens were attached to a logo, a roadmap, a Discord, and a promise. There was nothing to read, run, audit, or depend on. The market was real. The underlying work was not.

GitLaunch points the same market machinery at something different:

- public repositories
- visible commit history
- dependency graphs
- maintainers with GitHub provenance
- code that people can inspect
- projects already used by real developers

A repo-token is still risky. It can go to zero. But the market is no longer attached to thin air. It is attached to work anyone can verify.

4. The Category: Open Source Markets

An Open Source Market is a transparent market attached to a public repository.

It has four parts:

1. **A real project** - a public GitHub repository.
2. **A verified maintainer** - someone with admin rights to that repository.
3. **A live market** - a tokenized market with transparent rules.
4. **A fee stream** - a fixed share of trading fees routed to the maintainer.

This is not a donation page. It is not a crowdfunding campaign. It is not a token sale for future promises.

It is a market around open-source work.

Backers can enter and exit. Maintainers can earn recurring fees. The public can inspect the code, follow the data, and decide what deserves attention.

That combination has not existed before:

Model	Public repo	Liquid market	Maintainer fees	Verifiable provenance
Donation button	Yes	No	Maybe	Weak
Grant	Yes	No	One-time	Weak
Meme launchpad	No	Yes	Usually no	No
GitLaunch	Yes	Yes	Yes	Yes

5. How GitLaunch Works

The product flow is intentionally simple.

Maintainer Launch

A maintainer signs in with GitHub and selects a public repo they control. GitLaunch verifies repo admin access and deploys the market. The platform covers deployment gas so the maintainer does not need ETH, a wallet setup, or crypto knowledge to launch.

Backer Access

Backers can sign in with email, GitHub, or wallet. GitLaunch can provision a trading wallet for ordinary users, while crypto-native users can connect an existing wallet. Stripe onramp support lets users top up with USDC where available.

Market Mechanics

Each repo-token begins in a launch market. If it reaches the migration threshold, it migrates into a Uniswap V4 pool.

The maintainer receives the same core economic benefit in both phases: **0.6% of every trade**, paid in USDC, whenever trades occur.

Provenance

6. Fee Flow

The fee model is designed to be simple enough to understand and strict enough to enforce.

Launch Phase

During the launch market, each trade carries a 1.25% fee:

- 0.6% to the maintainer
- 0.4% to the platform
- 0.25% temporary gas-cover

The gas-cover component ends after migration.

Post-Migration Phase

After migration to Uniswap V4, the fee is 1%:

- 60% of the fee to the maintainer
- 40% of the fee to the platform

That keeps the maintainer's effective share at 0.6% of every trade.

Anti-Sniper Rule

The launch block includes a one-block anti-sniper fee on buys. It is designed to discourage bots from attacking the opening block. The fee routes to the protocol treasury, not to the maintainer.

7. Why Maintainers Care

GitLaunch gives maintainers a new funding rail without asking them to become crypto founders.

The maintainer does not need to:

- write token contracts
- pay gas
- build a trading frontend
- handle liquidity migration

- create a wallet manually
- run a fundraising campaign

They only need to control a public GitHub repo.

That matters because maintainers are not usually underfunded because their work lacks value. They are underfunded because the value is diffuse. Millions of users may rely on a dependency, but no single user has a strong enough reason to fund it alone.

Markets solve aggregation problems. GitLaunch lets many small expressions of belief, attention, and support gather around a repo and route part of that activity back to the maintainer.

The maintainer's fee stream may be meaningful, modest, or zero. It depends on whether the market attracts activity. But for the first time, the maintainer has a native way to capture recurring value from the attention around their work.

8. Why Backers Care

Backers get something donations never offered: a liquid way to support open-source projects they believe in.

A donation is generous, but it ends after the transfer. A GitLaunch position is different. It lets a backer:

- discover projects early
- support maintainers directly
- participate in a public market
- exit if they change their mind
- follow market and repo data over time

The emotional idea is simple: many important technologies begin as small public repositories. Some become infrastructure. Some disappear. GitLaunch gives backers a way to express conviction in those projects while they are still emerging.

But the risk must stay visible. Backing a repo-token is not a donation and not a guarantee. It is a speculative market position. The token can fall, fail to trade, or go to zero.

The right backer does not buy because returns are promised. The right backer buys because they want a liquid way to back verifiable work they believe should exist.

9. Why GitLaunch Can Win

GitLaunch is not just a launch button. The advantage is the full stack around verified open-source markets.

GitHub-Native Creation

The best source of truth for open-source authorship is GitHub itself. GitLaunch uses GitHub admin access as the launch gate, making provenance legible from the first step.

Retail-Ready Onboarding

Most people will not bridge, manage gas, or write down seed phrases to back an open-source repo. GitLaunch removes those steps with account-based login, provisioned wallets, and USDC onramp support.

Maintainer-Aligned Fees

The system routes a fixed share of every trade to the maintainer. That changes the incentive from extracting attention to funding the person doing the work.

Discovery Layer

GitLaunch pairs markets with repo data: stars, contributors, language, dependencies, activity, fees, market cap, and trading history. gitmarketcap can become the CoinMarketCap-style surface for open-source markets.

Provenance Standard

The `0x...917` suffix, GitHub verification, repo metadata, and contract-level routing give users a way to distinguish real GitLaunch markets from impostors.

Category Timing

Three trends are converging:

- AI is increasing the volume and importance of open-source software.
- Retail crypto onboarding is getting simpler.
- Open-source funding remains structurally broken.

GitLaunch sits at that intersection.

10. The Flywheel

GitLaunch works if the market creates a feedback loop between attention, funding, software quality, and community.

This flywheel is conditional. It can stall. Most repo markets may stay small. Some may see little or no trading activity.

But the loop is structurally different from existing funding models. Donations do not create liquidity. Grants do not create a public market. Meme launchpads do not fund real maintainers. GitLaunch combines these elements into one surface.

The thesis is not that every repo-token succeeds. The thesis is that the best open-source projects deserve a native market, and that market activity should fund the people maintaining them.

11. Built for Retail, Not Just Crypto Natives

If GitLaunch only works for people already deep in crypto, it misses the larger market.

Most potential backers are not trying to become DeFi users. They want to find interesting projects, support them, and understand what they bought.

GitLaunch is designed around that reality:

- email login for ordinary users
- GitHub login for builders
- wallet connect for crypto-native users
- automatic GitLaunch wallet provisioning
- USDC balances in the top bar
- Stripe onramp support where available
- simple buy/sell interface
- clear error states for limits, balances, and liquidity

The goal is not to hide risk. The goal is to remove needless friction while making risk understandable.

Retail users should not need to understand gas, bridges, slippage, private keys, or Base RPCs before backing a repo. Advanced users can still inspect contracts, adjust settings, export keys, and use their own wallet.

The default experience should feel like a checkout. The underlying system remains on-chain.

12. Trust, Safety, and Security

GitLaunch asks users to trust less by putting more in code, provenance, and public data.

The trust model has several layers:

GitHub Verification

Only someone with admin access to a public repository can launch that repo's market.

On-Chain Fee Routing

Fee routing is enforced by contracts, not by a private spreadsheet or manual payout process.

Burned Initial Liquidity

At migration, initial liquidity is sent to the standard dead address, preventing the platform or maintainer from pulling it.

Exportable Wallets

Account-based users receive a GitLaunch wallet. Keys are encrypted at rest and can be exported by the user.

Audit Requirement

Mainnet launch should require independent audit review and published findings.

Future Hardening

Before scale, GitLaunch should add:

- published audit reports
- bug bounty
- multisig treasury controls
- timelocks for sensitive protocol actions
- incident response process
- clear custody and key-management documentation

An anonymous founder increases the importance of these controls. GitLaunch should not ask users to trust a personality. It should ask them to inspect the system.

13. Risks

GitLaunch creates a new market category, and the risks are real.

Token Risk

Repo-tokens can be volatile, illiquid, and worth zero. No return is promised or implied.

Market Risk

Many repo markets may attract little or no trading activity. Maintainer fees depend on activity actually happening.

Regulatory Risk

Token markets, onramps, payouts, and custodial wallets can create legal and compliance obligations. GitLaunch must continue legal review before expanding.

Security Risk

Smart contracts, wallets, onramps, and web applications can fail. Audit, monitoring, and incident response are required.

Maintainer Risk

A verified maintainer may stop maintaining a project. GitLaunch verifies control at launch; it does not guarantee future development.

User Risk

Retail users may misunderstand repo-tokens as safer than they are because the attached repo is real. The interface and documentation must keep risk visible.

The strongest version of GitLaunch is honest about these risks without letting them bury the core idea.

14. What Must Be True Before Mainnet Scale

GitLaunch should not scale on narrative alone. The platform needs proof.

The important pre-scale milestones are:

1. **Independent contract audit**

•Published audit report.

- Resolved critical and high findings.

1. **Named maintainer cohort**

- Real open-source maintainers.
- Public case studies.
- Clear fee and market data.

1. **Security and custody documentation**

- How wallets are created.
- How keys are encrypted.
- How users export keys.
- What GitLaunch can and cannot access.

1. **Onramp and compliance clarity**

- Supported geographies.
- USDC flow.
- Wallet destination behavior.
- User restrictions.

1. **gitmarketcap integration**

- Market cap.
- Fees.
- Volume.
- repo fundamentals.
- provenance.

1. **Public risk language**

- Short, clear, repeated in UI.
- No return implication.
- No claim that code collateralizes the token.

These milestones turn the paper from a thesis into a credible launch path.

15. Platform Economics Questions

This document explains repo markets. If GitLaunch ever raises through a presale or platform token, that requires a separate platform-economics document.

That document should answer:

- What exactly is being sold?
- What value does the asset capture?
- How does platform revenue connect to it, if at all?

- What is the total supply?
- What are allocations, unlocks, and vesting?
- What rights does the asset not provide?
- What happens to treasury revenue?
- What are the regulatory assumptions?
- What is the use of funds?
- What milestones unlock future phases?

Without this, the whitepaper can make users understand GitLaunch, but it cannot make sophisticated buyers evaluate a GitLaunch presale.

The repo-token thesis and the platform-token thesis should stay separate. Mixing them would make both less clear.

16. Long-Term Vision

Open-source repositories are the first market. They are the right wedge because they are public, auditable, high-value, and underfunded.

Over time, the same primitive could apply to other forms of verifiable public work:

- open protocols
- developer tools
- research code
- datasets
- reproducible science
- public infrastructure

This is vision, not a current product promise. Expanding beyond repositories requires legal, regulatory, technical, and institutional review.

The immediate mission is narrower:

Make open-source markets real.

Launch credible repo markets. Pay maintainers. Make discovery useful. Make backing simple. Make the risk clear.

If that works, the category can expand later.

17. The Core Thesis

Open source is real work.

Crypto markets are real rails.

Maintainers are real people.

GitLaunch connects them.

For the first time, a public repository can have a native market. A maintainer can launch for free. A backer can support and exit. A fee stream can route to the person maintaining the code. The market can be watched, ranked, audited, and challenged in public.

The token can still go to zero. The risk is real. But the underlying work is real too.

That is the difference.

Open Source Markets are not about pretending every repo deserves a token. They are about giving the best public software a funding and discovery layer that matches its importance.

GitLaunch is building that layer.

Appendix A: Main Proof Points

The main paper intentionally keeps the number of visible statistics low. These are the proof points that matter most.

Claim	Evidence
Open source is economically massive	Estimated $8.8T replacement value
Open source is everywhere	86% of commercial codebases contain open source
The developer graph is huge	180M+ GitHub developers, 630M+ repositories
Maintainers are underpaid	60% unpaid, 46% paid nothing
The current funding model is weak	Donations, grants, sponsors, and foundations reach only a small slice
Crypto rails can create liquid markets	AMMs, public ledgers, programmable fee routing
Existing launchpads show demand but weak substance	Large fee volumes, low graduation rates

Appendix B: Sources

1. GitHub Octoverse 2025 - "A new developer joins GitHub every second as AI leads TypeScript to #1." https://github.blog/news-insights/octoverse/octoverse-a-new-developer-joins-github-every-second-as-ai-leads-typescript-to-1/
2. GitHub Octoverse 2024. https://github.blog/news-insights/octoverse/octoverse-2024/
3. Black Duck OSSRA 2025 - "86% of Commercial Codebases Contain Vulnerable Open Source." https://news.blackduck.com/2025-02-25-New-Black-Duck-Report-86-of-Commercial-Codebases-Contain-Vulnerable-Open-Source,-Exposing-Organizations-to-Security-Risks

4. Hoffmann, Nagle & Zhou - "The Value of Open Source Software," HBS Working Paper 24-038, 2024. https://www.hbs.edu/ris/Publication%20Files/24-038_51f8444f-502c-4139-8bf2-56eb4b65c58a.pdf
5. HBS Working Knowledge - "Open Source Software: The $9 Trillion Resource Companies Take for Granted," 2024. https://www.library.hbs.edu/working-knowledge/open-source-software-the-nine-trillion-resource-companies-take-for-granted
6. Linux Foundation / Harvard LISH - Census II, 2022. https://www.linuxfoundation.org/blog/blog/a-summary-of-census-ii-open-source-software-application-libraries-the-world-depends-on
7. Tidelift - 2024 State of the Open Source Maintainer Report. https://dev.to/tidelift/60-of-maintainers-are-still-not-paid-for-their-work-35jo
8. Tidelift / PR Newswire - maintainer compensation survey, 2021. https://www.prnewswire.com/news-releases/survey-finds-many-open-source-maintainers-are-stressed-out-and-underpaid-but-persist-so-they-can-make-a-positive-impact-301304925.html
9. The Register - "Open source maintainers underpaid," 2024. https://www.theregister.com/2024/09/18/open_source_maintainers_underpaid/
10. Sonar - "Maintainer burnout is real." https://www.sonarsource.com/blog/maintainer-burnout-is-real/
11. GitHub Sponsors Additional Terms. https://docs.github.com/en/site-policy/github-terms/github-sponsors-additional-terms
12. DeFiLlama - pump.fun protocol page. https://defillama.com/protocol/pump.fun
13. DL News - friend.tech shutdown coverage, 2024. https://www.dlnews.com/articles/defi/friend-tech-shuts-down-after-revenue-and-users-plummet/
14. Dakhel et al. - "4.5 Million (Suspected) Fake Stars in GitHub," arXiv:2412.13459, 2024. https://arxiv.org/abs/2412.13459
15. X Engineering - Twitter recommendation algorithm open source announcement, 2023. https://blog.x.com/engineering/en_us/topics/open-source/2023/twitter-recommendation-algorithm
16. xAI - Grok-1 open release, 2024. https://x.ai/news/grok-os
17. Meta AI - Llama ecosystem update, 2024. https://ai.meta.com/blog/future-of-ai-built-with-llama/