



Forked.gg



DECENTRALIZED INFRASTRUCTURE FOR ENTERTAINMENT THAT DESERVES TO KEEP LIVING

The Forked.gg ecosystem: durable infrastructure, predictable economics, player-owned assets, and governance run by the people who run the network.

EXECUTIVE SUMMARY

Forked.gg is decentralized infrastructure for keeping games and live entertainment alive, built by people who worked inside large web3 gaming companies and watched the usual failure modes up close. We would rather run a smaller system that holds than a bigger one that impresses for a quarter and breaks.

What it is

One network, and the things that run on it. The network is the product: independently owned machines doing real, paid infrastructure work for games, content delivery, storage, live game hosting, oracle data, AI inference and GPU compute, running live today. This is decentralized physical infrastructure, the category the market calls DePIN, pointed at an industry that needs it badly and currently rents it from three hyperscalers.

Everything else in this document sits on top of that network. The products a player touches without knowing a chain is involved: hosted game servers, a cross-store game library, browser games, media streaming, and one account across all of it. Aegis, open-source governance where it is one node, one vote, never weighted by token holdings. A DAO Launchpad so a community can take over a game that lost its stewards and keep it alive. FORKcast, a live prediction engine that runs on any stream you can link. And a marketplace with a free genesis collectible drop for the community.

What we are trying to do

Raise the floor for how web3 games are built, governed, and kept alive. We sell infrastructure that games genuinely need, priced in ordinary money and set below the major cloud providers, and we let a token fund and settle that work without being anyone's payday.

Why the economics are clean

Forked.gg was built from a blank page: its own technology, and a token model designed from scratch. The team holds no token at all. Its only upside is equity in the operating company, so no insider is ever a structural seller of \$FORK. The operators who ran the network first are honored at a fixed 50 LIX to 1 FORK, vesting over six months, because they did real work before there was anything to show for it.

Where the value comes from

Node operators are paid in \$FORK for verified work, never in cash, with the reward sized to the value of the work they do. Most of every service payment is used to buy \$FORK on the open market and route it to the operators who did the work, so real network usage, rather than speculation, is what supports the token over time. The company buys \$FORK and never sells it to fund itself. That demand grows as the network earns; it is not a promise about price.

What is live now

The metering backbone runs in production today, against 3,278 legacy licenses of which roughly 800 have been online. On top of it: hosted Minetest and Factorio game servers that community members pay credits to run, a cross-store game library of about nineteen thousand titles, uploaded browser games, encrypted media sharded across the fleet and streamed back, and a single account spanning all of it. FORKcast is live and cred-only. The price oracle is live. The whole token economy, including Aegis governance and operator settlement, is deployed and proven end to end on a public test network, with mainnet and an independent audit still ahead. What is planned, and what is still uncertain, is marked plainly throughout this document.

FROM THE CHIEF ADVISOR

I am an advisor to Forked.gg, and I will be honest that I have had more influence on the tokenomics here than an advisor usually does, or usually should. I have been called the give-it-all-away-for-free guy more than once, and I understand why it sounds that way. What you are reading is the result of more than twelve years in this industry, across more projects than I can count, watching them rise and watching them fall.

For a lot of those years I was the person screaming silently in the background while bad tokenomics shipped anyway, again and again, with the same predictable ending. Tokenomics are hard. Getting this right is hard. I am not going to pretend otherwise. What I will say is that by building something that genuinely aligns everyone, the team, the operators, and the people holding the token, we give ourselves a better shot at lasting than anything I have seen put in front of the market so far.

I am not interested in dominating this space. If we stay focused on building what games and gamers genuinely need, we can lift the whole ecosystem rather than try to own it. That is the bet, and I am willing to put my name on it.

To their real credit, the Forked team has been remarkably open to trying something new here. Everyone on this team has seen the wreckage of the last few cycles, and everyone is willing to try a radical answer rather than live through it one more time. That willingness is rarer than it should be, and it is part of why I think this one has a chance.



Jason "BitBender" Brink

Chief Advisor, Forked.gg

ABOUT THIS DOCUMENT

Note on this document

This is **Version 2** of the Forked.gg white paper, published for community and public review. It describes the direction of the ecosystem and the thinking behind it. Figures, timelines, incentive structures and system models remain **provisional and will change** as the system is built and reviewed, and the numbers that are fixed at the token generation event are named as such where they appear.

Forked.gg put nodes into operators' hands before it had a finished product to show them. The people who took one on and kept it running are honored in \$FORK. Section 2 explains how.

Legal notice

Participation involving digital assets and tokens carries risk, including total loss. This document is for general informational purposes only. It is not financial, legal, tax, or investment advice, and it is not an offer or solicitation to buy any asset. Nothing here is a promise of profit, yield, or future value. Do your own research and consult qualified professionals before participating. Forked.gg makes no representations or warranties and disclaims responsibility for losses. See the Risk Factors section for a fuller disclosure.

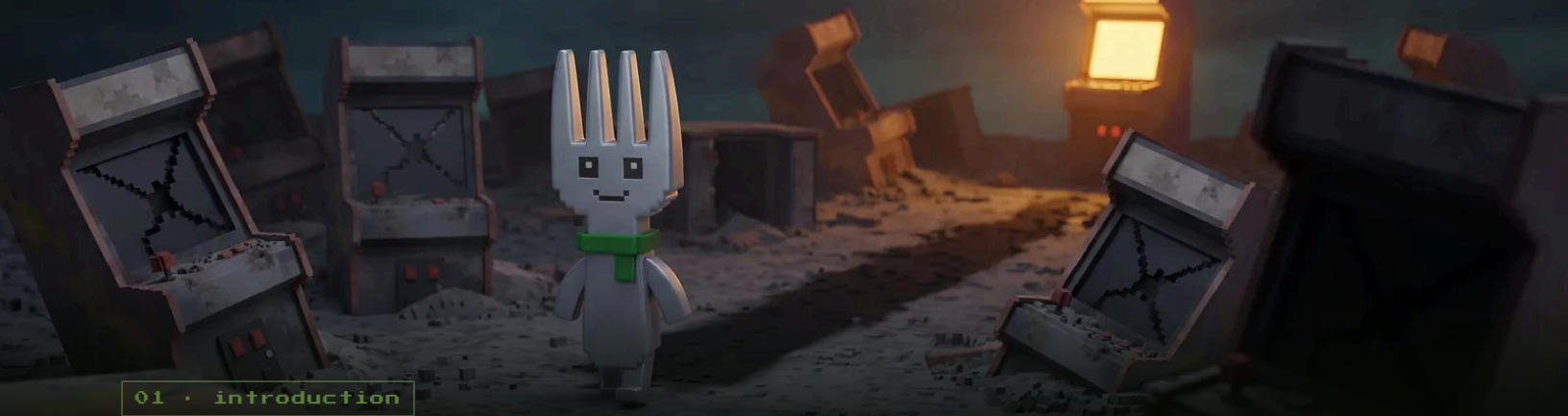
The corporate entity is being formed and is with counsel now. The structure that will hold the operating business, the intellectual property, and any token issuance is not complete, so this document deliberately does not name it rather than print something that changes. **No node licenses are offered or sold until that formation is finished.** The entity appears here once it is settled.

DRAFT STATUS

This is **Version 2**, and the only version that should be referenced going forward. Every earlier version is superseded. The *issuing entity is still being formed*, and the reward rate card is not final; both are flagged where they appear. Everything here is subject to legal review before any external use.

TABLE OF CONTENTS

	Executive Summary	02
	From the Chief Advisor	03
01	Introduction	06
02	The Operators Who Came First	07
03	The Forked Team	08
04	Core Tenets	09
05	What Makes Forked Different	10
06	The Node Ecosystem	11
07	What Runs on the Network	23
08	Aegis Governance Platform	26
09	The DAO Launchpad	27
10	FORKcast, the Predictive Engine	32
11	The Forked.gg Marketplace	35
12	Roadmap	37
13	The \$FORK Token Model	39
14	Go-to-Market & Adoption	55
15	Risk Factors	57
16	Conclusion	59



01 • introduction

INTRODUCTION

Web3 gaming spent its first wave burning trust. A lot of projects launched with loud token designs and quiet products, optimized payouts for early extraction instead of long-term play, and called themselves decentralized while a small group held every real decision. Players and developers who showed up early often paid for that.

Forked.gg starts from the wreckage and asks a narrower, more useful question: what does it take to keep a good game alive and fair over years, not months. The answer we keep arriving at is boring on purpose. Durable infrastructure. Predictable economics. Ownership that the player actually controls. Governance that real participants run. We would rather build a smaller system that holds than a bigger one that impresses for a quarter and breaks.

That answer put us in the infrastructure business, and the infrastructure is the center of this document. Games run on servers, storage, content delivery and increasingly on GPUs, and today almost all of that is rented from a handful of hyperscalers at prices set by them. A decentralized physical infrastructure network, DePIN, replaces those rented racks with independently owned machines that are paid for verified work. It is a well-understood category, it is the part of web3 with the clearest reason to exist, and nobody has pointed it squarely at games. That is what we are building. Section 6 describes the network and Section 7 describes what already runs on it; everything after those two sections exists because the network does.

One capability sits at the center of why we exist. Plenty of games had real communities and real mechanics and still died, not because the game was bad but because the economics or the operations around it were misaligned. Forked.gg gives projects like that another foundation to stand on: shared infrastructure, economic frameworks that are designed and reviewed rather than improvised, and tooling that points decisions back toward the people who play.

There is one more thing worth saying plainly, because it is the part of this design that has not really been tried before. Most token models of the last cycle would have been far more solid if the team had not kept its boot on the neck of the token. When a team holds a large allocation or takes a large percentage of the distribution, it sets the team against the node operators and the token holders, because the team has to keep selling to fund itself, and every one of those sales works against the people holding the token and building on it. We took that out. The team earns nothing from the token and makes money only if the product as a whole makes money, which means node operators earning real income and token holders recognizing the value of what has been built around the token. With no team supply to dip into, everyone is finally pulling in the same direction. It leaves the team carrying more uncertainty, since there is no reserve to lean on, and that is a risk we are willing to take for the sake of tokenomics that actually hold.

THE OPERATORS WHO CAME FIRST

A node network is only as real as the machines running it. Forked.gg put nodes into people's hands before it had a finished product to show them, because the alternative was a diagram. Those operators ran real infrastructure for roughly a year, and they are honored in \$FORK. This section sets out exactly how.

What the first year did

Operators bought nodes or claimed them through their affiliation with other projects, kept them online for roughly a year, and earned rewards for the uptime they delivered. Two things ran in production: a token on BNB Smart Chain, and a fleet of nodes doing IPFS pinning inside third-party node software. **Every reward was earned by verified uptime and recorded in a public registry**, which is why the figures in Section 13 are measured on-chain rather than estimated, and why any holder can trace what was earned, what has been paid, and what is still owed.

Built from scratch

Every system described in this document was designed and built by Forked.gg. The node agent, the metering and settlement backbone, the storage engine, the streaming layer, the consumer products, Aegis, the DAO Launchpad, FORKcast, the marketplace. None of it is inherited code, and none of it depends on the software that fleet originally ran.

How the honoring works

Holders of the legacy token (LIX) can convert into \$FORK at a fixed ratio of **50 LIX to 1 FORK**. This is a straight redenomination: it preserves each holder's proportion of the supply without depending on any price. Every genuine on-chain position converts at that ratio. The early backers hold **new standalone Forked.gg agreements** instead, their earlier ones having lapsed unexercised. To protect the market at launch, converted balances vest over six months rather than unlocking at once, with the option to claim early in exchange for forfeiting the unvested remainder. **The node licenses are honored too:** the 3,278 already issued each convert into a Forked node license one for one, so somebody who earned the right to run a node keeps it without paying twice.

Designed from a blank page

Starting clean also let us leave out the thing that quietly sinks most token projects: a large team allocation that turns the team into a permanent seller. **Forked.gg has no team token allocation at all.** The team's only upside is equity in the operating company, which leaves the token free to do one job, keeping the network and the community healthy. Section 13 describes the result in full.

THE FORKED TEAM

Forked.gg is built by people who worked inside large web3 gaming organizations and watched the failure modes up close. The operating philosophy is stewardship over ownership: keep the core infrastructure healthy, provide continuity for the games and communities that depend on it, and move decision-making toward the community over time.



Jared Dillinger • CEO

Jared Dillinger is a gaming entrepreneur, publisher, and ecosystem builder whose background spans professional sports, gaming, AI, creator media, and web3. As CEO, he leads Forked's focus on building the infrastructure that games run on, and on the partnerships that bring studios and communities onto it. Before Forked he held leadership and advisory roles across gaming and blockchain, including work with Gala Games, The Sandbox, and emerging gaming ventures across Asia, covering ecosystem growth, creator economies, partnerships, esports, and community-led product. A nine-time professional basketball champion and former member of the Philippine national team, he has built one of the largest athlete-driven digital communities in the country.



Christos Segkos • CMO

Christos Segkos is a fintech and web3 executive with over six years of leadership experience across digital banking, blockchain infrastructure, and decentralized finance. He has held C-level roles in high-growth ventures at the point where traditional finance meets the next era of digital assets. Since entering crypto in 2017, he has helped launch and scale multiple projects with token valuations surpassing \$100 million, combining technical fluency with strategic market insight and a blend of institutional discipline and startup agility.



Levi Hansen (ChairBandit) • COO

Levi (ChairBandit) is co-founder of Forked, a platform designed to merge the best of web2 and web3 gaming. He brings more than ten years in blockchain and web3, across decentralized systems, tokenomics, and community building. His earlier background as a Director of Customer Support and a Product Manager in traditional game development gives him a clear view of what mainstream players expect, which drives Forked's approach of weaving ownership, rewards, and interoperability into familiar gaming experiences.

Advisors



Jason Brink



Patrick Bergman

Two further advisors

Jason Brink serves as Chief Advisor to Forked.gg and Patrick Bergman as a founding advisor, joined by two further advisors who are named once their agreements with the new entity are executed. **Neither Jason Brink nor Patrick Bergman holds any token allocation.**

CORE TENETS

These are what we believe a good system should support and work toward. They are a moving target rather than a checklist we have already ticked: the ability to achieve them is a journey, some are considerably harder than others, and a few will take years of building before we can say we have got there. We hold to them as ideals and we would rather be measured against them honestly than quietly lower the bar.

- **Player ownership.** In-game assets exist as player-controlled primitives anchored to blockchain standards. Custody, transfer rights, and long-term control stay with the player.
- **Decentralized decision-making.** Anyone in the ecosystem can put a proposal up and argue for it. Binding votes are cast one node, one vote, never weighted by holdings, and smart contracts execute the outcome once a threshold is met.
- **No permanent central authority.** Control over rules, economics, and access is distributed across operators, developers, and players.
- **Community as a real stakeholder.** Mechanisms to influence priorities, features, and direction are operational, not decorative.
- **Transparency by design.** System changes, governance outcomes, and economic parameters are visible and auditable.
- **Fair play.** Reward mastery, coordination, and strategy. Economic advantage is bounded by design, and we say plainly where money does and does not buy an edge.
- **Continuous, stable innovation.** Ship improvements without breaking the things people rely on.
- **Participation rewarded broadly.** Governance, infrastructure, testing, feedback, and community leadership earn standing, not only financial expenditure.
- **Respect and inclusion.** Clear standards of conduct for everyone in the ecosystem.

WHAT MAKES FORKED DIFFERENT

- **Enforced on-chain.** Emission and allocation are governed by publicly verifiable smart contracts, with safeguards against excessive accumulation. The contracts are deployed on BNB Chain testnet today, with the mainnet set and its audit published before the token generation event. An **independent third-party audit is commissioned before any mainnet deployment**, and no mainnet contract goes live without it. No audit has been completed at the date of this draft.
- **No team token allocation.** The team holds no portion of \$FORK. Its upside is equity in the operating company, so no insider is ever a structural seller of the token. This is the single biggest difference from the models that came before it, and it lets the token be designed for the ecosystem rather than for a payout.
- **Predictable emissions.** Issuance is deterministic and governed by smart contract, released from a capped reserve on a declining schedule, so operators and developers can plan against a known curve.
- **Supply cycles, it is not destroyed.** Tokens spent inside the ecosystem return to the Emissions Reserve to be earned again by operators doing work, rather than being destroyed for effect. We do not run the burn-to-pump playbook of the last cycle. This manages how much supply sits in circulation and is not a promise about price or value.
- **Community infrastructure.** Communication channels (Discord, Telegram, X, and others) are treated as core infrastructure for coordination, not as a marketing afterthought.
- **Our own verification technology.** The mechanisms that let untrusted machines prove work and get paid for it are ours, built here rather than assembled from other people's protocols. Two applications covering the node verification layer and the nested-shard streaming layer are prepared and with patent counsel. Nothing is filed yet, and we will not describe either as patent pending until it is.
- **Chain-flexible architecture.** Core logic is decoupled from any single chain to reduce dependency risk and improve interoperability. The points ledger is built chain-agnostic and anchor-ready (append-only, deterministic, hash-chained), so per-market or per-epoch state can be anchored on-chain for tamper-evident public audit. The first token deployment target for \$FORK is BNB Chain; the system is not bound to it.

Forked.gg has no plan to launch its own chain or L1. There are already more than enough L1s and L2s, and the world does not need another one. The goal is to stay chain-agnostic, to deploy where it makes sense, and to let anyone who wants to build on Forked do so without being locked to a single network.

THE NODE ECOSYSTEM

Nodes are the infrastructure layer underneath the core services. They distribute operational responsibility (content availability, hosting, CDN, platform services) across independent operators for resilience, uptime, and censorship resistance.

- **A supply of 200,000 node licenses**, each issued on-chain as the right to operate a node and perform real work for the network. That ceiling can be raised only by a vote of the operators themselves, on the terms in Section 6.
- **Distribution favors long-term operators.** Licenses carry transfer restrictions during an initial stabilization period. After it, they may be transferred only to a party who will actually operate the node. **A license is a right to do work, and it moves to someone who intends to do that work**, which is a different thing from a freely tradable instrument and is meant to be.
- **Operators are rewarded for service**, measured by availability, consistency, and reliability. Rewards are tied to work performed, not to passive holding.

Running a node

A node runs on a standard VPS or a capable home machine. These are the **base** requirements. They will rise as the network grows and gets more demanding, and when they do we give **at least 30 days notice before any increase takes effect**, so nobody wakes up to a machine that stopped qualifying overnight. We also intend to keep a place for low-powered machines to contribute: raising the floor is about capability the network genuinely needs, never about pushing out the small operator.

RESOURCE	BASE REQUIREMENT
CPU	4+ cores
RAM	8 GB
Storage	60 GB
Operating system	Windows, Mac, or Linux (64-bit)
Internet	Stable connection

The node agent is a single self-contained program with no external dependencies. It installs in one step and its setup instructions ship with it, so an operator never has to assemble a stack.

NODE LICENSES

A license is the right to operate a node on the network. It is a piece of software, priced in dollars. **Buying one grants no tokens, no yield and no revenue share.** The only way any token reaches an operator is by doing real work for a paying customer and having that work measured, proved and audited. No work, no payment, however many licenses are held. **No license is offered or sold until the corporate entity is formed;** that work is with counsel now. **The company can afford to wait, and it is worth saying why:** nobody is drawing a salary, the operating cost is servers, and that cost falls as node operators take the work over. A design that releases nothing until demand arrives only works for a company that does not need the sale, and that is the position we are in today rather than a claim about the future.

LICENSE PRICING

Licenses are priced in US dollars and paid in stablecoins. Two things govern the sale, and both are arithmetic on published on-chain numbers rather than decisions anyone makes month to month.

How many are available. 25 licenses a month are available irrespective of demand. If demand grows past what the online fleet can serve, more are released instead: the network reports settled service revenue on-chain each epoch, and if that revenue would support more capacity than is currently online, the shortfall is released, up to **5% of the live fleet per month**. That ceiling **contains the 25, it does not sit on top of them:** at the 800 nodes online today it means a maximum of 40 licenses in a month, not 65. Nothing is lost to that ceiling. Unmet demand carries into the following month and is worked off. Demand governs the total; the ceiling governs only the speed. Supply is capped at **200,000**. When they are gone they are gone, and the only thing that can change that is a vote of the operators, under the conditions set out three pages on. **We cannot raise it ourselves at any turnout.**

Available is not sold. Twenty-five a month is what we put on the shelf, not what anyone has to take. **3,278 licenses are already issued** and honored from the network's first year, which is more capacity than current demand justifies, so the demand-driven half of the rule releases nothing until service demand grows into that standing fleet. Everything in this document that depends on license sales is therefore stated as a **ceiling rather than a forecast**, and the token figures in Section 13 assume **no license sales at all**.

Emissions are paid on at most 100 nodes per operator. Anyone may own more licenses than that; the cap binds at the point rewards are paid, not on what you are allowed to hold. **It binds on a verified operator rather than on an address**, through a proof-of-personhood check rather than identity documents, because a per-address cap costs one transaction to multiply and would not be a cap at all. The same check gates voting.

ON THE SEC LETTER

The 2025 SEC staff no-action letter on DePIN token distributions[†] informs how we structure **emissions and work-based rewards**, which is the only thing it is cited for here. **It is not authority for the sale of node licenses**, we do not hold it out as clearing that sale, and nothing in this document should be read as a regulatory opinion on the license itself. Rewards are never a promise of yield or profit.

[†] SEC Division of Corporation Finance, DoubleZero no-action letter (September 29, 2025): [sec.gov/rules-regulations/no-action-interpretive-exemptive-letters/division-corporation-finance-no-action/doublezero-092925](https://www.sec.gov/rules-regulations/no-action-interpretive-exemptive-letters/division-corporation-finance-no-action/doublezero-092925). SEC statement on the letter: [sec.gov/newsroom/speeches-statement-s/pierce-092925-deep-statement-doublezero-no-action-letter](https://www.sec.gov/newsroom/speeches-statement-s/pierce-092925-deep-statement-doublezero-no-action-letter).

WHAT A LICENSE COSTS

What they cost. The price starts at **\$1,700** and rises only as the network proves it is being used. It is tied to utilisation, on the same on-chain measure that governs release: at launch utilisation the price sits at the \$1,700 floor, and at the **80% target** it reaches a **\$5,000 ceiling**, moving in a straight line between the two on a trailing twelve-month average. **The price can fall, in two ways, and neither is at our discretion.** If utilisation settles below the level that set the current price, the price **drifts down** toward what current use justifies, closing a tenth of the gap each month. It never goes below what use supports, so this is not a price that collapses on a bad quarter; it just stops charging for a busy period that is over. Separately, any month with **no licenses sold at all** takes 5% off, which can carry the price below the utilisation line and stops at the \$1,700 floor. Rising utilisation always wins: the second rule only applies when use is flat or falling. It is not a date ladder and not a volume ladder. What you pay depends on how much work the network is actually doing when you buy, which is the only thing that determines what a license is worth. The price you pay is fixed at purchase.

Where the money goes

65% of every purchase goes back into the network, not into our pocket. Where it lands depends on one thing. Until the liquidity fund reaches **\$500,000**, that 65 splits **35% into the Emissions Reserve**, the same pool operator rewards are emitted from, and **30% held in stablecoin to make a market in \$FORK**, because a token nobody can trade at a fair price serves no one. Once the fund is full the slice stops and **all 65% goes to the reserve. The fund fills as licenses sell, and not before.** We are not going to promise a date for it, because that would be promising our own sales. At the 25 available each month and the \$1,700 floor, the 30% slice raises \$12,750 in a month where all of them sell, so **\$500,000 takes about 39 months at full take-up and longer at any rate below it. It is not funded out of the company's 33%**, which is what builds node services and is thin already. **Read that against the release schedule on the previous pages, because the two do not line up.** The honored operator layer completes inside six months, and on that day the fund holds around \$76,500 even in the case where every license sells, which is roughly 15% of the target. **The market-making side is thin exactly when the largest release lands**, and it fills slowest in the case where demand is weak, which is the case where a thin book hurts most. We would rather set that out here than let it be discovered. It is the sharpest unresolved tension in this document.

2% goes to the operator-controlled DAO, the same cut it takes of service revenue. The remaining **33%** is company revenue, used to build out node services and platform functionality. Service revenue is split on the same principle but in different proportions, set out in Section 13.

WHEN THE NETWORK RUNS OUT OF ROOM

Why the release rule is shaped this way

A calendar ladder charges more as time passes whether or not anything has been built, and it issues capacity on the same schedule. This one cannot do either. If demand does not arrive, the price stays at \$1,700 and the network does not put nodes into service that nobody is paying to use. **Supply follows demand, and so does price.** They run off the same measured number, so there is only one thing to verify and only one thing to argue about.

That leaves one case to answer for. If demand grows faster than **5% of the fleet a month**, the ceiling holds supply back, the network runs at its limit, and customers get turned away while licences that would serve them sit unreleased. A rule that cannot handle its own success is not finished.

The capacity vote

When the network is at capacity and the ceiling is what is blocking release, a vote opens **automatically**. Not when we decide to call one: the trigger is measured on-chain, so the timing is not ours to choose. It asks node operators one question, whether to release more licences than the ceiling would otherwise allow.

WHAT HAPPENS IF NOBODY VOTES

Quorum is **25% of active nodes** on a rolling thirty-day average, the same bar and the same denominator Aegis uses for any other proposal, and the same eligibility test: online, running an approved workload, with at least 80% historical uptime. **Only operators actually running the network vote on how much more network to sell.** Reaching quorum, the vote decides, up to **20% of the fleet in one round**. Below quorum, **we decide**, and what we may decide alone is bounded far tighter: no more than **5% of the fleet in that round**, and no more than **10% of the fleet across any twelve months**.

The bounds run that way round on purpose. **Apathy should be the conservative outcome, not the permissive one.** The most we can do without operators is a fraction of what an engaged vote can authorise, so a quiet network stays tight rather than becoming ours to open. Turnout and outcome are published every round, including the rounds we decided.

This is deliberately outside the Aegis governance structure described in Section 8. Aegis governs the platform. This governs the one thing operators have a direct financial stake in that the platform does not touch, which is how many of them there are.

THE CEILING ON LICENCES

The capacity vote governs how fast licences reach the market. It does not govern how many exist altogether. That number is **200,000**, and every licence ever issued counts against it, including the 3,278 already honored.

We have been asked, fairly, why a network should have a ceiling on its own size at all. The question behind it is whether a cap on licences is a cap on how much the network can ever do. **It is not**, and the reason is easy to check rather than take on trust. What a single node contributes is not fixed. An operator who takes on heavier work earns on a capability multiplier, so one machine's contribution rises without a new licence being issued anywhere. **The ceiling bounds how many operators there are. It does not bound what they can do**, and those are different numbers.

It is also not what constrains us. At any sales rate we are able to model, 200,000 is decades out. The live constraint runs the other way: **3,278 licences are already issued** against demand that supports a fraction of them, which is why the demand-driven half of the release rule is releasing nothing at all today.

IF THE CEILING EVER BINDS

A rule that cannot handle its own success is not finished, and that applies to the ceiling as much as to the speed limit. So there is one path through it, and it is deliberately narrow.

A vote may raise the ceiling. It opens only once **90% of the standing ceiling has been issued**, the network is running at its serving limit, and demand is still going unmet, all three holding for six consecutive epochs. **Quorum is a third of active nodes and approval takes two thirds of votes cast**, both higher than any other vote in this document, because this is the only decision here that cannot be undone. A licence, once issued, cannot be recalled. One approved vote may raise the ceiling by at most **25%**, and no second vote may open for **twelve months** after one passes.

Below quorum, nothing happens. There is no fallback to us, at any turnout, in any circumstance. That is the opposite of how the capacity vote is built, and the difference is deliberate: the capacity vote moves a *rate*, and a rate corrects itself as the fleet grows into it. This moves a *total*, and a total does not correct. **Where we can act alone on speed, we can never act alone on size.**

WHAT RAISING IT WOULD NOT DO

We would rather state the limit of this than have it read as more than it is. **Raising the ceiling would not let the network serve meaningfully more customers.** The 5% monthly speed limit is what binds when demand outruns supply, not the ceiling, and our own modelling of a raise moves unserved demand by low single-digit percentages at best. What a raise restores is the flow of licence proceeds back into the Emissions Reserve, the 65% recycle described on the previous page, which stops when the ceiling is reached. **It is a valve on that recycle, not a fix for turning customers away.**

We are not going to attach a figure to what a raise would be worth. Every scenario in which the ceiling actually binds sits outside the range our model can price honestly, and a number produced there would be decoration.

STAKING TO SERVE CRITICAL WORK

Most workloads need no bond. A node that goes offline earns nothing, so failure already punishes itself and there is nothing to add. A handful of workloads are different: if they go wrong quietly, they damage *other people*. An oracle that signs a bad price corrupts every settlement in that window, and the operator who did it may not even notice. Those workloads require a stake to activate, and that stake is at risk.

CAPABILITY	STAKE TO ACTIVATE	WHY
Base availability, content delivery	none	Failure is self-punishing; worst case is a slow byte
Storage, proof of possession	3,000	Erasure coding survives a lost shard; repeated loss costs real money to heal
Live game hosting	7,000	A dropped world costs a paying customer their session
RPC	7,000	Wrong chain data breaks integrations silently
Oracle and settlement	14,000	A wrong price corrupts every settlement that hour. Highest blast radius in the system
Anchor, high-trust coordination	29,000	Holds coordination state that other nodes depend on

Stakes are shown in \$FORK and hold a constant economic value, so the token amount is **re-struck once a quarter** against the volume-weighted average price across the deepest venues over the preceding thirty days, computed by the settlement agent and published with the inputs. Disputes go through the slashing-dispute route, holding at the prior amount until resolved, and **no re-strike may move a bond more than 25%**. An observed market rate, not a price we publish. The oracle bond is sized against settlement volume rather than a flat figure.

What happens when it goes wrong

- **First failure: no slash.** The capability is suspended and the operator re-qualifies before it switches back on. Most failures are a bad disk or a bad week, not malice, and treating them as theft would be both wrong and a good way to lose operators.
- **Repeat inside 30 days:** a partial slash of **10%** of the bond, and the capability stays suspended until re-qualified.
- **Provable malice**, such as signing a price the node knew to be false or colluding on a challenge: **the full bond**.
- **Sustained one-sided drift:** a statistically significant one-sided deviation from the independent reference feeds, held over a stated window, is slashable in itself even when no single reading ever breaches the review band. A small persistent bias is profitable, because operators receive more \$FORK for every dollar of service revenue when the printed price is low, and it is neither an accident nor a single provable lie.
- **Slashed stake goes to the Emissions Reserve, never to the company.** We should not profit from punishing an operator, and this way the value returns to the people still doing the work.
- **Appeals run through Aegis** once governance is enacted. Until then, any slash above the partial tier requires published evidence.
- **When the oracle gate halts settlement, the procedure is published rather than improvised.** A named reviewer is on call, **operator accrual continues throughout the hold** so nobody loses earnings to a pause they did not cause, the maximum time to clear is fixed in the operator agreement in advance, and every trip is logged publicly with its cause and duration.

THE FLEET, RUNNING LIVE

The node software is real and running today across two fleets plus the coordinators that drive them. Those are our own machines. The operator base they grow into is a separate and larger population: **3,278 licenses were issued under the earlier project**, and roughly 800 of those machines have been online. The model in Section 13 ramps from the 800 toward the 3,278. Every node runs the same agent, and the agent is ours: a single program with no external dependencies that measures the work a node does, signs the measurement, and settles it. Earnings come out of one formula, units of work multiplied by the weight of that workload, and the credit ledger behind it is hash-chained and re-executed from the beginning every time the coordinator boots, so code that has drifted cannot quietly change history.

Storage does not lean on anyone else's network either. A file is split into shards, compressed, encrypted, spread with erasure coding so it survives lost nodes, and committed to a Merkle tree. Operators are paid for gigabyte-months they can prove they are actually holding, and a damaged shard can be healed by nodes that never see the plaintext. The same shard machinery, taken down to the fragment, is what streams media across the fleet over the open internet today. The control plane below is the content-availability view of that fleet, with each node's status, pin count, and last-seen heartbeat.

Forked · Node Control LIVE 0x7Fa9...3c21 · Forked admin

NODES THAT FORKING DO THINGS

The Forked fleet runs IPFS today. This is the control plane for what it pins, serves, and powers, starting with the storage and CDN layer behind marketplace.forked.gg. Uptime keeps reporting to the LFG registry, untouched.

9 / 9
Nodes online

6 cids
Objects pinned

210 pins
Total replicas

35.0 ./obj
Avg replicas / object

FLEET

9 shown

All
Online
Offline

NODE	STATUS	PINNED	OK PINS	LAST SEEN
0x154bc2...	online	6	332	2s ago
0x10d124...	online	6	327	2s ago
0x165727...	online	6	324	2s ago
0xfe6453...	online	6	333	2s ago
0x84d334...	online	6	328	2s ago
0x244d9f...	online	6	326	2s ago

LIVE WORKLOAD

IPFS PINNING & REPLICATION

Every node runs Kubo and pins the content set you publish from the master. Polls every 5 min, reports oks and fails. Gateway is live on each node at :8080.

• Uptime still reports on the separate health wire to the LFG registry. This plane does not touch it.

WHAT THEY CAN DO

Pinning & storage
decentralized object store

ACTIVE

Gateway / CDN

BUILDING

FORKED NODE CONTROL • THE LIVE FLEET, CONTENT-AVAILABILITY VIEW

WHAT A NODE CAN DO

Beyond keeping content available, a node can take on optional, opt-in workloads. The network is a two-sided marketplace: games and developers need real work done, operators have spare capacity to do it, and putting that capacity to use is the whole point. An operator runs only the workloads their hardware supports and that they choose to accept.

The power ladder

Workloads sort onto a hardware ladder, from the light base tier up to heavy compute. Operators climb it as far as their machine and their appetite allow.

TIER	ADDS	TYPICAL WORK
T0 Base	4 cores / 8 GB / 60 GB	Availability and content delivery
T1 Storage	More disk	Preservation and archival hosting
T2 Compute	More CPU / RAM	Web3 and economic services, indexing
T3 GPU	A GPU	AI inference, media transcoding
T4 Anchor	Strong, well-connected host	High-trust anchoring and coordination

The workload catalog

Ordered from lightest to heaviest: content delivery (game clients, patches and media served close to players); preservation (sunset or community game data on durable decentralized storage); identity and social services; web3 and economic services (indexing, settlement helpers, on-chain data); live game hosting (dedicated multiplayer servers near players); AI inference; and streaming and heavy compute (transcoding and other GPU-bound work).

Four of these carry real traffic today: storage with proof of possession and paid gigabyte-months, media streaming across a twenty-city fleet, live game hosting running paying customer worlds, and the price oracle. An RPC workload and adaptive-bitrate video are built and waiting on deployment gates; the rest sits behind opt-in flags reporting an honest zero until switched on.

Live example: the Price Oracle

One of these has run the longest. Each node independently pulls prices from many venues, discards stale or manipulated quotes with a deviation filter, blends the survivors into one liquidity-weighted number, and signs it. A second pass across the fleet takes a stake-weighted median of those signed reports, so no single broken or dishonest node can move the published price. If that fleet price lands more than **2% away from the broad market**, settlement stops and a person reviews it before anything clears. That gate is temporary, there to prove the machinery behaves while the network is young, and we expect to retire it after about twelve months of running clean. It is the template for how every heavier workload plugs into the network.

REWARDS

Rewards are weighted by a workload's demand and cost and paid for verified work, never for passive holding.

WHAT A NODE EARNS

The node network runs as a real business. Games, studios, and developers pay for the work nodes do, and operators are paid for doing it.

ON WHAT A NODE EARNS

We do not publish an expected return, because we would be making it up. What a node earns depends on demand for network services, on which workloads that particular machine can actually serve, and on what the market decides \$FORK is worth, and all three move independently. A well-provisioned machine in a period of high demand and a base machine in a quiet month are not remotely the same investment. This may become more predictable as the network grows and there is history to reason from. You are always free to run your own numbers, so here are the inputs rather than the answer. In the first year the Emissions Reserve pays node operators **121.2M \$FORK** in total; the model assumes **800 nodes online** ramping toward the 3,278 issued licences; and **we publish no token price at all**. We publish the inputs and decline to publish the answer, because the answer depends on your machine, the workloads you choose to serve and a price we neither set nor forecast. **Nothing about a return is ever guaranteed.**

Settled cleanly, paid in token

Customers pay in ordinary money or stablecoins for storage, content delivery, live hosting, RPC and oracle data, AI inference, and GPU compute, so a buyer never has to hold a token to use the network. Customers who do pay in \$FORK get a 10% discount on node services, which rewards using the token and adds to its demand. Prices are set to what the market will absorb, which sits well under the major cloud providers, and each operator decides whether that return is worth running a given workload. Each service payment splits four ways: **65% buys \$FORK on the open market and routes it to the operators who did the work, 23% stays with the company in fiat to run operations, 10% is bought back into the Emissions Reserve, and 2% goes to the operator-controlled DAO in fiat.** That 10% is the company putting a tenth of every service dollar back into the pool that pays operators, which is the same thing 65% of every license sale already does. Operators are paid entirely in \$FORK, never in cash, and the company never sells the token to fund itself. The economics are strongest on a home machine you already pay to run, where the marginal cost of an extra workload is close to nothing. A rented VPS works too, though its fees come out of what the operator nets. Home hosting is the intended model.

Credits go one way, on purpose

A customer buys credits with ordinary money or with \$FORK, and spends them on services. **Credits never travel in the other direction.** They cannot be converted back into \$FORK, cannot be cashed out, and cannot be transferred between accounts. This is the Steam Wallet model, and it is a wall in the ledger rather than a policy we could quietly relax. It keeps credits a closed-loop utility for buying compute instead of something that behaves like a deposit or an investment, and it is the reason operators are paid in \$FORK bought on the open market with service revenue rather than out of a credit balance. The two systems never touch.

HOW THE TWO CHANNELS PAY

Operator pay arrives through two separate channels that stack on top of one another. A node has to be live to earn from either.

- **Availability rewards** pay for being a live, healthy node and for standing ready to serve heavier work. Every node earns a base reward for uptime, and each additional capability it can prove it is ready to serve adds a multiple on top, so a machine that keeps more capacity ready earns more. These come from a declining draw on the Emissions Reserve and run on an earn-then-claim-then-vest cycle: an operator earns while the node runs, claims when they choose, then waits out a fixed vesting window during which the node must stay online and no new rewards accrue. Section 13 sets out the window and what taking it early costs. Readiness is checked, not taken on trust: nodes challenge one another on an unpredictable schedule, and a capability that fails its challenge stops earning its multiple.
- **Workload rewards** are the 65% of each real service payment, recycled into \$FORK and routed to the operators who did the verified work, paid straight through on top of availability rewards. This channel carries no separate multiple, the payment itself already reflects what the work was worth, and it scales with real paid service, so the network is paid mostly by customers as it matures.

TIER	TYPICAL PAID WORK
T0 Base	Availability, content
T1 Storage	Storage, CDN
T2 Compute	Hosting, RPC, oracle
T3 GPU	AI inference, transcode
T4 Anchor	Wholesale capacity

WE ARE NOT PUBLISHING AN EARNINGS FIGURE

An earlier draft carried indicative dollar ranges per tier. **We have taken them out.** Any such number is a product of assumed utilisation, an assumed workload mix and a token price, and this document publishes none of those, so the figure would have been the only place in it that quietly did.

What an operator earns depends on their hosting cost, the workloads they can serve and how busy the network is. **Those are your numbers rather than ours, and the arithmetic is yours to do.** The rate card is not final and will move with supply and demand in the industry and on the network. Node rewards are always for verified work, never a yield.

WHAT EACH WORKLOAD EARNs

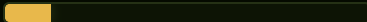
Each capability's multiple is weighted by what it costs an operator to keep ready and how much it is worth to the network. Base availability is the reference point at **1.0×**. Heavier and scarcer capabilities earn a higher multiple for staying ready to serve them, and because the multiples are additive, a well-provisioned machine stacks several at once. The chart shows every workload in the catalog, grouped by power tier.

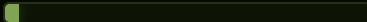
■ BASE 1.0× ■ T1 STORAGE ■ T2 COMPUTE ■ T3 GPU ■ T4 ANCHOR

REFERENCE

Base availability  1.0×

A • CONTENT DELIVERY

Media CDN  5.0×

Mod & UGC hosting  1.5×

Patch & update delivery  2.0×

Asset streaming  2.0×

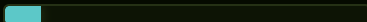
B • PRESERVATION

Decentralized museum  1.5×

Save & mod vault  1.5×

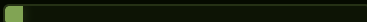
Source dead-man switch  1.5×

Sunset escrow  1.8×

Server revival  4.0×

C • IDENTITY & SOCIAL

Decentralized leaderboards  1.5×

Cross-game identity  2.0×

Reputation graph  2.0×

Voice & chat relays  3.0×

D • WEB3 & ECONOMIC

Asset media pinning  2.0×

Provenance & attestation  2.5×

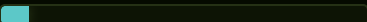
Verifiable randomness  3.5×

Chain infra (RPC)  4.5×

Archive node  7.5×

Oracle & settlement  5.5×

E • LIVE HOSTING

Relay & NAT punch  3.0×

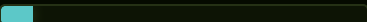
On-demand sessions  4.0×

Rented headless servers  4.0×

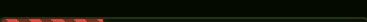
Geo-optimal servers  4.5×

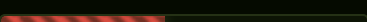
Tournament lobbies  4.5×

F • AI & GPU INFERENCE

Matchmaking & rating  3.5×

Distributed anti-cheat  4.0×

Inference • consumer 24GB  11×

Inference • prosumer 48GB  18×

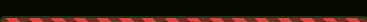
G • STREAMING & HEAVY COMPUTE

Physics & sim offload  7.0×

Spillover compute  7.0×

Render / pixel streaming  40×

H • ANCHOR

Datacenter GPU / anchor  40×+

ESTIMATES, SUBJECT TO CHANGE

These are illustrative estimations of how the reward weighting will work over the long run, anchored to base service at 1.0×. Multiples are additive across every capability a node can prove it is ready to serve, so a well-provisioned machine stacks several. GPU inference is tiered by the card's memory, roughly 11× for a consumer 24GB tier, 18× for a prosumer 48GB tier, and 40× for datacenter-class cards, where the anchor capacity also sits. Readiness is verified by nodes challenging one another, and a capability that fails its challenge stops earning its multiple. The numbers set the shape of the curve, heavier work earns more, and are subject to change as the reward formula is finalized. These figures are not a promise of yield.

WHERE THE HEAVY WORK GOES

The table on the previous page runs from **1.0× for base availability to 40× and beyond for anchor-class work**. That spread is the whole of this section. A node earns the multiple of the work it actually performs, and the 65% passthrough is paid on verified work per node, so **a node doing heavier work already earns more without anything new being built to make it happen**.

Work goes to the nodes that can carry it

Heavy workloads are routed to nodes that can serve them. Render and pixel streaming need a GPU. Archive and oracle work need storage and uptime that ordinary hardware does not hold. Rather than spreading that work thinly across a fleet that mostly cannot do it, the network sends it where it can actually be served, and the operator who serves it is paid for it directly.

Nothing about this is compulsory. We set the specification for each tier and publish it. An operator who meets it is eligible for that work. An operator who does not keeps their license, keeps their node, keeps earning base availability, and can upgrade whenever they choose or never. **No node is switched off, downgraded or bought out for failing a specification introduced after it was purchased.**

WHY THIS RATHER THAN RAISING THE FLOOR

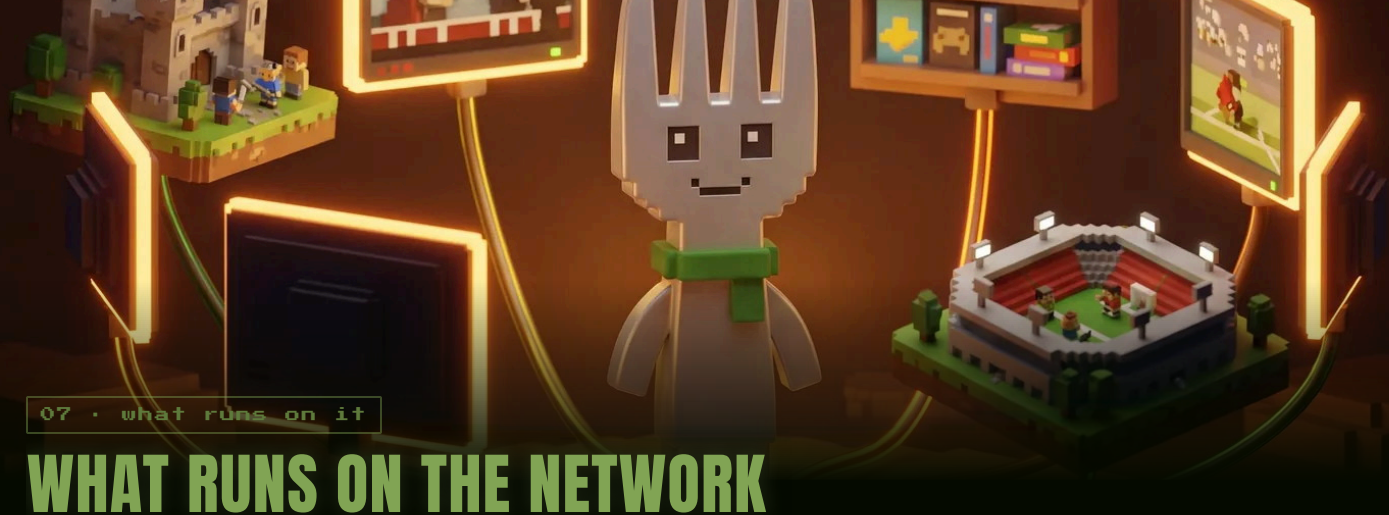
The alternative is to raise the minimum specification for everyone and disqualify whoever cannot meet it. That would be faster and we are not doing it. **It would bill every operator for hardware on our schedule and take the position of anyone who could not pay**, on a standard that did not exist when they bought. Routing reaches the same place by consent instead: the operators who invest do better, and the ones who do not are not expropriated for it.

What we are not going to pretend

Two things follow from this that are worth stating rather than leaving to be discovered.

An operator who never upgrades earns base rates. Base availability is the bottom of a table that reaches 40×, and for some operators base rates will not cover what the machine costs to run. The choice stays theirs and the license stays theirs, but **choosing not to upgrade has a real cost and we are not going to describe it as neutral**.

Reward concentrates toward capable hardware. The heaviest tiers carry a large share of total service revenue, so if only a small part of the fleet can serve them, a small part of the fleet earns a large share. **That is in tension with what a distributed network is for**, and it is a live design question rather than a settled one. **So we will publish the distribution monthly**, showing what share of workload reward the most capable nodes take, alongside the rest of the network's reporting. We are not going to name a threshold at which it becomes a failure, because the heaviest tiers carry most of the revenue by design and a number set today would be arbitrary. **It sits with us, and it gets reviewed every time a capacity vote fires.** We would rather name it here than have it read off the payout data later.



07 • what runs on it

WHAT RUNS ON THE NETWORK

A node network is only worth anything if real things run on it. This section is the layer a player actually touches. None of it asks anyone to understand a blockchain, none of it requires a wallet to start, and every one of these is live today on the fleet described in the previous section.

Hosted game servers

Buy credits, pick a game, and get a dedicated multiplayer world of your own in about a minute. The scheduler places it on a node with capacity, keeps a snapshot so the world survives a host swap, and pauses it when the last player leaves so nobody pays for an empty server. Minetest and Factorio run today, with the world file living in the network's own proof-backed storage rather than on one machine's disk. Community members are running their own worlds on it now, which makes this the plainest proof in the whole document that the network does paid work for real people.

Forked Play

Upload a file and it is encrypted, split into shards, and spread across the fleet, then streamed back to you on demand. No single node holds your file and no node can read it. It runs as a private beta on the web with a native Android client, and it is the consumer face of the same storage and streaming machinery the node section describes. It is also the honest test of that machinery: media playback punishes a storage layer that is slow or lossy in a way that a benchmark never will.

Browser games, uploaded by anyone

A creator uploads a self-contained browser game, it passes moderation, and it is hosted across the fleet and playable from a link with a leaderboard attached. The nodes serving it never need to know what it is, which is the point: the network sells capacity, not curation. It has become the easiest way for someone in the community to put something they made in front of other people.

THE LIBRARY, THE ACCOUNT, AND THE PROOF

A game library that spans the stores

A cross-store discovery portal covering roughly nineteen thousand titles from Steam and Epic on a single catalog spine, with a filter for whether a game touches web3 at all and a per-player switch for adult content that is off by default. It exists because the wall between web2 and web3 gaming is mostly an artifact of where things are listed, and because a player looking for something to play should not have to care which storefront it came from.

One account across all of it

A single profile carries a player across every surface here. Sign in once, link the accounts you want to link, and your credits, your standing, and your history follow you. Linking a wallet is done by signing a challenge rather than by pasting an address, so the link is proven rather than claimed, and a linked wallet is what lets service payments and rewards find the right person. A player who never links a wallet loses none of the products.

The proof anyone can watch

The clearest demonstration is a public page that plays a feature film while showing you which nodes on six continents are serving each fragment of it as it arrives. Kill nodes and the film keeps playing, because the erasure coding means the missing fragments are rebuilt from the ones that survive. It takes a claim that is usually made in a diagram and makes it something a person can sit and watch for ninety minutes.

ALL LIVE

Every product on these two pages is deployed and running in production today. Their limits are stated where they exist: media streaming is a private beta, and the paid paths that would move real money through them wait on the same counsel and audit gates as the rest of the token economy.

DEVELOPER COMPONENTS

Underneath the consumer products sit the pieces a third-party studio builds against. These are the composable layers, not standalone products.

- **Marketplace infrastructure.** The marketplace (Section 11) is built by Forked.gg: a lazy-mint drop contract for settlement, NFT data indexing, wallet connection, and decentralized media storage on the node network, with fiat and compliance handled through a regulated merchant-of-record partner when a paid path opens. The initial integration chain is BNB Smart Chain, and we intend to stay as chain-agnostic as possible: expansion to other EVM chains should follow quickly, while non-EVM chains are a much larger engineering lift and will take more time.
- **Developer marketplace API.** The interface behind Section 11's second stream. A game lists its items here, on-chain or off-chain, from the systems it already runs. The API and the developer console are built and deployed behind a login, with the money and asset rails deliberately stubbed until counsel and the token audit clear.
- **Game Integration SDK.** A modular toolkit for embedding blockchain functionality into existing or new games without rewriting them. In-game trigger points map to on-chain actions; the chain runs in the background while players keep the systems they know.
- **Content Marketplace (UGC).** Players capture, package, and distribute in-game experiences as reusable digital content, valued by contribution quality rather than follower count. The same ownership primitives that back collectibles back creator content, so a clip, a level, or a cosmetic is something its maker actually holds and can trade.

STATUS

The marketplace and the genesis collection are complete with the claim flow proven on test networks. The developer marketplace API is built and login-gated. The Game Integration SDK and the UGC content marketplace are planned, and the marketplace core is built asset-type-agnostic to carry them.

AEGIS GOVERNANCE PLATFORM

Aegis is the mechanism for decentralized coordination and capital allocation, proposed by Jason Brink as an open, permissionless framework and implemented here as shared public tooling. Forked.gg, including its executives, is subject to the same rules as anyone else, and it is open source: anyone can fork it and run their own instance, with no privileged upgrade key held over other deployments.

Aegis stays dormant until the network switches it on. It takes **at least 250 operational nodes voting to enact governance** before any proposal can bind anything, and no mechanism exists for the company to start that clock early. When the network governs itself is entirely the network's call. **Until it does, we set the parameters.**

- **Scope.** Initially the Forked.gg Grant Pool, funded by 10% of daily token emissions, supports developers, infrastructure contributors, and ecosystem initiatives.
- **Node model.** Maximum supply 200,000 nodes, set high on purpose and raisable only by operator vote. Fleet sizing is taken from real numbers rather than a target: the **3,278 legacy licenses** described in Section 2, held across 642 wallets. **Roughly 800 have actually been online**, which is the figure the model starts from, growing toward the issued count as operators bring machines back up and more licenses sell. One node, one vote. Voting power is not weighted by stake or balance, and emissions and votes alike are counted on at most 100 nodes per verified operator, so extra wallets buy neither rewards nor influence.
- **Eligibility.** Nodes must be online, running an approved workload, verified in real time, with at least 80% historical uptime.
- **Process.** The 250-node threshold turns governance on once; it is not the bar for an individual proposal, which opens on a sponsor threshold of **25 nodes**, set at activation and amendable by governance thereafter so an ordinary operator can put one up without assembling a third of the network. Legitimacy comes from the vote: passage is a simple majority (at least 51% approve) with a quorum of at least 25% of active nodes (rolling 30-day average). Voting windows are typically 7 days, recorded on-chain. One vote per node; delegation is allowed and does not bypass uptime requirements.
- **Proposal categories.** Funding proposals (grants and initiatives, capped around \$100,000 equivalent each) and Critical proposals (protocol upgrades and security, higher thresholds). Critical thresholds are left to evolve as the system matures, so early rules cannot lock the project into something that stops working.
- **Security.** NFT-based node identity, real-time validation, immutable records.

DEPLOYED, PROVEN, AND NOT PAY-TO-PLAY

Aegis is not a design on paper. The node registry, the governor and the workload registry are deployed on BNB Chain testnet, and a full cycle has run on-chain there: register, propose, vote, execute, and pay out of the treasury. What remains before it governs anything real is mainnet and the audit that gates it. Voting is one node, one vote, never weighted by token balance, and the costs to submit or signal a proposal are kept low or waivable so ordinary community members are never priced out.

WHY THE DAO LAUNCHPAD

The structure problem

One of the hardest parts of saving a game is structural rather than technical. A studio or a rights holder will almost never sign a game over to an individual, no matter how capable or well-meaning that person is, because there is no one to hold accountable, no entity to sign a contract with, and no protection for anyone involved. So the title gets abandoned, the community that loves it has nowhere to stand, and a good game quietly dies. The DAO Launchpad exists to fill that gap with a structure a rights holder can actually deal with.

Why we care

We have all watched a favorite game rise, build a real community, and then fall, usually for reasons that had nothing to do with whether the game was good. We would rather give those games a path forward than watch the same pattern repeat, and we are genuinely excited to build the scaffolding that makes that possible.

What this is not

We want to be as clear as we can be about this. The DAO Launchpad is not Forked.gg picking up abandoned games and continuing to develop them. We do not make games, and we have no interest in making games. What we provide is the toolset, the legal, technical, and funding structure that lets the developers and the communities who actually love a game keep building it, with everything they need to give it a real chance over the long run. The work, and the credit, stay with them.

THE DAO LAUNCHPAD

A framework for games that have lost active stewardship, so a community can take one over and keep it alive: hold the rights, maintain the code, run the servers, and fund the people who do the work.

The frame

We are gamers first. We use blockchains only where they earn their place: holding a shared treasury no one can quietly walk off with, coordinating maintainers who do not all know each other, keeping a tamper-evident record of who owns and agreed to what, and funding long-horizon work in the open. Anywhere a normal company, contract, or database does the job better, we reach for that instead. If a revival needs no token and no chain at all, that is a perfectly good outcome.

Three machines in one

A revival is a legal vehicle that can hold the IP and sign things, a technical commons that keeps the game buildable and runnable, and a governance layer that funds and steers the work. The order matters and it is the whole point: rights first, a runnable build second, community and money last. Most attempts die because they build the token before anyone has confirmed the right to touch the code.

Reference cases

When Knockout City was sunset in 2024, its studio released self-hosting server software so communities could keep playing, proof that an owner sometimes will cooperate. Clean-room projects like OpenRA and OpenMW keep older games alive by reimplementing the engine and owning none of the original assets, proof of the path that needs no permission at all.

THE THREE LAYERS

Layer 1, the legal vehicle (do this first)

There are three clean paths to a legitimate revival: the owner grants a license or open-sources the game (fastest, when you can get it); you acquire the rights outright (cleanest end state, hardest to close); or a clean-room reimplementation that owns none of the original assets and needs no permission (the most engineering). Nothing technical or financial counts until one of these is true for the title. Once rights are settled, the DAO needs a real legal wrapper, a Wyoming DAO LLC or a foundation, so it can hold IP, sign contracts, hold a treasury, and shield contributors from personal liability.

Layer 2, the technical commons

This is mostly ordinary software preservation. A reproducible build, so a clean machine can compile a working binary years from now. Decoupling from dead infrastructure, replacing auth, matchmaking, and backends with community-runnable equivalents. Repo custodianship, where a small trusted maintainer group holds merge rights while the DAO votes on direction and budget rather than on individual pull requests. And asset durability, where huge game assets live on normal plus durable storage with only manifests and content hashes recorded on-chain.

Layer 3, funding and governance (done safely)

A token can fund the work, coordinate governance, and reward contribution. It must not be sold as an investment tied to the game's revenue, which creates an unregistered security and the wrong incentives. It must not pre-sell governance over an IP no one controls yet. It must not become pay-to-win. The model that fits is a non-financial membership and contribution DAO, governed cred-only or by contribution rather than by a tradeable speculative asset, the same posture locked for FORKcast. For many titles the right answer is no token at all, just a grant pool and a maintainer multisig.

DISCIPLINE

This runs on the same read-only-until-rights-clear discipline as the rest of Forked.gg. A revival is analysis until the rights are confirmed in writing, never a claim of ownership before then.

BUILDING THE LAUNCHPAD

The DAO Launchpad is built in stages, and most of its hardest parts are structure and tooling rather than new technology. The components below are what it takes to carry a revival from a confirmed right all the way to a funded, running, community-governed game. Each one leans on infrastructure Forked.gg is already building, so the Launchpad assembles existing pieces rather than starting from zero.

Phase 1, the legal toolkit and intake

Standardized DAO legal wrappers, a Wyoming DAO LLC and a foundation template, alongside rights-intake and licensing templates and a contributor-agreement set. A rights-first assessment checklist sits in front of all of it, so no work begins on a title before a right is confirmed in writing.

Phase 2, governance and treasury

A cred-only or contribution-based governance module running on Aegis, a grant pool and maintainer-multisig pattern, and contribution tracking so funding reaches the people actually doing the work. This is where a revival gets a transparent treasury and a way to steer it without handing control to a tradeable token.

Phase 3, the technical commons

A reproducible-build pipeline so a clean machine can compile a working binary years from now, drop-in replacements for dead auth, matchmaking, and backend services running on the Forked.gg node network, and an asset-durability layer that keeps large game files on durable storage with manifests and content hashes anchored on-chain.

Phase 4, the self-serve launchpad

An end-to-end flow that carries a confirmed revival from intake through entity formation, repo custody, funding, and hosting, proven out on a first reference revival run start to finish. At that point a community that holds the rights to a game can stand one up without rebuilding the scaffolding every time.

THE FRONT DOOR EXISTS

The intake end of Phase 1 is built and running. A developer who is shutting a game down can hand it off through a live flow on the platform: they describe the title and what they are willing to release, we confirm the rights position in writing before anything else happens, and the game moves onto the node network for hosting. It has been run end to end on production. It sits behind an admin gate today because the funding half of the flow waits on the same token gates as everything else, and because the first handful of real hand-offs should be walked through by a person rather than self-served.

OUR ROLE, AND ITS LIMITS

Forked.gg provides the tools and the framework. We are not the ones who run a revival for a game. The structure described in this section is something a community stands up and operates for itself, using what we build, not a service we perform on anyone's behalf.

We are not lawyers, and nothing here is legal, financial, or tax advice. We cannot and will not form your legal entity, draft your binding agreements, run your governance, or hold or manage your treasury. We have no interest in taking on those roles. Those are decisions a community makes with its own qualified counsel and its own people.

What we can do is give you a roadmap and a clean way to integrate a rebuilt game into the Forked.gg ecosystem: the node network for hosting, the marketplace for assets, and Aegis for cred-based coordination. Everything in this section is offered as a general framework and a set of starting instructions, not as specific advice for any particular game, project, or jurisdiction.

DISCLAIMER

The DAO Launchpad is a set of self-service tools and reference materials. Forked.gg is not a law firm, financial adviser, fiduciary, or service provider to any revival, and provides no legal, regulatory, tax, financial, or investment advice. Nothing in this document creates any advisory, agency, partnership, or professional relationship, or any obligation on Forked.gg to form entities, draft agreements, operate governance, or custody assets for any project. Any community or individual using these tools is solely responsible for confirming the rights to a title and for its own legal structure, compliance, governance, and treasury, and should obtain independent professional advice before acting. Frameworks and examples here are general information only and may not fit any specific situation or jurisdiction.



10 forkcast

FORKCAST

FORKcast, the Forked.gg predictive engine, is a prediction layer for things that actually happen on a livestream. It works on any stream you can link from Twitch, YouTube, or Kick: you put a position on a clearly defined, observable outcome, and if you read it right you gain standing. It is a composable, rules-based module that any game or tool in the ecosystem can build on.

What it is

A market is a single, clearly defined question about an observable outcome, with the resolution criteria written down at the moment the market is created. Examples: "is the streamer's character eliminated within the first 10 minutes," or "how many times is a chosen keyword spoken on stream." One outcome per market. Qualified participants create markets; the protocol sets the rules they have to follow. Streamers are first-class participants and can opt into enhanced automation and visibility on their own channels.

Points

Points are the internal unit you use to take a position. They reflect conviction inside the system. Markets are pari-mutuel pools: you back an outcome, and the pool pays the side that was right, proportional to your position in it.

How v1 launches: cred-only, no cash

The first version is points-only. Every account gets a one-time starting grant of points, and the payoff is standing, a persistent net-worth number and a leaderboard rank, rather than money. There is no way to buy in with cash and no way to cash out in v1. This is a hard architectural wall in the ledger, not a setting. The model is the Hollywood Stock Exchange: you play to be right and to rank, not to withdraw.

Points are not built to last

Cred-era FORKcast points are non-convertible play tokens and nothing else. They cannot be bought with \$FORK, cannot be redeemed for \$FORK, and carry no claim on anything. They exist so we can exercise the engine, the resolution path and the dispute path against real usage while the network is young, and are deliberately not built to be durable. A future version may reset, re-base or retire them entirely. Nobody should treat a points balance as a store of value or expect it to survive into whatever comes next.

Fees, in points

A flat protocol fee is taken from each market in points and redistributed inside the system: **5% to the streamer whose channel hosted it**, and a share to a leaderboard fund. The remainder pays out to the participants who were right, in proportion to their position on the winning side. **Every one of these flows is denominated in points.** No money enters or leaves a FORKcast market, and there is no revenue line attached to it.

WATCH PARTIES & THE TROLLBOX

Any stream, its own page

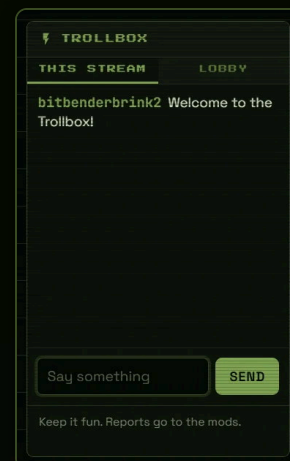
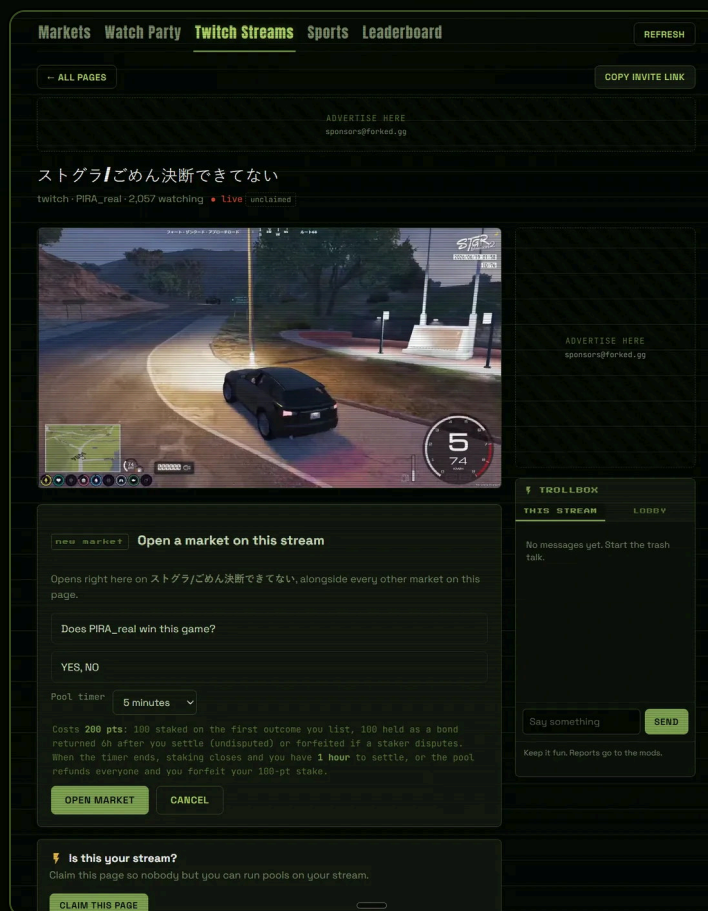
FORKcast sits on top of any livestream. Paste a link from Twitch, YouTube, or Kick and FORKcast gives that stream its own page, with the video embedded and prediction pools running beside it. There is no partnership requirement and nothing to install, so the engine reaches any public stream rather than a fixed list of games or channels. A game, a match, a tournament, a debate, an awards show, anything you can pull up live becomes something a room can predict on.

Watch parties

A watch party turns that page into a room you share. You invite a group of friends with a single link, everyone lands on the same embedded stream, and you open pools and play them against each other in real time. Someone calls a market, the room takes sides, and the leaderboard settles who read it best. It makes watching together competitive without anyone leaving the stream or installing anything.

The trollbox

Every FORKcast page carries a trollbox, an old-school chat in the spirit of the early crypto communities, running outside the streaming platform's own comments. It gives the people in the FORKcast pools their own shared channel, so the banter, the calls, and the trash talk stay among everyone actually playing instead of getting lost in a creator's chat. A site-wide lobby handles general talk, and each stream gets its own room for whatever is on screen.



RESOLUTION & AI

Resolution

In v1, a market is resolved by the participant who created it, which is not necessarily the streamer. The creator has a set window to resolve, backed by a full audit log and a dispute path. If they do not resolve in time, the market is automatically voided: every position is refunded and the creator forfeits the points they staked to open it. Once a market is resolved, the pool pays out and everyone moves on. Manual resolution with accountability is more honest than claiming an automated oracle we have not proven, and the roadmap adds AI-assisted and API-based resolution (game telemetry, video, audio transcription, time-stamped stream metadata) where it can be made reliable and auditable, starting with outcomes that have a clean machine-readable source.

AI-assisted market creation (live)

A market is only as good as the questions people put up, so we shipped an assistant that helps create them. On a streamer's page, a "Suggest Pools" action reads the current stream context and returns several ready-to-run market drafts, each with a proposed question, a set of outcomes, and a suggested timer. The streamer accepts, edits, or ignores them, and nothing goes live without that human step. The assistant only suggests. It never opens a market, never resolves one, and never pulls private player data from third-party game APIs, which keeps the system inside the same streamer-controlled, opt-in posture as the rest of FORKcast. It is built on Minds by Animoca Brands and labeled as such in the product. The same human-gated pattern is the model we intend to extend to resolution assistance over time.

For developers

The engine is exposed as a composable primitive other games and tools can build on. Aggregated, privacy-preserving prediction data can become a useful signal layer over time. We anchor market state on-chain by publishing hashes, not individual bet histories, so the system stays auditable without exposing participants. Market and epoch state is anchored hourly as Merkle roots on BNB Chain testnet today, with offline proofs anyone can check.

LIVE

The engine is public at forkcast.forked.gg: cred-only markets with the one-time points grant, a public front page, Discord as the primary login, private watch-party rooms with their own chat, the trollbox and lobby, multi-platform stream pages, Suggest Pools, and automated result adapters that read finished matches from public game APIs (Dota 2 and PUBG) and settle from what actually happened. A Resolution Provider SDK for outside data sources is built and not yet mounted.



11 • marketplace

THE FORKED.GG MARKETPLACE

The Forked.gg Marketplace removes the artificial wall between web2 and web3 gaming and treats blockchain as an optional capability layer rather than a prerequisite. The long-term aim is one player experience that spans web2-native and web3-native games, where ownership is real and the chain stays in the background for anyone who does not want to think about it.

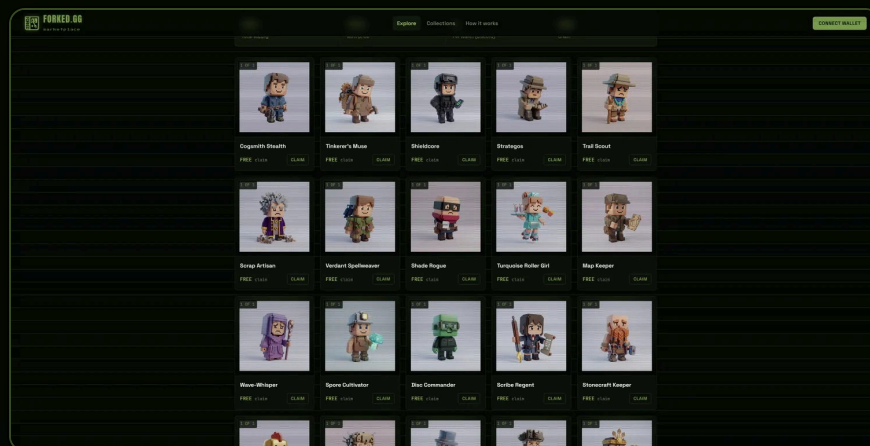
Two streams

The marketplace is built in two stages on a deliberately asset-type-agnostic core, so the work we ship now also carries the work that comes later.

- **Stream 1, community collectibles (building now).** A branded storefront for Forked.gg's own collections, beginning with the genesis drop on the next page.
- **Stream 2, a developer marketplace (built, gated).** A toolkit so any game can list and sell its items here, on-chain or off-chain, from the systems it already runs. It shares plumbing with the Game Integration SDK in Section 7. Web2 studios can enable asset sharing, trading, and secondary markets without migrating their core systems on-chain, so adoption is incremental and integration risk stays low. The API and the developer console are built and deployed behind a login, with the money and asset rails held until counsel and the token audit clear.

How it is built

The marketplace is ours. It runs on BNB Smart Chain with a lazy-mint drop contract, so a collection can be listed without minting the whole supply up front and a claimant pays only network fees. The genesis collection is deployed and the claim flow is proven end to end on test networks. The front end, the NFT data indexing, the wallet support, and the claim path are built by Forked.gg, with decentralized media storage on the node network. Fiat and compliance, when a paid path opens, are handled through a regulated merchant-of-record partner rather than built in-house.



THE FORKED.GG MARKETPLACE • THE FORKED 444 GENESIS COLLECTION, FREE TO CLAIM

THE GENESIS DROP: FORKED 444

The first collection is **Forked 444**, forkhundred and forkty fork, a set of 444 one-of-one collectible characters called Forkies. They are chunky retro-voxel figures in the Forked.gg house style, each a distinct named character with no traits and no rarity tiers. A selection is shown below.



Flametongue #086



Chromalux #081



Lightbringer #015



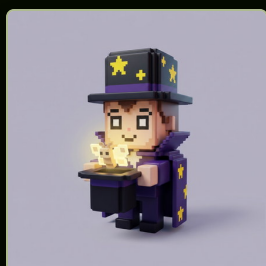
Spark Wielder #178



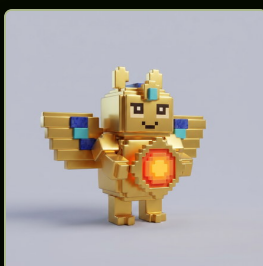
Crystal Sovereign #190



Mystic Oracle #308



Starlight Conjuror #362



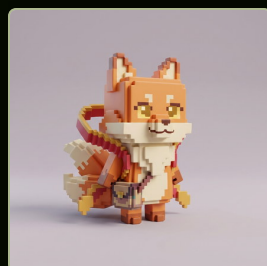
Solar Jewel #381



Crimson Virtuoso #206



Sky Ace #110



Kitsune Courier #341



Aeon Guard #388

Aurealis, the Radiant
Jester #395Symphonia, the Winged Idol
#405

Void Admiral #419

A free mint for the community

Forked 444 is a free claim, one per wallet, open to active community members with a linked wallet, first come first served. There is no sale, no presale, and no cost beyond network fees. It is an origin badge for the people who showed up early.

No utility, on purpose

The Forkies have no planned utility of any kind. They are not tied to the token, the nodes, governance, or any reward, and they carry no promise of value. They are here for one reason: they are cool, and they give the community something real to own before any money is involved.

STATUS

The storefront and the full genesis collection are complete, the drop contract is deployed on BNB Chain testnet with all 444 listed, and the free one-per-wallet claim is proven end to end there. Counsel review on payments and AML applies before any public or mainnet launch.

ROADMAP

Ordered by gate rather than by date. Most of what is left is not waiting on engineering, it is waiting on a specific external thing: an independent audit, a counsel answer, or a deliberate decision to open a door. Saying which gate holds each item is more useful than a quarter that slips.

Done, and running in production

- **Nodes.** The fleet runs the metering backbone in production. Storage with proof of possession and paid gigabyte-months, media streaming across a twenty-city network over the open internet, live game hosting carrying paying customer worlds, and the price oracle, all live. Node uptime is committed on-chain as Merkle roots and independently reproducible.
- **FORKcast.** Public and cred-only. Watch parties, the trollbox and lobby, multi-platform stream pages, Suggest Pools, timers with auto-void and the dispute path, automated result adapters, and Discord sign-in.
- **Products.** Hosted game servers, media streaming in private beta with a native Android client, uploaded browser games, the cross-store game library, one account across all of it, and the public six-continent streaming demonstration.
- **Marketplace.** The storefront and the full genesis collection, with the free one-per-wallet claim proven end to end on test networks.
- **Economy, on a public test network.** \$FORK and the treasury, reserve, vesting, recycler, and migration contracts all deployed and verified on BNB Chain testnet with their buckets funded. A public faucet. Market state anchored hourly. Operators paid real test-network tokens for real measured work, both directions of the loop, proven end to end.
- **Governance.** Aegis deployed on the same test network with a full register, propose, vote, execute, and treasury-pay cycle run on-chain.
- **DAO Launchpad.** The developer hand-off intake, live behind an admin gate, run end to end once.

Built, waiting on a named gate

- **Waiting on the audit.** Mainnet deployment of every contract above. The node license contract. The paid rails inside the developer marketplace.
- **Waiting on counsel.** Anything that moves real money through user-uploaded content, and the adaptive-bitrate video path that sits behind the same content question.
- **Waiting on deployment, not on code.** The peer-challenge availability proof, the RPC endpoint workload, the operator payout dashboard, and the remaining catalog workloads, which are committed behind opt-in flags and report an honest zero until they are switched on.

ROADMAP

Next, once the audit clears

The independent audit is the single gate holding the largest block of finished work. Everything in this group follows it directly and in this order.

- **Token.** \$FORK to mainnet with initial liquidity. The LIX conversion claim opens on its six-month vest. Service revenue begins recycling into open-market buys.
- **Nodes.** The license model live and the first licenses sold, which carry a second gate on top of the audit: none is offered or sold until the corporate entity is formed. Emission live on the earn-then-vest cycle. The workload registry under Aegis control. Paid workloads billed in credits with the discount for paying in \$FORK.
- **Governance.** Aegis on mainnet, one node one vote, the grant pool funded, and the first proposals that decide something real.
- **Marketplace.** The genesis collection to mainnet, and secondary trading.

After that

- **Nodes.** The GPU tier opens: AI inference, transcoding, then render and pixel streaming. Behind-the-meter anchor capacity onboards across regions. Staking and slashing go network-wide across every bonded capability.
- **Developers.** The Game Integration SDK, the developer marketplace rails opened, and the first third-party games integrating through them.
- **Players.** The UGC content marketplace, and more streaming platforms behind FORKcast.
- **FORKcast cash on-ramp.** Only if counsel clears it, only where it clears it, and with identity verification and sanctions screening on every participant. This one may never open, and the product is designed so that it does not need to.
- **DAO Launchpad.** A first community revival carried the whole way, from confirmed rights to a running game.

The long run

The network runs primarily on real service revenue, with the base reward for uptime still underneath it, refilled by the activity it supports. Governance and treasury decisions move further toward operators and the community, and the DAO Launchpad takes on more titles. The measure of whether any of this worked is not the token price. It is whether games that would have died are still being played.

THE \$FORK TOKEN MODEL

\$FORK gives the ecosystem a real working currency while deliberately avoiding the pattern that sank the first wave of web3 games, where tokens were emitted faster than anyone could use them, farmers dumped them, and the price and the community collapsed together.

Design principles

- **Fixed maximum supply.** A hard cap. No function can mint \$FORK beyond it. All emissions are scheduled releases from a pre-allocated reserve, never new inflation on top of the cap.
- **No team allocation, equity only.** The team holds zero \$FORK. Its only upside is equity in the operating company, so no insider is ever structurally a seller of the token.
- **Operators paid in token, never cash; the company never sells to fund itself.** Node operators are compensated entirely in \$FORK for verified work. The company buys \$FORK with service revenue and never sells it to fund itself. The one deliberate exception, market making, is set out in full later in this section.
- **Emission that follows demand.** Issuance is a fixed small share of whatever remains in a capped reserve, so the daily draw falls as the reserve falls. The reserve is refilled by real activity. In a network nobody uses it drains and the base reward tapers with it; in a network being used it holds. No operator has to plan around a date when the floor disappears.
- **Rewards for real work, never passive holding.** There is no "stake your tokens and the protocol prints you more" yield. Operators earn for verified service; contributors earn through governance-approved grants.
- **Real, usage-tied sinks.** Demand mechanisms are tied to actual activity, so as usage grows, net new supply trends toward neutral.

WHERE THIS STANDS

This section is a design under review, and the numbers in it are still subject to counsel. The machinery, though, is no longer hypothetical. \$FORK and the treasury, reserve, vesting, recycler, and migration contracts are all deployed on **BNB Chain testnet** with their buckets funded, a public faucet dispenses test tokens, market state is anchored on-chain hourly, and operators have been paid real test-network tokens for real measured work with the full loop closed end to end. **Addresses are not published in this draft**, because the test deployment will be superseded and pointing you at contracts that are about to be replaced invites you to check the wrong thing. **The full contract list is published before the token generation event**, with the audit, and it is the mainnet set rather than this one. Test-network tokens have no value and are not for sale. Mainnet waits on the independent audit.

HOW \$FORK DIFFERS FROM \$LIX

\$LIX was this network's earlier token. \$FORK is a new token on a smaller, cleaner supply, and the holders who earned it are honored in full. **The token generation event is gated on the corporate entity being formed**, which is with counsel now and is expected to complete before the date below. If formation runs late, the event moves with it. The date is **Wednesday 16 September 2026**, and every \$LIX figure here is measured or projected to that date. The table sets the two side by side: verified on-chain facts for \$LIX, and the current proposal for \$FORK. *(Proposal for review.)*

DIMENSION	\$LIX · LEGACY	\$FORK · NEW
Status	Snapshot taken. No longer supported; do not buy, sell, or transfer.	New, separate token, deploying on BNB Chain. Snapshot balances convert at the fixed ratio.
Maximum supply	50,000,000,000 (50B)	1,000,000,000 (1B) , fixed, no minting beyond the cap
Supply minted	~19.3B projected to the 16 Sep 2026 TGE (17.81B verified at the 23 Jun snapshot); the rest still issuable	Released only on a published schedule from pre-allocated buckets
Conversion	Not applicable	50 LIX : 1 FORK (1 LIX = 0.02 FORK), proportion-preserving
Team / insider allocation	~51.6% of minted supply sat in company and insider multisigs	0% to the team. Equity only.
Holder structure	368 holders, ~84% held by contracts and multisigs	Each honored holder keeps the exact same percentage of supply
Base emission rate	~0.056% / day of the remaining latent pool, measured on-chain. The published design said 0.15%; the chain does not agree with it	0.088% / day gross of the remaining reserve, declining; 90% to node operators, 10% to their DAO
Demand backing	None. No liquidity pool, \$0.00 on-chain market.	75% of node-service revenue becomes buy pressure for \$FORK: 65% to operators, 10% back to the reserve
Unsold token sale	~4.76B LIX unsold seed-round pool sat minted	Not carried forward into \$FORK
Chain	BNB Smart Chain (BEP-20)	BNB Chain first, chain-flexible by design

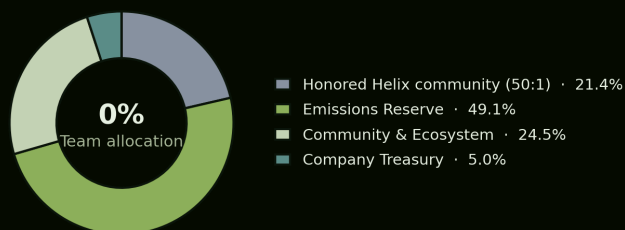
PROPORTION-PRESERVING BY DESIGN

The 50:1 step is a clean redenomination of the supply. Every honored holder keeps the exact same percentage of total supply they held in \$LIX, with no dependence on any price. The \$LIX figures here are reproducible from BscScan; the \$FORK column is a working design subject to legal review before any external use.

SUPPLY & ALLOCATION

Fixed maximum supply: **1,000,000,000 \$FORK**, with no mint beyond the cap. Supply splits into two layers: the commitments we chose to honor to the network's first operators, redenominated at a clean 50:1 and taken off the top, and a new economy designed underneath. Every figure below is pinned to the **token generation event of 16 September 2026**. *(Proposal for review.)*

ALLOCATION	%	TOKENS	VESTING / RELEASE
Honored operator community (50:1)	21.4%	~213,700,000	6-month linear vest, emergency-claim with forfeit. The advisor and SAFT lines vest on their own contractual schedules; see the Layer A breakdown
Emissions Reserve (node rewards + DAO Grant Pool)	49.1%	~491,300,000	Declining draw, refilled by real network activity
Community & Ecosystem	24.5%	~245,000,000	Programmatic, contribution-based; multi-year
Company Treasury	5.0%	~50,000,000	Market making, liquidity, and exchange listings only
Team	0%	0	No allocation. Equity only.



No team allocation, and a public treasury

The team holds zero \$FORK. Its only upside is equity in the operating company, so no team member is ever structurally a seller of the token. The company treasury is **5% and is earmarked for market making, liquidity, and exchange listings**, nothing else. Company operations, salaries, audits and business development are funded out of the company's 23% fiat share of service revenue, not out of tokens. The company is owned by the team through equity, so we state plainly that there is an indirect line from treasury value to the team rather than imply the team touches nothing. Every token the company holds sits in a multi-signature wallet whose address is published, so anyone can watch it on-chain and see exactly when it is used and for what, and no single person can move those tokens alone.

THE ALLOCATION MATH, SHOWN

None of the slices above are picked by feel. The honored operator community (Layer A) comes off the top first, sized by real on-chain balances at the TGE date. The company treasury is then fixed at a deliberate 5%, and **everything left over goes to the Emissions Reserve**, because the reserve is what pays node operators and the operators are the point of the network. Here is that arithmetic in full.

```
# Layer A is measured, not chosen: on-chain $LIX at TGE, divided by 50
```

```
Layer A = 213.7M $FORK → 21.4%
```

```
Layer B = 1,000M - 213.7M = 786.3M $FORK
```

```
# treasury and community are fixed; emissions takes the remainder
```

```
Company Treasury = fixed = 50.0M → 5.0%
```

```
Community & Ecosystem = fixed = 245.0M → 24.5%
```

```
Emissions Reserve = the remainder = 491.3M → 49.1%
```

```
Team = zero = 0 → 0.0%
```

HONORED OUTSIDE THE TOKEN

Separately from the 213.7M above, the **3,278 node licences already issued convert one for one** into Forked node licences. They are not part of the one-billion supply and take nothing from it, but they are a real transfer and belong in the same place as the rest: at the current \$1,700 floor they represent about **\$5.6M** of licence value, and more than that if the network fills and the price rises with it.

Inside Layer A, line by line

The ~214M honored layer is itself fully itemized. Every line but one is a real on-chain \$LIX balance divided by 50, projected to the **16 September 2026** token generation event. The node-reward line is **capped at 151.2M \$FORK**, struck on end-September accrual rather than the event date so there is headroom if it slips. The pool is still accruing until then at roughly **18M \$LIX a day**, declining with the latent supply, which is why it is larger than the June snapshot. That rate is measured, not assumed: the pool held 5.869B on 23 June 2026 and the projection to TGE applies the observed daily accrual decaying at 0.15% a day. Re-read on 16 August 2026 the same contract held 6.817B against a predicted 6.823B, a difference of 0.09%.

LAYER A LINE	\$LIX	\$FORK	% OF 1B
Node-reward contract redemption	7.337B	146.7M	14.67%
Community redemption (362 EOAs)	2.806B	56.1M	5.61%
Early advisors (6 × 0.1%, contractual) [†]	0.300B	6.0M	0.60%
Early token investors (SAFT, honored) [‡]	n/a	4.86M	0.49%

[†] Six advisors contracted at 0.1% of supply each, held together in a single wallet of **300,000,000 \$LIX**, converting at 50:1. They are, by role rather than by name until their agreements with the new entity are executed: two game industry executives, two web3 industry executives, one platform lead and one capital-markets adviser. They vest over **36 months, monthly, with no cliff**, on a schedule of their own rather than the six-month redemption vest. **Patrick Bergman and Jason Brink are advisors to Forked.gg and receive no token allocation of any kind.** [‡] Two early token purchasers. **Their earlier agreements lapsed unexercised, the \$LIX behind them never having been distributed, so nothing converts on this line.** They hold new standalone Forked.gg agreements for the \$FORK shown, vesting 10% at the event and the rest linearly over twelve months. The node-reward line is projected to the 16 September 2026 event and **capped at 151.2M \$FORK** on end-September accrual, so the cap carries headroom.

WHAT ACTUALLY CIRCULATES

The allocation table says where the supply is assigned. It does not say when any of it can move, and those are different questions. This is the release schedule for the first two years, generated from the same model as every other figure in this section. Months one to six are shown individually because that is where most of the movement is; after that, quarterly.

MONTH	HONORED LAYER	EMISSION	COMMUNITY	CUMULATIVE	% OF 1B
1	34.8M	12.9M	5.1M	52.9M	5.3%
2	34.3M	12.6M	5.1M	104.9M	10.5%
3	34.3M	12.3M	5.1M	156.6M	15.7%
4	34.3M	11.9M	5.1M	208.0M	20.8%
5	34.3M	11.6M	5.1M	259.1M	25.9%
6	34.3M	11.3M	5.1M	309.8M	31.0%
9	0.5M	10.5M	5.1M	359.0M	35.9%
12	0.5M	9.7M	5.1M	405.7M	40.6%
15	0.2M	8.9M	5.1M	449.0M	44.9%
18	0.2M	8.3M	5.1M	490.3M	49.0%
21	0.2M	7.6M	5.1M	529.6M	53.0%
24	0.2M	7.1M	5.1M	567.3M	56.7%

The **50M company treasury is deliberately excluded**. It is market-making inventory that has to be placed as liquidity to do its job at all, so counting it as distributed supply would overstate what is actually loose in the market, and counting it as locked would understate it. It is disclosed separately and the wallet address is published before the token generation event.

READ THIS ONE HONESTLY TOO

Roughly **31% of the cap is circulating by month six** and **40.6% by the end of year one**. The reason is the honored operator layer: 202.9M of it clears in six months, with the advisor and SAFT lines running longer, because those are commitments made to people who have already waited a year and we would rather pay them than smooth our own chart. That front-loading is the single largest driver of the negative net flow on the next pages, and it is the thing a stretched vest would change. It is a real cost of honoring the community first, and we are choosing to pay it rather than hide it.

HOW THE LEGACY REWARDS WERE RELEASED

The node-operator reward track from the network's first year is fully accounted for on-chain. Every reward was earned by verified uptime and recorded in a public registry, so a holder can trace exactly what was earned, what has been paid, and what is still owed.

The chain of record

Each step is reproducible from public data, the on-chain uptime registry and the LIX token contract, with no need to take our word for any of it:

- **On-chain uptime registry.** Node uptime is committed as daily and hourly Merkle roots. We independently rebuilt all 2,201 roots from the underlying data, and every one matched with zero mismatches.
- **Daily points.** A node that stays up earns one point for that day, taken straight from those verified roots.
- **LIX earned.** Each day's operator emission, set by the published schedule, is divided across that day's points to give the LIX earned per node.
- **On-chain payouts.** LIX paid to operators is visible transfer by transfer from the reward contract.
- **Owed.** What is earned but not yet paid stays minted in the reward pool, waiting to be claimed.

NODE-OPERATOR REWARD TRACK (PROJECTED TO TGE, 16 SEP 2026)	LIX
Earned by operators (7.81B verified on-chain to 23 Jun, accruing to TGE)	~9.28B
Paid out to operators (through April 2026)	1.94B
Owed, already minted and held in the reward pool	~7.34B

ALREADY MINTED, NOT NEW SUPPLY

That ~7.34B owed LIX converts at 50 LIX to 1 FORK into the ~146.7M FORK node-reward line in Layer A, projected to the 16 September 2026 TGE from the pool balance verified on 23 June 2026. It is not new issuance: the LIX was minted as it was earned and sits in the node-reward pool today, so converting it is a distribution of tokens that already exist. There is no separate or additional reward allocation, and nothing is owed off-chain.

PROVING UPTIME ON-CHAIN

Every claim on the previous page is checkable by anyone, because node uptime is committed to a public smart contract rather than asserted in a spreadsheet. The registry is the **GlobalNodeUptimeMerkleRegistry** on BNB Smart Chain, with verified source, at `0x4123933f0250b2966008878dccc00663444be746`.

How a Merkle root proves uptime

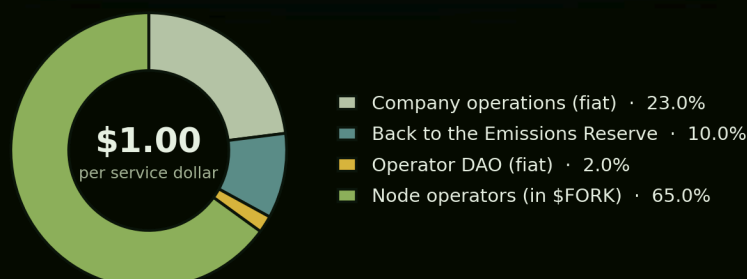
For each period, every node's record becomes one leaf: a keccak256 hash of the node's network slug, its on-chain address, and its measured uptime in milliseconds, where a perfect day is 86,400,000. All of a period's leaves are combined pairwise up a Merkle tree, and the tree's single root hash is written on-chain alongside an IPFS link to the full underlying list. That root is a fingerprint of the entire set. Change any one node's uptime by a millisecond and its leaf changes, the root changes, and the on-chain commitment no longer matches the data.

What that lets anyone do

Anyone can pull the IPFS list, recompute every leaf, rebuild the tree, and check the root against the chain. A single operator can also prove their own record is included with a short Merkle proof, just the handful of sibling hashes along the path to the root, without needing the whole set. We ran the full check across every committed period, daily and hourly, and all **2,201 roots reproduced exactly from the public data, with zero mismatches**, covering roughly 2.72 million verified node-hours. The reward math on the previous page is built on top of that verified record, so the whole chain from uptime to tokens owed is reproducible by an outsider end to end.

HOW OPERATORS ARE PAID

Customers buy node services, RPC, hosting, CDN, storage, AI inference, oracle settlement, render, and pay in fiat or stablecoin through ordinary rails. No one has to touch \$FORK to buy a service. Each service dollar then splits four ways.



Each service payment splits four ways. **65%** is owed to the people who ran the work, and it reaches them as \$FORK. The company keeps **23%** in fiat to run itself. **10% buys \$FORK on the open market and goes back into the Emissions Reserve**, the pool the base reward is paid from, so the company funds the operator floor out of its own revenue rather than only out of a fixed starting balance. **2% goes to the operator-controlled DAO in fiat**, so the people running the network have an operating budget that does not depend on the token. Operators are paid entirely in token, never in cash, and the company never sells the token to fund itself. The buyer touches no token, the operator touches no cash.

The \$FORK buffer, and why we sometimes buy less

Not every customer pays in dollars. Some settle in \$FORK directly, and anyone buying network credits can pay for them in \$FORK at a discount. Every token that arrives that way lands in a single **buffer pool** rather than being sold.

That pool is then the first place we look when operators need paying. If a fiat-paying customer generates an obligation to hand operators \$FORK, we draw the buffer down to meet it and only go to the open market for whatever the buffer cannot cover. **A large buffer never means the company sells; it means the company buys less.** Tokens that came in from one customer go straight back out to the operators serving another, and the open-market purchase shrinks by exactly that much. It is the same pool on both sides, so \$FORK spent on credits and \$FORK spent on services are interchangeable once they land in it.

How the conversion price is set

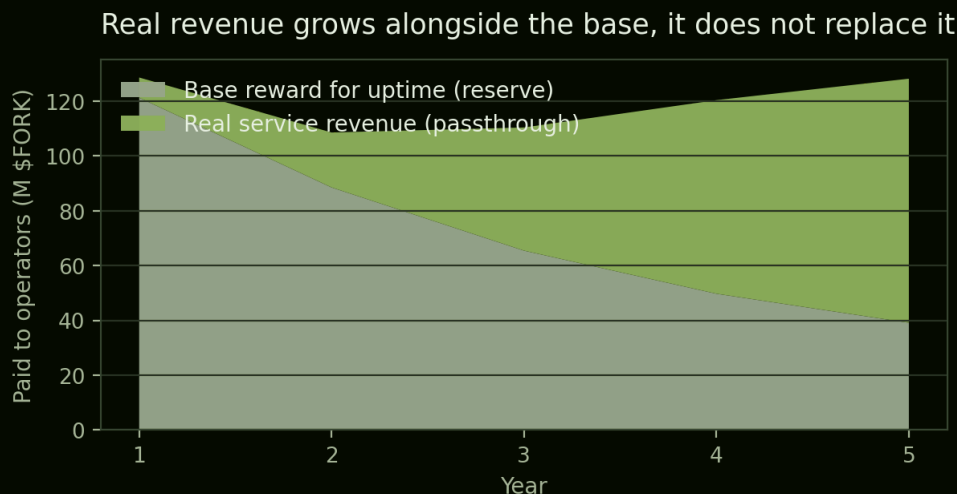
The price used to turn each fiat payment into \$FORK is produced by the Oracle network running on the nodes themselves, which independently pulls from many trading venues, discards outliers, and takes a stake-weighted median across the fleet. That node price is then cross-checked against two further oracles, CoinGecko and CoinMarketCap, which are independent of the node network and are used to validate its output rather than to compute it. If the fleet price diverges from those checks by more than **2%**, the system stops and requires manual review before any settlement clears, rather than quietly failing over. This gate is a temporary safeguard for the network's first year and is expected to be retired once the oracle has a long enough clean record to stand on its own. A primary feed plus two independent validators means there is no single point of failure in how the token is priced.

TWO REWARD CHANNELS

Throughout this document **“operators”** means the node operators together with the DAO they control. They are one payee class, not two, and the DAO is not a company pool: it is governed one node, one vote through Aegis by the same people running the machines.

Operator pay flows through two channels that work together. The first is a base reward for being a live, healthy node, drawn from the Emissions Reserve on a declining schedule. **That emission channel splits 90% to the node operators themselves and 10% to their DAO**, so a tenth of every emitted token stays with the operator community to spend on itself through governance. The second channel is the workload reward: the recycled service revenue, paid in full to the operators who performed the verified work and weighted by the demand and cost of that work. The 90/10 applies to emissions only; service revenue passes through whole to the operators who earned it. A node has to be live to earn the base, and live to earn workload pay on top.

The base reward’s job changes over time, and it does not end. Early on it carries most of operator pay, while the network is young and real revenue is thin. As service revenue grows the passthrough grows alongside it and becomes the larger share, so what falls away is an operator’s dependence on the reserve rather than the base reward itself. The base stays because standing ready has a cost even in a quiet month, and a machine that earns only when a job happens to land is a machine that gets switched off.



EMISSION & THE CURVE

Daily emission is a declining draw from the Emissions Reserve, split **90% to node operators and 10% to the DAO Grant Pool** governed through Aegis. The company takes none of it. **Operator pay has two independent sources and emission is only one of them.** The baseline reward for uptime is fixed to the declining curve on the next page and does not move with revenue. The passthrough on real customer spend is paid on top of it and runs on its own track. Emission is never topped up or throttled to hit a target, and it is the baseline channel alone that carries the vesting friction. Nothing is ever minted beyond the cap.

Emission is denominated in \$FORK and never in dollars. We will not promise an operator a fixed dollar amount per day, because doing that would quietly make us responsible for the token's price, and we are not. One \$FORK is worth one \$FORK; what that is worth in any other currency is signalled by a market, not by us. The gross draw is a small fraction of the remaining reserve each day, **0.088%**, of which 90% reaches node operators, so the reserve releases on a smooth declining curve and never hits a cliff. Recycled service revenue is paid on top of that base, so operator income grows with real usage instead of depending on how fast the reserve is drawn down.

This rate is lower than the one the earlier track ran, and the difference does not come out of what node operators earn. It comes from taking the company out of the token. The company draws nothing from emission and is funded only by its portion of real service revenue, so every token released goes to node operators and the community grant pool, with none of it going to the company. What was removed is the company's claim on the token, not the operators' reward.

The reserve is not a closed tank, and this is the part worth being precise about. Four flows put tokens back into it: **65% of every license sale, 10% of every service dollar**, stake slashed from an operator who broke something, and any unvested remainder forfeited by an early claim. So the reserve drains when the network is quiet and holds when it is busy. We think that is right in both directions. A subsidy that keeps paying for uptime nobody wants is a token hose. A subsidy that quits on operators while the network is working is a broken promise. The draw never reaches zero in any case, because a fixed percentage of a remaining balance only approaches it. More useful than that arithmetic is where the reserve settles once the flows above are running. Where it settles depends on how fast service demand grows, because that is what governs both how many licences release and what they sell for. The next page traces it. There is no last day of emission. **Every figure in this section is stated in the unit it is actually denominated in:** token amounts in tokens, dollar amounts in dollars. We do not convert between them anywhere in this document, because that conversion needs a token price and we do not have one to give you.

THE EMISSION CURVE, IN NUMBERS

A fixed rate on a shrinking base falls every day, with no cliff and no governance vote needed to wind it down. The table below is the **closed-tank floor**: no license proceeds recycled, no buyback, no inflow of any kind. The four flows on the previous page all lift it above this line, by an amount that depends on what the token trades at. **The floor is what we print, because the floor is what we can state without inventing a price.**

$$E(t) = r \times L(t)$$

$$L(t) = L_0 \times (1 - r)^t$$

$r = 0.088\%$ / day gross draw (90% nodes, 10% their DAO); $L_0 = 491\text{M}$ \$FORK starting reserve (t in days)

YEAR END	RESERVE LEFT	DRAWN THAT YEAR
Start	491M	.
Year 1	357M	135M
Year 2	259M	98M
Year 3	188M	71M
Year 4	137M	52M
Year 5	99M	37M

Where it settles, and why we stop there

The table above is the floor and it is stated with **no inflow at all**: no license proceeds recycled, no company buyback, nothing coming back. It is the closed-tank case, and it is the only version of this curve that does not need a token price to draw.

What actually happens is better than that, because **75 cents of every service dollar buys \$FORK on the open market** and goes back into the reserve. How much supply those dollars retire depends on what the token trades at, which is not ours to say. **A lower price retires more of it; a higher price retires less.** We are not going to draw that curve, because every version of it we could draw would be a price forecast wearing a chart's clothing.

WHAT WE REMOVED, AND WHY

Earlier drafts of this document converted every token figure at a flat reference price struck from the early-purchase basis. **We have taken it out.** It was a single invented constant that every token number in this section rested on, held flat across five years, and it produced precise-looking figures that were really just that one assumption restated.

Taking it out costs something. **You can no longer calculate a payback period from this document, and neither can we.** That is the honest position: payback on a token-denominated income stream is not knowable in advance, and a paper that implies otherwise is telling you something it does not know.

DEMAND AGAINST THE SELL SIDE

What reaches holders, and what buys it

Two quantities matter here and they are denominated in different things. We are going to give you both, and we are not going to combine them.

Reaching holders over five years, in tokens. Emission runs to **392M** on the closed-tank floor, of which **353M** goes to operators and 39M to the operator DAO. The honored operator layer releases **214M**, of which **202.9M** clears **inside six months**; the advisor and SAFT lines carry the remaining 10.9M on their own longer schedules, running to months thirty-six and twelve. The community pool releases **245M** over four years. That is **851M in total, about 85% of the 1,000M supply**, reaching holders inside five years.

Buying it, in dollars. 75 cents of every service dollar buys \$FORK on the open market: 65 cents passed through to the operators who did the work, and 10 cents the company buys back into the reserve. On top of that, 65% of any license proceeds do the same. On the fleet modelled in Section 6, over five years, that is roughly **\$9.4M** of committed buying, and more if licenses sell.

IF OPERATORS SELL THIS SHARE OF THEIR EMISSION	TOKENS SOLD OVER 5 YEARS
30%	375M
50% (base case)	445M
70%	516M

Includes the honored layer at 80% sell-through and the community pool at 40%, which together account for 269M of every row and are the larger part of the sell side at any operator rate.

THE PASSTHROUGH IS NOT IN THAT TABLE, AND HERE IS WHY

Operators receive the 65 cents as well, and sell some of it. It is left out because **it cannot be stated in tokens without a price**, and it is **largely a wash**: we buy those tokens on the open market with service dollars, hand them to the operators who did the work, and they sell a fraction back. The buying and the selling are the same tokens. **To reconcile the two sides yourself, that wash is the step to get right.** Convert the dollar buying at whatever price you think right, add the operators' share of it to the sell side at your chosen rate, and net the two. Skip that step and you understate the sell side, in our favour.

READ THIS HONESTLY

Hundreds of millions of tokens reach holders in the first five years and a large share of them will be sold.

There is no arrangement of prices at which that is a small number. Against it sits a buy commitment fixed in dollars rather than in tokens, which means the amount of supply it retires rises as the price falls and shrinks as the price rises.

We are not going to turn those two into one figure. Doing so needs a token price, we would have to invent it, and the invented number would drive the answer entirely. The structure is what we can tell you honestly: supply fixed in tokens, buying fixed in dollars, and a market in between that we do not control. **A model, not a promise of value or yield.**

CLAIMING REWARDS AND THE 30-DAY VEST

Base rewards are not minted the moment they are earned. While your node runs and stays healthy, its base rewards build up as an accumulated, unclaimed balance. Nothing reaches circulation until you decide to claim, which puts the timing in the operator's hands instead of dripping a constant stream onto the market.

The 30-day vest

When you claim, the accumulated balance begins a 30-day vest. Leave it for the full thirty days and the entire amount unlocks and is yours. You can take it sooner, but taking it early means receiving only the portion that has vested so far and surrendering the rest, which returns to the Emissions Reserve to be earned again by someone still working. Forfeited balances never reach the company, and the same rule covers early legacy claims and slashed stake. The closer to day 30 you wait, the more of the balance you keep, and waiting the full window keeps all of it.

Earning pauses while you vest

While a claim is vesting, the node keeps running but stops accumulating new base rewards. **That share does not vanish. It goes to the operators who are still accruing**, because each day's emission is divided across that day's active nodes. So the cost of claiming is **relative rather than absolute**, and it is worth being precise about which. If every operator claims on the same cadence, the vest costs nothing in tokens: everyone pauses for the same share of the year and everyone's slice grows while they are active. It bites when you claim faster than the people you are sharing the pool with. **An operator claiming monthly against a field that claims annually earns roughly half what the patient operators earn.** Every day spent vesting is a day not earning, and someone else is earning it.

Why the friction is there

This is deliberate. It is what stops the daily claim-and-sell cycle that drained most reward tokens straight onto the market in the projects that came before. Each operator picks their own cadence and weighs earning against access, which paces how fast rewards reach the market and keeps reliable nodes online through the wait. Workload rewards paid from real service revenue run on their own track and are not held by this cool-down, so paid service work keeps moving.

VESTING AND CLAIMS, IN THE OPEN

Legacy redemptions vest on a separate rule from operator rewards. Every redeemed balance, the node-reward pool and community wallets alike, unlocks linearly over the same six months. Anyone can emergency-claim early: they receive whatever has vested to that point and forfeit the entire unvested remainder, which recycles to the Emissions Reserve. The earmark stays at full par, so no one is shortchanged on value, only on the speed of a panic exit.

```
unlocked(d) = B × ( d ÷ 180 )
# B = redeemed balance, d = days since launch, capped at day 180
```

No insider view to exploit

A community member known as Free, in the Forked.gg Discord, asked a sharp question: when people claim and vest tokens, can someone with early sight of those numbers trade ahead of everyone else? The answer is that there is no private feed to trade on. Every claim and every vesting balance settles on-chain, so the amounts, the schedules, and the day-by-day unlocks are public to everyone at the same instant. No one at Forked.gg sees this data earlier or in more detail than the rest of the market.

A public read-only endpoint

To make that legible without digging through a block explorer, a public, key-free endpoint will return the same figures straight from the migration and vesting contracts. Exchanges, market makers, and any community member will be able to verify the numbers independently or build their own dashboards on top. It ships alongside the on-chain migration and vesting contracts, and the shape it will return is below. It is not live yet, and the address is published when it is.

```
# planned: public, read-only, no key, no account. Not live yet.
GET /api/v1/vesting/status

{
  "as_of": "2026-10-01T00:00:00Z",
  "unclaimed_fork": 46200000,
  "fully_unlocked_fork": 21750000,
  "vesting_emergency_claimable_fork": 88940000
} # sample response, not live data
```

THE MULTISIG THAT HOLDS EVERYTHING

Someone holds the keys. Pretending otherwise is how projects get robbed, so here is the arrangement stated plainly.

Every privileged key in this system sits behind a multi-signature wallet, and no single person can move anything. That covers the company treasury, the Emissions Reserve, the administrative functions on the conversion and vesting contracts, and any upgrade authority that exists on the token contracts. There is no personal wallet anywhere in that list.

What it can and cannot do

- **It cannot mint.** The one-billion cap is enforced in the contract itself, not by policy. No combination of signatures creates a token beyond it.
- **It cannot take operator rewards.** Earned balances settle to the operator who earned them; the multisig does not sit between an operator and their claim.
- **It can move treasury tokens,** which is the point of it, and every such movement is visible on-chain the moment it happens.
- **It cannot pause the network.** There is no emergency stop, no freeze switch, and no contract in the set exposes a pause function to the signers. Settlement either works or it is broken in the open, and we would rather explain a bug than hold a lever that could be used to halt operators mid-earning.

Addresses

The signer set, the threshold, and every wallet address are **published before the token generation event**, so anyone can watch them from the first day the token exists rather than trusting a description written months earlier. They are pending below because the entity that will hold them is still being formed with counsel, and we would rather print nothing than print an address that changes.

```
# published before TGE, 16 September 2026
Company treasury TBA
Emissions Reserve TBA
Conversion + vesting TBA
Signers / threshold TBA
```

HOLD US TO THIS

If those addresses are not published before the token generation event, that is a broken commitment and should be treated as one. It costs us nothing to publish them and it is the cheapest proof that the arrangement described here is real.

\$FORK UTILITY

\$FORK is the fuel, access key, and integrity bond of the ecosystem, never a passive-yield instrument. Every use below is consumptive, access-granting, or bonded. There is no “stake and the protocol prints you more” mechanic anywhere.

- **Infrastructure settlement.** Service revenue is recycled into \$FORK and routed to operators for verified work. This is the fundamental demand anchor that rests the token on a real service business. Customers who pay in \$FORK get a 10% discount on node services.
- **Marketplace fees.** Marketplace fees can be paid in \$FORK at a discount, and a portion of what is paid cycles back to the Emissions Reserve. **The discount rate and that portion are not yet set**, and neither are the premium SDK tiers below. We would rather mark them undecided than print a number we would have to move. FORKcast is deliberately absent from this list: it runs on points only and no \$FORK flows through it.
- **SDK & Launchpad access.** Premium SDK tiers and DAO activation are priced in \$FORK, and what is spent cycles back to the Emissions Reserve rather than to the company. Node licenses are deliberately not on this list: they are priced in dollars and paid in stablecoins, with the split set out in Section 6.
- **Integrity bonds.** The stakes that activate oracle, settlement and the other critical workloads are slashable \$FORK, set out in full in Section 6. They hold an operator accountable for work whose failure damages other people, and they lock tokens away for as long as the capability runs.
- **Lock-for-access.** Locking \$FORK unlocks higher limits and reduced fees. It buys access and discounts only, never minted rewards.
- **Buying credits.** \$FORK spent on network credits cannot be converted back into \$FORK, cashed out, or moved between accounts. Those tokens are not destroyed, though: they land in the buffer pool and go back out to operators as payment for work. It is a one-way door for the customer and a supply of working capital for the network, rather than a sink. Section 6 sets out why that wall exists and why it is a property of the ledger rather than a policy.

ON THE WORD BURN

We avoid it. In the last cycle a burn usually meant destroying tokens to move a price, and it earned the suspicion it now gets. Nothing here is destroyed. Tokens spent on services, credits, fees and licenses leave circulation and return to the Emissions Reserve, where operators earn them again by doing work. Supply in circulation rises and falls with real usage, the one-billion cap never moves, and none of it is a statement about what the token is worth.

ON MARKET MAKING, AND WHAT “NEVER SELLS” MEANS HERE

We say the company never sells \$FORK, and we mean something precise by it. **No \$FORK is ever sold to fund the company.** Operations, salaries and audits are paid out of the fiat share of service revenue, and every token bought with that revenue goes to operators or into the Emissions Reserve. There is one deliberate exception and it is worth naming rather than burying. The 5% treasury exists to make a market, and a liquidity position transacts in both directions by design: it buys when the price falls and sells when it rises. That is the job of the position, not a way for us to take money out. Every position is published, the treasury sits in a multi-signature wallet whose address is public, and no token leaves it to pay a salary or a bonus.

GO-TO-MARKET & ADOPTION

The business is selling infrastructure. The plan reaches four groups in the order that makes each next one cheaper: players first, because the consumer products are the cheapest thing to grow and they double as live proof that the network does real work; then the streamers who carry those players; then the operators who supply the capacity; then the games and studios who pay for it.

The wedge: FORKcast, with no front door to guard

FORKcast is the cheapest way in because it asks almost nothing. It works on any public stream, there is no partnership to sign and nothing to install, and v1 is cred-only, so the regulatory gate that slows most token products is simply not at the front door. A player lands on a stream page, plays for standing on the leaderboard, and stays. Watch parties are the growth loop: one link puts a group of friends on the same stream calling markets against each other, and the people they beat are the next people who sign up.

Streamers as the distribution channel

Every market pays the streamer 5% of its fee, so a creator who turns FORKcast on takes a growing share of that points pool while their audience plays, and the AI Suggest-Pools tool means they barely have to lift a finger to do it. That turns streamers into the distribution channel rather than an advertising line item. We start with mid-tier creators and the audiences the team already reaches, where a single channel can seed a whole room of new players.

The node network as a two-sided market

Supply is the easy half: there is an existing base of operators who already ran nodes for this network, plus home-hardware operators for whom an extra workload on a machine they already run costs almost nothing. The work is bringing demand to meet them. Operators only ever run the workloads they choose, so the network grows by landing real customers who need content delivery, hosting, oracle data, and compute, and letting operators opt into serving them.

GO-TO-MARKET & ADOPTION

Selling infrastructure that pays for itself

Node services are priced in ordinary money and set below the major clouds, so a buyer never has to hold a token to use the network. The first customers turned out to be closer to home than the plan assumed. They are community members buying credits to run their own game worlds, which is a smaller sale than a studio contract and a much faster one, and it produced paying usage before any enterprise conversation started. From there the ladder runs outward: the ecosystem's own products, then indie and web3 studios who need cheaper CDN, hosting, and RPC, then the games revived through the DAO Launchpad. The motion is land-and-expand, from one workload to several as an operator base proves it can deliver.

Community and ownership

The Forked 444 free mint is an origin badge that costs nothing and is gated to active community members, which gives people a real reason to show up and pulls Discord growth along with it. Participation programs reward genuine contribution rather than noise. And the first operator community, honored at the 50:1 ratio, is a group of people who already believe in this cause rather than one we have to buy.

Partnerships for reach and credibility

FORKcast already runs a live integration with Animoca Brands' Minds platform for AI market creation, which is both a product feature and a credibility signal. Further platform and brand partnerships are pursued the same way: each one has to bring real reach or real demand, not just a logo on a slide.

Phasing, tied to the roadmap

Grow users first on the cred-only beta, where there is no regulatory gate. Reward and widen the community with the genesis mint. Turn on service revenue with node licenses and paid workloads. Open a cash on-ramp only after counsel clears it. Bring third-party games in through the SDK and the developer marketplace, and make the DAO Launchpad revivals the flagship demand that shows the whole system working end to end.

What we hold ourselves to

Delivery is not demand, so we measure demand directly: weekly active FORKcast players and markets, paying node-service customers and monthly service revenue, and operator count and uptime. Service revenue is the number the token economy actually rests on, so it is the one we report against, rather than downloads or signups that look good and prove little.

RISK FACTORS

Honest disclosure of the main risks of participating in or building on Forked.gg.

- **Regulatory risk.** Digital assets, token sales, node licenses, and prediction markets are regulated differently across jurisdictions, and the rules are changing. Features described here, especially any cash on-ramp or cash-out in FORKcast, may be restricted, delayed, geofenced, or changed to comply. Nothing here is a commitment to offer any feature in any specific place. When a cash on-ramp launches, it is intended to be available in major markets subject to legal review, with identity verification on every participant and exclusion of anyone subject to OFAC or equivalent sanctions. The final jurisdiction list follows that review, and we do not name target markets before it is complete.
- **Node license payback.** A node license is a cost, not an investment, and nothing guarantees it ever pays for itself. A license earns nothing on its own: the operator has to run the machine, do work for paying customers, and have that work proved before any payment occurs. If demand for network services does not materialize, if the hardware cannot serve the workloads that pay well, or if the token price falls, a license may return less than it cost, take far longer than expected to break even, or never break even at all. Do not buy one with money you need back.
- **Node license classification.** Licenses are sold as software licenses and confer no tokens, yield, revenue share, or economic rights of any kind. A regulator in a given jurisdiction could nonetheless take a different view and treat them as securities or another regulated instrument, particularly if a resale market develops. That could force us to change terms, restrict or refund sales, geofence markets, or stop offering licenses somewhere entirely, and it could happen after purchases have been made. The analysis is not settled and goes to counsel as part of forming the issuing entity.
- **No promise of value.** \$FORK and node licenses may lose value or become worthless. Recycling, emissions, and other mechanics manage supply; they are not promises about price.
- **LIX conversion risk.** Honoring the first operator community depends on correctly identifying genuine holders and converting them at the stated ratio. Disputes, errors, or unclaimed balances are possible, and the six-month vest manages timing but does not guarantee price stability.

RISK FACTORS

- **Unaudited contracts.** No independent third-party audit of the \$FORK contracts has been completed at the date of this draft. The contracts are deployed on a test network only. No mainnet deployment happens before an audit, and an audit reduces but never eliminates smart-contract risk.
- **Technology and smart-contract risk.** Bugs, exploits, and failures can cause loss. Audits reduce but do not eliminate this.
- **User-uploaded content.** Several live products accept files from users: uploaded browser games, uploaded media, and community forums. Material that infringes copyright, breaks a platform rule, or is otherwise unlawful can be uploaded despite moderation. We operate a takedown process, and the network is designed so that operators serving a file cannot read it, which is good for privacy and means a node operator is not in a position to police content themselves. The legal treatment of that arrangement differs by jurisdiction and is under review.
- **Third-party game licensing.** The hosted game-server product runs binaries published by other studios. Whether a given title's license permits hosting it commercially on behalf of a customer varies title by title, and a change of terms or an objection from a publisher could require us to stop offering a specific game.
- **Key-person and execution risk.** The project depends on its team and may not deliver the roadmap on time or at all.
- **Third-party and integration risk.** Some described capabilities depend on third-party platforms and partnerships that may change.
- **Intellectual-property risk.** Any game revival depends on rights that must be confirmed in writing first. Until they are, a title is read-only evidence and not a claim of ownership.
- **Liquidity and market risk.** Assets may be illiquid or volatile.



16 • conclusion

CONCLUSION

Forked.gg is a deliberate attempt to raise the floor for how decentralized games are built, governed, and kept alive.

Infrastructure ownership is distributed from the start through capped, on-chain node licenses. Economic design is treated as ongoing work, with predictable emissions and supply discipline rather than improvisation. Ownership sits with players and operators, governance runs on-chain, and coordination stays open and auditable. We intend to earn trust the slow way, by shipping things that work and saying plainly what is built, what is planned, and what is still uncertain.

What we are attempting has not been done in quite this form before, and we know that being first carries its own risk. Yet the history of technology is largely a history of people choosing alignment over extraction before it was obviously safe to do so. The open protocols underneath the internet spread because no one owned a toll booth on them. Linux outlasted far better funded rivals because the people who built it shared what they made instead of renting it back to one another. Again and again, the systems that endure are the ones where the people building them, running them, and using them are pointed the same way.

Forked.gg is a wager that the same holds for a token economy. The individual pieces are ordinary. Funding a company from real revenue is ordinary. Paying operators for real work is ordinary. Holding a token that tracks a product people actually use is ordinary. What is rare, and genuinely untried at this scale, is doing all of it with no team allocation at all, so the people who build the system prosper only when the people who run it and hold it prosper too. We think that is worth the risk. The communities who showed up early for the last wave of web3 games deserve a model built to outlast a single cycle, and we would rather be the ones who tried to build it than watch the same story reach the same ending again.

FORKED.GG

Decentralized infrastructure for entertainment that deserves to keep living.