

# 200,000,000 API3

*Ending emissions, retiring the Aragon DAO, and  
moving governance to Snapshot.*

# In one page

190,071,000

TOTAL SUPPLY AT  
POSTING

9,929,000

FINAL MINT, TO THE  
TREASURY

200,000,000

MAXIMUM SUPPLY,  
FOREVER

0

API3 MINTED AFTER  
EXECUTION

## What changes

API3 is minted via proposal one last time to bring the total supply to exactly 200,000,000. After that, minting capability is removed and ownership of the token contract goes to the dead address (0x...dEaD), so no more API3 can ever be created. Staking rewards end. Governance moves to Snapshot, where each API3 token is one vote, whether it is still staked in the deprecated Aragon DAO, or simply held in your wallet. The DAO's 3-of-5 multisigs execute what passes under the terms set forth below and is not barred by the API3 Foundation's bylaws. What the old DAO's contracts still hold moves to the Foundation-controlled multisigs. The primary and secondary voting apps of the Aragon DAO are set to the highest passing requirement the contract accepts, a hair under 100%, effectively closing them down forever, while leaving the staking pool open for DAO members to unstake and withdraw.

## Why

Since 2021 the pool has minted about 90.1M API3 in staking rewards, 47% of today's supply, and it still mints about 950,000 every Thursday. Most of it gets withdrawn and sold. CoinGecko lists API3's maximum supply as infinite, and under the current setup a vote by stakers could mint any amount through five-year-old contracts built on Aragon. This proposal stops the minting and removes the ability to vote it back in.

## The business

We'll keep working in DeFi and use our experience with APIs to build for AI. Two examples today are curation and AirnodeHub. Curation runs lending markets on our own price feeds, returns the OEV (the value recovered when a price update moves a market) to those markets, and earns a fee when borrowers use them. AirnodeHub lets agents find, call and pay for APIs whose responses are signed by the provider, and will take a cut once payments are live. Revenue buys API3 on the market every month.

## What a holder does

Staked API3 keeps its vote, including delegation, earned rewards keep vesting, and from the first Snapshot proposal everyone votes with what they hold. Once this proposal passes we recommend unstaking and withdrawing as your tokens unlock.

## The numbers

This week's reward was minted on Thursday 29 October, and this proposal was posted after it. Total supply is now 190.07M (rounded; section 3.1 has the figures to the token), so the final mint is 9.93M, which goes to the Treasury multisig. If the vote executes before next Thursday's mint on 5 November, the supply lands on exactly 200,000,000. If it executes later, the DAO burns the excess from its multisigs right after execution, so the supply still ends up at exactly 200,000,000. On Snapshot the quorum is 10,000,000 API3 in favour, the posting threshold is 500,000 API3, and votes run for seven days and are executed by the multisigs within seven more.

# Contents

---

Sections 1 and 2 cover how things work now and what's wrong with that. Sections 3 to 6 are the proposal itself: the change, the rules the DAO runs on afterwards, the multisigs' mandate and charter, and the supply after execution. Sections 7 to 9 cover the business and the treasury, section 10 sums up, and sections 11 and 12 cover risks and the timeline.

|           |                                                   |           |
|-----------|---------------------------------------------------|-----------|
| <b>1</b>  | <b>Background</b>                                 | <b>4</b>  |
| 1.1       | The token and the DAO                             | 4         |
| 1.2       | Staking rewards                                   | 4         |
| <b>2</b>  | <b>Problem</b>                                    | <b>5</b>  |
| 2.1       | Emissions                                         | 5         |
| 2.2       | Governance risk                                   | 5         |
| <b>3</b>  | <b>Proposal</b>                                   | <b>8</b>  |
| 3.1       | Supply                                            | 8         |
| 3.2       | The final mint                                    | 8         |
| 3.3       | Governance                                        | 9         |
| 3.4       | Retiring the current DAO                          | 9         |
| 3.5       | What happens to current stakers                   | 10        |
| <b>4</b>  | <b>How the DAO operates</b>                       | <b>12</b> |
| 4.1       | Voting power                                      | 12        |
| 4.2       | Proposals                                         | 12        |
| 4.3       | Quorum and passing                                | 13        |
| 4.4       | Execution                                         | 14        |
| 4.5       | The rules at a glance                             | 16        |
| <b>5</b>  | <b>The mandate and the charter</b>                | <b>18</b> |
| 5.1       | The mandate                                       | 18        |
| 5.2       | What the multisigs will not execute               | 18        |
| 5.3       | Changing the mandate and the signers              | 20        |
| 5.4       | Attack surface before and after                   | 20        |
| 5.5       | Snapshot as a trusted party                       | 21        |
| <b>6</b>  | <b>Token economics after execution</b>            | <b>22</b> |
| <b>7</b>  | <b>Curation</b>                                   | <b>23</b> |
| 7.1       | What curation is                                  | 23        |
| 7.2       | Our markets today                                 | 23        |
| <b>8</b>  | <b>AirnodeHub</b>                                 | <b>25</b> |
| 8.1       | Agents as customers                               | 25        |
| 8.2       | The provider problem                              | 25        |
| 8.3       | What AirnodeHub does                              | 25        |
| 8.4       | What it means for API3                            | 26        |
| <b>9</b>  | <b>Treasury and revenue</b>                       | <b>27</b> |
| <b>10</b> | <b>Summary</b>                                    | <b>28</b> |
| <b>11</b> | <b>Risks and open items</b>                       | <b>29</b> |
| <b>12</b> | <b>Timeline</b>                                   | <b>30</b> |
| <b>A</b>  | <b>What the proposal submits on chain</b>         | <b>31</b> |
| <b>B</b>  | <b>Proposal parameters</b>                        | <b>33</b> |
| <b>C</b>  | <b>The vesting bypass, verified in the source</b> | <b>34</b> |
|           | <b>References</b>                                 | <b>35</b> |

# Background

---

## 1.1

### The token and the DAO

The API3 token is governed by a DAO that runs on contracts deployed in 2021. To vote, a holder stakes API3 in the [staking pool](#) and votes on the [DAO dashboard](#). Every position, vote and reward can be followed on the [DAO tracker](#). Voting power is staked tokens only. Tokens held in a wallet do not vote.

There are two kinds of proposal, described in the [DAO documentation](#). A secondary proposal needs 15% of all voting power in favour and is used for routine changes. A primary proposal needs more than 50% of all voting power in favour and controls the core contracts, including the token itself. Either executes the moment more than half of all voting power is in favour, and a secondary that clears its 15% executes when its seven days end. There is no delay after that and no veto.

## 1.2

### Staking rewards

Stakers are paid in newly minted API3. The [token contract](#) keeps a list of addresses allowed to mint, and the staking pool is on it. Every week the pool mints rewards at an APR that the contract adjusts by one percentage point a week, up when the staked share of supply is below a target and down when it is above ([mechanism](#)). Rewards vest over one year. The current state is on the [DAO tracker](#).

The rewards were meant to get more people involved in governance. At today's level of inflation we think they do more harm than good.

## Problem

### 2.1

#### Emissions

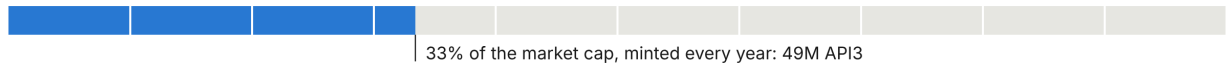
API3 launched with 100M tokens. The pool has minted about 90.1M more as rewards since then, 47% of everything that exists today. The APR was at its 75% cap for most of this year and is 68% after this week's mint, seven weeks into the wind-down. At that rate the pool mints about 950,000 API3 a week, roughly 50M a year or 26% of supply. At \$0.26 that's about \$12.7M of new tokens a year, against a circulating market cap of about \$38M (Figure 1).

A YEAR OF MINTING AT THE CURRENT RATE

\$12.7M

API3 MARKET CAP

\$38.1M



About 950k API3 a week at a 68% APR, at \$0.26. Market cap on circulating supply, CoinGecko, 29 October 2026. Each segment of the bar is 10% of the market cap.

Figure 1. What the pool mints in a year against the market cap

Stakers withdrew 73% of last year's rewards from the pool. On 14 August 2026, with the APR at its maximum, API3 hit an all-time low of \$0.1837 ([CoinGecko](#)). CoinGecko lists API3's maximum supply as infinite.

In theory, rewards only move supply share from holders who don't stake to holders who do. In practice they haven't worked out for stakers either: most rewards are sold, and the dilution hurts everyone who holds the token.

The wind-down started with [secondary proposal #132](#), which cut the staking target to 30%. It was posted on 4 September 2026, passed on 11 September with 33% of all voting power in favour and was executed on 12 September. This proposal ends emissions completely.

### 2.2

#### Governance risk

In a governance attack, someone gets enough voting power to pass a proposal that pays them and executes it before anyone can react. In the current DAO the token contract, its list of minters and the permission system are all controlled by votes, so a hostile primary proposal could make itself a minter and print as much API3 as it liked. That takes half of whatever is staked, about 36M API3 today, and less as stake leaves the pool. Most of the treasury already sits in the [multisigs](#), where a vote can't reach it, so the real exposure is the token contract.

Other DAOs have been attacked or taken over in similar ways (Table 1).

| CASE         | WHEN     | COST TO ATTACKER          | AT STAKE                        | HOW                                                                         |
|--------------|----------|---------------------------|---------------------------------|-----------------------------------------------------------------------------|
| BonkDAO      | Jul 2026 | \$4.4M of BONK            | \$20M treasury, drained         | 1% quorum met by 7 wallets, instant execution                               |
| Compound     | Jul 2024 | ~\$16M of COMP            | \$24M proposal, passed 52 to 48 | Weekend vote, fresh delegate wallets, settled under threat of team multisig |
| Aragon       | 2023     | Open market ANT           | \$200M treasury                 | Treasury above market cap, activists forced \$155M payout and dissolution   |
| Tornado Cash | May 2023 | Hidden code in a proposal | 483k TORN, token down 48%       | Malicious code behind a copied description                                  |
| Beanstalk    | Apr 2022 | One flash loan            | \$182M, drained                 | Borrowed 78% of votes for one block                                         |
| Arbitrum     | Apr 2025 | 5 ETH                     | 19.3M ARB of votes              | Rented on LobbyFi for about 1% of token cost                                |

Table 1. Governance attacks and takeovers, 2022 to 2026. References 16 to 22.

[BonkDAO](#) lost \$20M in July after an attacker spent \$4.4M on tokens, met a 1% quorum with seven wallets while 18,000 holders did not vote, and executed instantly. [Compound](#) saw a \$24M proposal pass 52 to 48 in 2024 after the attacker timed the vote over a weekend, and only a [team multisig](#) forced a settlement. [Aragon](#) was pushed into dissolving and paying out \$155M once activists noticed its treasury was worth more than its token. Votes on Arbitrum can be [rented](#) for about 1% of what the tokens would cost to buy. In August [Binance](#) froze deposits to stop a \$1.2M raid on a DAO.

The API3 DAO has never been attacked, but it's exposed in the same way, and more so whenever stake leaves the pool. We think the market already discounts a token that a single vote could inflate without limit, and holders pay that discount whether or not an attack ever happens.

The other risk is the code itself. Our data feed contracts are small, we wrote them ourselves and we know them well. The DAO is a different story. It runs on Aragon, a large external framework, with an API3 voting app, staking pool and deployment template [built on top of it](#) in 2021. That code was audited four times, most recently in 2022, and hasn't changed since. That used to be an acceptable risk, because finding a bug in old contracts took a specialist weeks of work. It isn't anymore. In December 2025 [Anthropic](#) reported that AI agents exploited 56% of a set of contracts deployed after their training data ended, at an average cost of \$1.22 a run, and that their exploit revenue had doubled roughly every 1.3 months over the previous year. In April 2026 a [model](#) found thousands of zero-days in mature software, including a 27-year-old bug in OpenBSD and a 16-year-old one in FFmpeg. Old custom contracts with an

unlimited right to mint behind them are an obvious target for tools like these, and audits can't be repeated often enough to keep up.

In August 2026 we received a bug bounty report on the staking pool. A staker who schedules an unstake in the same block the weekly reward is minted (and the unstake itself can trigger that mint) gets that week's reward without the one-year vesting every other reward has. We confirmed it in the source and paid the bounty, and Appendix C goes through it line by line. It hasn't been patched, and it's live today. The bug itself is minor. No treasury funds were at risk, and we checked every unstake since February 2024: thirteen landed in a reward block by coincidence, releasing under 1,000 API3 early in total, and none looks deliberate. Once this proposal passes the pool stops minting, so there is no reward left to release early and nothing to fix. What bothers us is that it went unnoticed for five years through four audits, until someone with the right tools went looking, and that it sits in the same contract that holds the only right to mint API3. Given the developments in AI, it is reckless to leave the ability to create API3 behind any contract at all.

So instead of trying to keep the old contracts safe indefinitely, this proposal removes what's worth attacking. After execution the token has no minters and an owner nobody controls, the treasury is in the multisigs, and no vote in the old contracts can pass. A bug in the old code could still affect tokens that stakers leave in the pool, which is another reason we recommend withdrawing.

# Proposal

## 3.1

### Supply

The DAO mints one last time, directly to the Treasury, to bring the total supply to exactly 200,000,000 API3. This week's reward was minted on Thursday 29 October and this vote was posted after it, so the amount is fixed: total supply at posting was 190,071,000 API3, and the final mint is the difference to 200,000,000, 9,929,000 API3. Both figures are rounded to the nearest token here; the vote carries the exact amount in wei. If the vote executes before the next weekly mint on Thursday 5 November, the supply lands on exactly 200,000,000. If it executes later, the pool will have minted again in the meantime, and right after execution the DAO burns the excess from its multisigs, so the supply still ends up at exactly 200,000,000. The execution transaction removes the pool from the minter list before the final mint and every other minter after it, and transfers ownership of the token contract to the dead address, `0x0000000000000000000000000000000000000000000000000000000000000000dEaD`, which nobody has a key for. The token contract doesn't allow ownership to be renounced, so this is how we get the same result. After that the minter list can never be changed again (Figure 2). Appendix A lists every step.

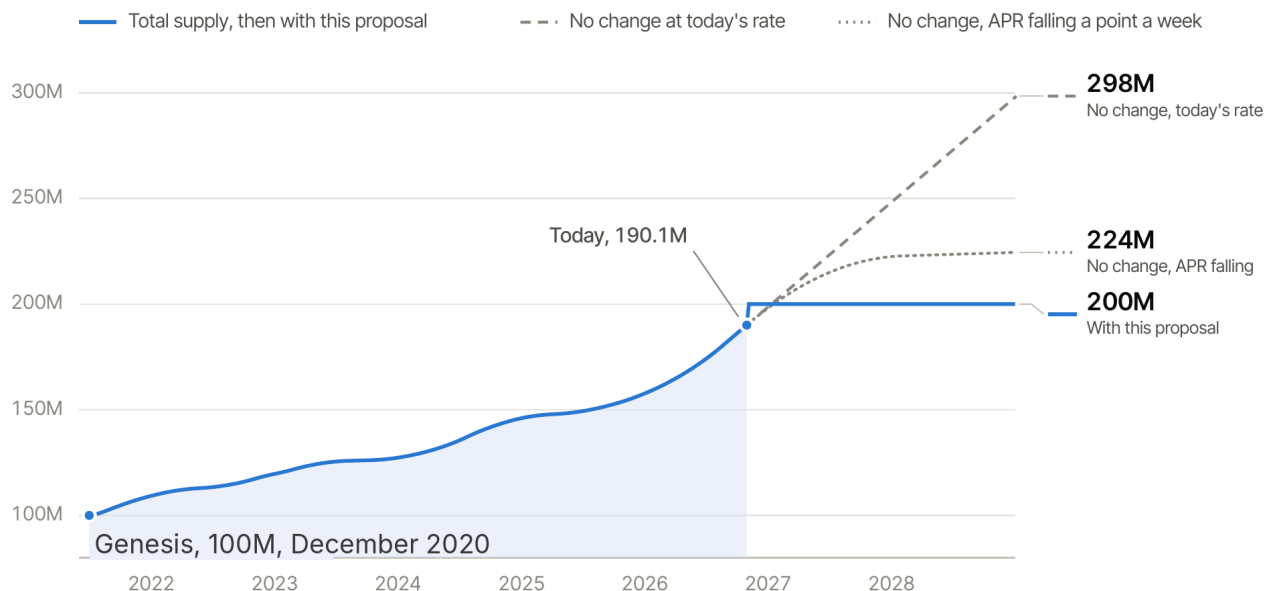


**Figure 2.** Total supply of API3, 2021 to 2028

Genesis supply of 100M, then 90.1M minted as staking rewards, from the pool's reward records. Both projections hold 73M staked; the falling path lowers the APR one point a week from 68%.

## 3.2

### The final mint

The final mint goes to the [Treasury](#), the new multisig that holds the DAO's API3. It's meant for incentivizing and growing the Foundation-stewarded business,

and the API3 costs we already have, such as contributor vesting. Nothing is allocated in advance and there's no release schedule. Every use of it needs a proposal, like any other treasury asset. It dilutes holders by about 5%, once, and after that there's no way to mint again.

### 3.3

## Governance

Votes move to [Snapshot](#). One API3 is one vote, staked or not, and the DAO's 3-of-5 multisigs execute what passes. Section 4 describes how it works.

This vote needs more than 50% of all voting power in favour to pass. It was posted from the Vault, which has enough voting power to post because of the stake delegated to it. Everything the vote will execute is public at [github.com/api3dao/api3-dao-migration-proposal](https://github.com/api3dao/api3-dao-migration-proposal): the source that builds it, a check of the posted vote against that source, fork tests, and a decoder that prints any vote's bytes call by call. The tool the signers use to check Snapshot votes (section 4.4) is public too. This text is pinned on IPFS, and its hash is in the vote's description.

### 3.4

## Retiring the current DAO

The DAO has three multisigs, all 3-of-5 with the same five signers who have legal agreements with the API3 Foundation: the Vault, the Agent and the Treasury (Figure 3). The [Vault](#) holds 2,746 wstETH and about \$1.4M of USDC in lending vaults, and owns api3.eth. Many stakers delegated to it in the old DAO, about 15M API3 of stake today, so the Vault votes on Snapshot with that stake, and it posts the team's proposals. The [Agent](#) handles day-to-day transactions like contributor payments, DeFi positions, owns the Morpho vaults we curate, and only holds what those need, including about 0.35M API3. The Treasury is new. It holds the DAO's API3 and nothing else: the final mint, the Aragon primary agent's balance, and whatever API3 the Vault holds that isn't earmarked for other things, which the Vault moves over right after execution, before the first Snapshot proposal. The Treasury never posts proposals. The Treasury and the Agent both delegate their Snapshot voting power to the dead address. When a budget is approved, the cycle proposal being the usual example, funds move from the Vault or the Treasury to the Agent, which pays out from there. Section 5.1 sets out what the multisigs can do without a vote each time.

MOVED ON EXECUTION

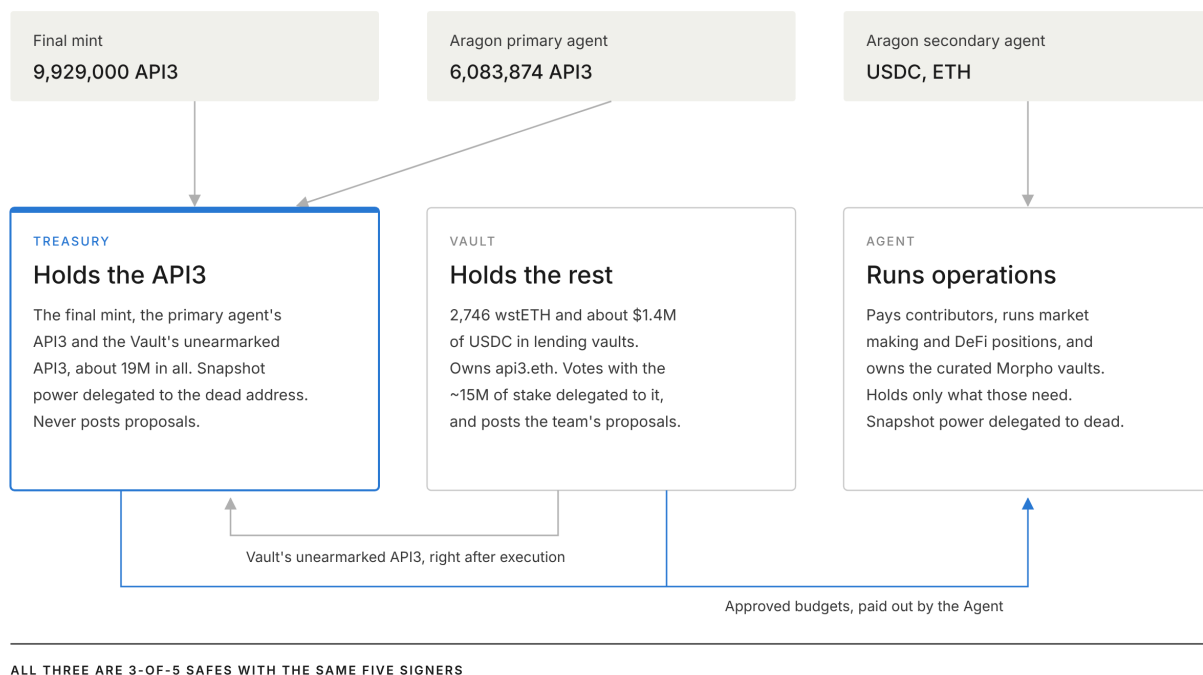


**Figure 3.** The three multisigs after execution

Balances from the DAO tracker, 29 October 2026. The Treasury and the Agent delegate their Snapshot voting power to the dead address; the Vault votes only with stake delegated to it in the old DAO.

On execution, whatever the two Aragon agents (the contracts that hold the old DAO's funds) still hold moves over. The primary agent's 6,083,874 API3 goes to the Treasury along with the final mint, and the secondary agent's 59,420 USDC and 0.06 ETH go to the Agent. Its PNK and wPOKT move to the Agent through secondary votes 133 and 134, posted on 27 October, which execute when they close on 3 November whether or not this vote has executed by then.

The right to create votes on both voting apps is revoked and the permission that controls it is left without a manager, so no vote can ever be created in the old DAO again. As a second lock, the passing requirement on both apps is raised to the highest value the contract accepts, so even a vote created in some way we haven't foreseen couldn't pass, and the DAO's root permissions are revoked as well. The staking pool keeps running. Staking, unstaking, withdrawing and delegating all still work; the pool just doesn't mint anything.

### 3.5 What happens to current stakers

Staked API3 stays in the pool and keeps its vote in the new system, one token one vote, the same as tokens in a wallet, with old delegations counting as well. Rewards already earned keep vesting on their existing schedule and can be claimed as they vest. No new rewards are paid after execution. The pool parameters do not change. The unstake wait is one epoch, seven days. The pool doesn't accept a wait shorter than its epoch, and the epoch length is fixed in the contract, so the wait can't be shortened. The one-year lock on minted rewards stays as it is.

We recommend that stakers unstake and, once their tokens unlock, withdraw anything they aren't comfortable leaving in a deprecated contract. Nobody is forced out and nobody is locked in, but the pool won't be maintained, and section 2.2 explains why we'd rather nobody's tokens depended on it for longer than necessary.

The DAO contracts stay on chain indefinitely and remain ownerless. The [DAO dashboard](#) stays online until the end of March 2028 with a deprecation notice pointing to the Snapshot space, and only unstake, withdraw and undelegate remain available on it. Stakers will be reminded to unstake and withdraw before then. The [DAO tracker](#) stays up over the same period as the record of the old DAO, its votes, positions and rewards. After March 2028 both are retired and we publish step-by-step Etherscan instructions for calling `scheduleUnstake`, unstake and withdraw on the pool directly.

|                                        | TODAY                              | AFTER EXECUTION                               |
|----------------------------------------|------------------------------------|-----------------------------------------------|
| <b>Total supply</b>                    | 190.1M, growing                    | 200M, capped                                  |
| <b>New API3 per year</b>               | ≈50M (26%)                         | 0                                             |
| <b>Maximum supply on CoinGecko</b>     | Infinite                           | 200,000,000                                   |
| <b>Who can mint</b>                    | The staking pool                   | Nobody                                        |
| <b>Who votes</b>                       | Stakers only                       | Every holder                                  |
| <b>Voting power needed for control</b> | half of what is staked (≈36M API3) | Section 4.3 and the charter                   |
| <b>Execution</b>                       | Instant, on chain, no delay        | By the multisigs within seven days of passing |

Table 2. Before and after execution.

# How the DAO operates

---

## 4.1

### Voting power

Today you have to stake to vote, and you're paid for staking. After this proposal you vote with the API3 you hold, whether it's in a wallet, staked in the old pool (unvested rewards included), deposited in the pool but not staked, waiting out the unstake period, or posted as collateral in the Kabu API3/USDC market on Morpho at 62.5% LTV ([market 0x70df...6c67a1](#)). Borrowing against your API3 doesn't cost you your vote. Nothing is locked, and there's no reward for holding API3 or for voting.

The treasury's own API3 doesn't vote. The Treasury and the Agent delegate their voting power to the dead address, the Vault moves its own unearmarked API3 to the Treasury before the first Snapshot proposal, and from then on the Vault votes only with the stake its delegators left with it in the old DAO. The 5.1M API3 lent to third parties is in their wallets, so technically it can vote. Future agreements that put treasury API3 in someone else's hands will prevent governance use.

A vote on Snapshot is a message you sign with your wallet. It's free and recorded off chain. Voting power is read at a block the author picks when posting, at most 7,200 blocks, about a day before Snapshot receives the proposal. The Snapshot website picks the latest block, which satisfies this, and the signers check it. Nothing that happens after that block, a transfer or a change of delegation, affects voting power for that proposal.

Delegations made in the old DAO carry over for staked tokens, so nobody has to come back and delegate again. Everything else is delegated on Snapshot under `api3.eth`, and a Snapshot delegation moves all the voting power you have to your delegate. While you're delegated your own vote counts zero, so to vote yourself you take the delegation back first. Snapshot delegations don't chain: if you delegate to someone who has delegated to a third person, your votes stay with the person you picked. A Snapshot delegation can be changed or withdrawn at any time. A delegation in the staking pool can be changed once a week, which is a rule of the pool contract. The exact strategies are published in the space settings.

## 4.2

### Proposals

Proposals go in the [API3 space](#) on Snapshot, which is tied to `api3.eth`. There's one kind of proposal and one set of rules for all of them. Anyone with at least 500,000 API3 of voting power, held or delegated (0.25% of supply), can post. Snapshot limits how long a proposal can be, so the full text goes on IPFS and the Snapshot post has a summary and a link.

A proposal has to say exactly what it asks for: the action, the amount, the recipient, the timing and what counts as done, as far as each applies. A rule change has no amount or recipient, but it still says exactly which text or setting changes and to what. "Spend more on marketing" isn't a proposal, but "pay X to Y by date Z for W" is. The signers won't execute a proposal that leaves any of these open. Discussion happens in the governance channel on the [API3 Discord](#). A proposal doesn't have to be discussed before it's posted. Discussion can happen while the vote runs, or not at all.

Voting opens as soon as a proposal is posted and runs for seven days, the same as in the old DAO. We considered a delay before voting, but seven days of voting already gives holders time to see a proposal and prepare accordingly. We also considered letting only known addresses post. That would close off some attacks, but it would turn the DAO into a committee, so we decided against it. The space has no admins, members or moderators. Its settings can only be changed by its controller, the Vault, which owns api3.eth, and the Vault only signs changes that a proposal has approved. Spam proposals are removed by Snapshot's review, which comes with the Pro plan, or by their author.

## 4.3

### Quorum and passing

The current DAO measures its thresholds against staked tokens. A secondary proposal needs 15% of all voting power in favour, about 10.9M API3 at today's stake of 73M, and every routine proposal since 2021 has passed under that rule. A primary needs more than half, about 36M. After execution the voting base is the whole supply minus what the Treasury holds, about 181M, and 15% of that would be 27M, which most past votes wouldn't have reached. So the quorum is a fixed number of tokens instead, close to what the old rule means today: 10,000,000 API3 in favour.

Past votes support that number (Figure 4). Every secondary that passed in 2025 and 2026 drew between 16% and 41% of voting power, which works out to 12M to 30M API3 at today's stake, with a median of 22%. The two that failed drew 4% and 7%, under 5.5M. Ten million is below every proposal that passed in that period and above both that failed.

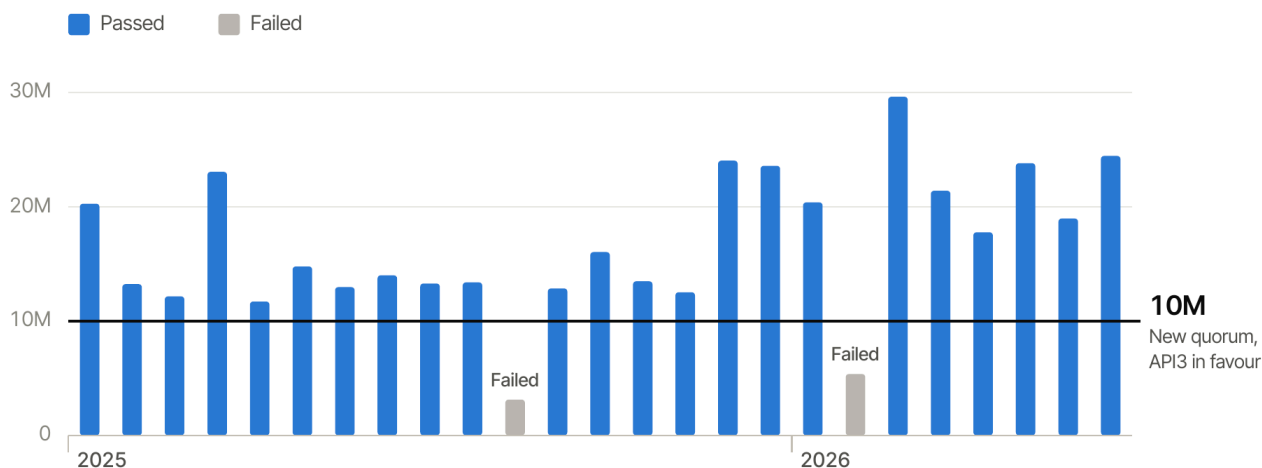


**Figure 4.** Votes in favour of past secondary proposals, against the new quorum

Votes in favour of each secondary proposal in 2025 and 2026, as a share of all voting power, expressed in API3 at today's 73M staked. From the DAO tracker.

The quorum only counts votes in favour. If abstentions counted, a proposal could pass with one vote for and 9,999,999 abstaining. After this proposal a lot of API3 that has never voted will be able to, including tokens held by exchanges, and some of them abstain as a matter of policy. That's fine in itself, but an exchange's abstention shouldn't be what gets a proposal over the line. If an exchange votes in favour, people can at least ask it why. So there's no abstain option. A proposal passes with at least 10,000,000 API3 in favour and more votes for than against. In the space settings that's a quorum of 10,000,000 with abstain hidden. Snapshot's own quorum counts every vote cast, so the verifier counts the For choice only, and it ignores any vote cast as abstain by other means; such a vote doesn't count toward quorum and doesn't invalidate the proposal. We'll review the quorum after the first three votes. Changing it is a rule change, with the 30-day delay described in section 4.4.

## 4.4

### Execution

A Snapshot vote doesn't move any funds by itself. The three multisigs hold the DAO's assets, execute what passes within seven days, and otherwise act only under the operating authority in section 5.1. Before executing, the signers check that the proposal reached quorum, passed, was posted by an address with enough voting power, is fully specified, and isn't ruled out by the Foundation's charter in section 5. Balancer and Sushi run their governance the same way, with an off-chain vote and a multisig that executes, and this setup doesn't need any new contracts.

The signers don't take the result from the Snapshot website. They recompute it from the chain with an [open-source tool](#) we published for this. It recalculates every voter's power at the proposal's block, re-tallies the vote, checks every signature, and checks the proposal against the rules in this section, including the block window. That check is needed because Snapshot itself accepts any past block. A proposal that gets around the space's settings like this fails the

check and isn't executed. An individual vote cast outside the settings, such as an abstain, is simply not counted. Because the verifier decides what counts, its rules are part of the DAO's rules, and changing them is a rule change (see below).

A proposal that passes these checks is executed within seven days. If the signers can't manage that, they say so publicly before the seven days are up, with the reason and a new date no more than seven days later, and they can do this once per proposal. Letting the deadline pass without saying anything, or missing the new date, is a breach of their mandate.

Changes to the rules themselves take longer. A rule change is a proposal that changes any of these: section 4, the mandate and its operating authority, the charter, what counts as voting power and the Snapshot strategy behind it, the space settings, the verifier's rules, the ownership of api3.eth, or which multisig holds the treasury or executes, except an emergency move to a new Safe under section 5.3, which is confirmed afterwards. A rule change takes effect 30 days after it passes rather than seven, at the same threshold. The extra time means a rule change pushed through with borrowed tokens can be noticed and reversed before it takes effect. A proposal that does nothing but cancel a pending rule change isn't delayed, and is executed in the normal seven days, so the reversal lands first. Anyone with 500,000 API3 can post it. A change nobody objects to just waits three weeks longer. Figure 5 shows how the timings work.

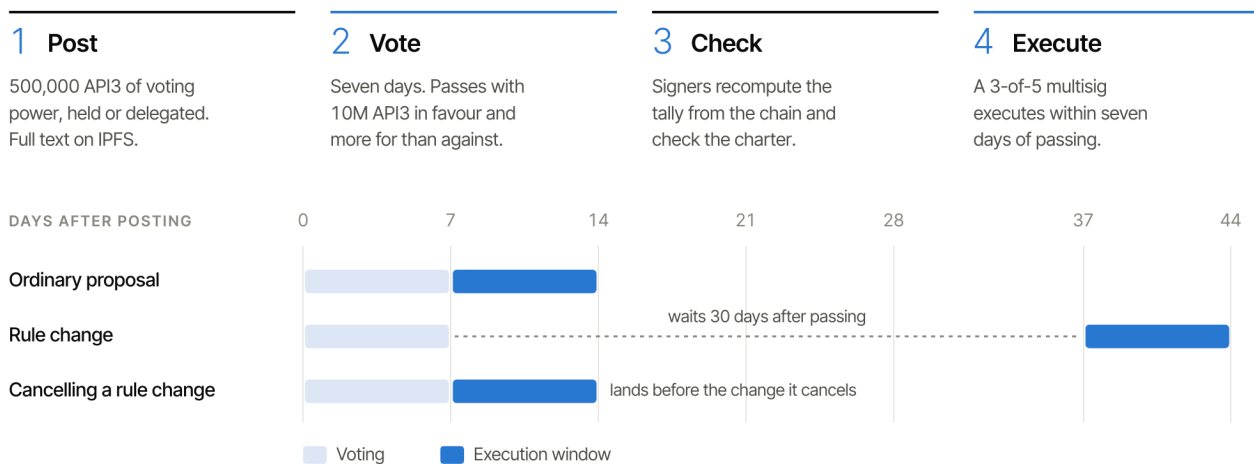


**Figure 5.** How a proposal runs, and how long each kind takes

A rule change is any change to section 4, the mandate, the charter, the voting strategy, the space settings, the verifier, api3.eth or which multisig holds the treasury.

|              | TODAY                    | ON SNAPSHOT                                                                                                   |
|--------------|--------------------------|---------------------------------------------------------------------------------------------------------------|
| Who votes    | Stakers only             | Every holder                                                                                                  |
| Cost to vote | Stake, then gas per vote | Free                                                                                                          |
| Voting power | Staked pool shares       | One token, one vote, held, staked in the old pool or posted as collateral in the Kabu API3/USDC Morpho market |

|                           | TODAY                                     | ON SNAPSHOT                             |
|---------------------------|-------------------------------------------|-----------------------------------------|
| <b>Threshold</b>          | 15% or more than 50% of staked            | 10M in favour, more for than against    |
| <b>Proposal rights</b>    | 0.1% of voting power, one proposal a week | 500,000 API3 (0.25%), held or delegated |
| <b>Execution</b>          | Instant, on chain                         | By the multisigs within seven days      |
| <b>Changing the rules</b> | Primary vote                              | Proposal, effective after 30 days       |

Table 3. Governance today and on Snapshot.

## 4.5 The rules at a glance

Table 4 lists every rule the DAO runs on from the execution block. Together with the charter in section 5, that's the complete set.

| RULE                      | SETTING                                                                                                                                                                                                                                                                                  |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Voting power</b>       | One API3, one vote. Counted in wallets, in the staking pool staked or merely deposited, including unvested rewards and tokens in the unstake wait, and as collateral in the Kabu API3/USDC Morpho market. Delegable; a delegation in the pool can change once a week.                    |
| <b>Snapshot</b>           | Voting power and delegations are read at a block the author names at posting. Old DAO delegations carry over for staked tokens; wallet tokens delegate on Snapshot. The Treasury's and the Agent's power is delegated to the dead address.                                               |
| <b>Who can post</b>       | 500,000 API3 (0.25% of supply), held or delegated.                                                                                                                                                                                                                                       |
| <b>Format</b>             | Action, amount, recipient, timing and what counts as done, as far as each applies. Full text on IPFS, summary and link on Snapshot.                                                                                                                                                      |
| <b>Timing</b>             | No delay. Seven days of voting from posting.                                                                                                                                                                                                                                             |
| <b>Quorum</b>             | 10,000,000 API3 in favour. Abstain is not on the ballot.                                                                                                                                                                                                                                 |
| <b>Passing</b>            | Quorum reached and more for than against.                                                                                                                                                                                                                                                |
| <b>Execution</b>          | By the multisigs within seven days of passing, after checking the rules above and the charter. If execution slips, the signers say why before the window closes and set a new date within seven more days, once. Day-to-day operations run under the operating authority in section 5.1. |
| <b>Charter</b>            | The ten clauses in section 5.2, binding on the signers from the execution block. The signers are publicly identified and legally accountable for following it.                                                                                                                           |
| <b>Changing the rules</b> | By proposal, taking effect 30 days after it passes. A proposal that only cancels a pending rule change is executed in the normal seven days.                                                                                                                                             |

| RULE                        | SETTING                                                                                                                                                                                                                                                                                                            |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Replacing a multisig</b> | By a proposal naming the new Safe and its signers, taking effect 30 days after it passes. The outgoing multisig hands over assets, positions, access and runbooks within 30 days of execution. Signers and threshold are up to the multisig, with a majority threshold; emergency moves allowed under section 5.3. |
| <b>Discussion</b>           | The dedicated channel on the API3 Discord. A forum can be added later by proposal. Optional, before or during a vote.                                                                                                                                                                                              |
| <b>Reporting</b>            | Treasury positions on the API3 website, votes on the Snapshot space. The DAO tracker covers the old DAO until it is wound down.                                                                                                                                                                                    |

*Table 4. The rules of the DAO from the execution block.*

# The mandate and the charter

---

## 5.1

### The mandate

This proposal gives the three multisigs a mandate: hold the treasury on behalf of the DAO, execute what passes on Snapshot under the rules in section 4 within the execution window, run the DAO's day-to-day operations under the operating authority below, and do nothing else and nothing unlawful. The charter below sets the limits of that mandate. It's part of this proposal, so voting for the proposal means voting for the charter, and it binds the signers from the execution block.

#### Operating authority

Some things can't wait for a seven-day vote, and others happen every week. Without a separate vote, the multisigs may pay what an approved budget covers, including contributor pay and vesting; move assets between the three multisigs and the DAO's own Safes on other chains; deposit into, withdraw from and run the Morpho vaults we curate, and manage the positions the treasury already holds, such as the Uniswap API3/ETH position and its lending positions, as the cycle proposal sets out; make the monthly API3 purchases; and pay gas. If funds are at risk, for example in an exploit, they may pull any position out at once and report it within a day, and if a multisig itself is compromised they may move what it holds to a new Safe, as section 5.3 sets out. Everything done this way is reported on the website. Anything new, such as a new protocol, counterparty, kind of position or budget, needs a proposal.

## 5.2

### What the multisigs will not execute

The charter lists things the signers won't execute however a vote turns out, so that buying enough votes isn't enough to get them done. Clause 1 rules out proposals that harm holders as a group, and clause 2 rules out paying the treasury out to holders. Clause 3 rules out paying the people who voted for a proposal, other than under budgets and agreements already approved. Clauses 4 and 5 rule out moving treasury assets anywhere no proposal or the operating authority names, or trading them at off-market prices. Clause 6 caps what a vote can spend, clause 7 rules out vague proposals, and clause 8 says how a multisig is replaced and keeps the choice of signers out of governance. Clause 9 rules out anything that would break the law or a contract the API3 Foundation has signed, and clause 10 delays rule changes by 30 days so the charter can't be rewritten and then used straight away. If a proposal falls under the charter, the signers don't execute it, and within the execution window they say publicly which clause it falls under and why.

Clause 2 rules out a payout even when every holder gets the same share. Paying out a treasury makes sense when a token trades below what the treasury holds per token. That's what happened to Aragon, and it's what makes a raid worth

attempting. API3 trades well above that. Its circulating market cap is about \$38M, against a treasury of about \$14M excluding its own API3. A payout would hand each holder a fraction of what the market already pays for their token, and shut down the business that accounts for the rest of the price. Holders would get less, and only someone who had bought in below the payout value could come out ahead, which isn't possible while the token trades this far above it.

#### CHARTER

#### *The multisigs will not execute a proposal that*

- 1** Harms API3 holders, including anything that restricts their ability to acquire or dispose of API3.
- 2** Distributes treasury assets to holders, whether pro rata to all of them or to a few.
- 3** Pays the proposer, the addresses that voted for it, or anyone acting for them, unless the payment falls under a budget or agreement the DAO has already approved or is for work delivered at market rates, and the proposal discloses the connection.
- 4** Moves treasury assets to an address, custody or contract that neither the proposal itself nor the operating authority in section 5.1 names, or gives anyone standing permission to move them.
- 5** Trades a treasury asset with a named counterparty at worse than market terms at execution, sells API3 from the treasury below market, or buys anything above it.
- 6** Commits more than 5% of the treasury's value in one vote, or more than 15% of it in any 90 days, outside the budget in the cycle proposal. Every outflow to someone outside the multisigs counts, including deposits, loans and swaps. The treasury's value, counting the API3 it holds, is its 30-day average as reported on the API3 website. The cycle proposal is the team's budget for the next three months of operations, and only one is in force at a time.
- 7** Leaves the action, amount, recipient, timing or what counts as done undefined, as far as each applies.
- 8** Replaces a multisig without naming the new Safe and its signers, or otherwise changes who signs on one, except as section 5.3 allows.
- 9** Reasonably requires the signers to break the law or a contract the API3 Foundation is party to.

Is a rule change as defined in section 4.4 and is executed before 30 days have passed since it passed. A proposal that only cancels a pending rule change is executed in the normal seven days.

*Charter. Binding on the signers from the execution block and by contract with the API3 Foundation.*

Nothing on chain enforces the charter. What enforces it is that the signers are known people who are contractually obligated by the API3 Foundation to follow it.

## 5.3

### Changing the mandate and the signers

Everything in section 4, the mandate and this charter can be changed by proposal, as a rule change with the 30-day delay described in section 4.4. Who signs on a multisig is up to the multisig, as it is today. The signers can add, remove or replace a signer, or change the number of signers and the threshold, at any time without a proposal, for example when someone leaves, as long as the threshold stays a majority of the signers: more than half, rounded up, so 3 of 5, 4 of 6 or 7, 5 of 8 or 9, 6 of 10. They announce any change within seven days. If a multisig is compromised or can no longer operate, the signers may move what it holds to a new Safe at once and report it within a day, and a proposal to confirm the new Safe follows within 30 days. What a proposal can do is replace one multisig with another. That proposal names the new Safe and its signers, so holders know who they're voting for, and it waits 30 days like any rule change. Once it's executed, the outgoing signers have 30 days to hand over every asset, a list of every position and counterparty, access to every service the multisig runs, and the runbooks for the transactions it makes. The handover follows the vote. It isn't a condition the outgoing signers can withhold, and failing to do it is a breach of the charter like any other. All of the foregoing and this section generally is subject to the API3 Foundation's bylaws and agreements with the applicable signers.

## 5.4

### Attack surface before and after

Moving to Snapshot does remove one protection. Voting power no longer has to sit in a pool for seven days, so someone can buy or borrow tokens for the length of a vote and sell them afterwards. The old DAO had this problem too, with a smaller voting base, and a worse one on top: instant execution against a token contract that could mint without limit.

After execution there's less to gain from an attack, and it's harder to pull off. There's nothing left to mint. The treasury is in 3-of-5 multisigs that execute within a window rather than in the block the vote closes. The charter rules out proposals that would pay the voters or pay the treasury out to holders, and rule changes wait 30 days, long enough for a cancellation to land first. What's left for

an attacker is a proposal within the spending limit in clause 6, passed with rented tokens, with the signers and everyone else able to see who voted for it.

## 5.5

### Snapshot as a trusted party

Voting off chain means trusting Snapshot, as Balancer, Sushi, Yearn, Aave and most other large DAOs already do. That trust is limited. Votes are EIP-712 signatures stored on IPFS, balances and delegations are fixed at the proposal's block, and anyone can recompute the tally from the chain, so Snapshot can't forge or change a vote. It can refuse one, though. Its sequencer decides which votes it accepts, and a refused vote simply doesn't appear. Downtime can delay a vote but can't change it. The space is tied to api3.eth, which the Vault owns. Execution moves from the Aragon contracts to the multisigs, where nothing on chain enforces the result, so the mandate and the charter make following the recomputed tally the signers' duty. API3's treasury already works this way.

If Snapshot Labs shut down, the DAO wouldn't lose anything it depends on. Voting power comes from API3 balances on chain, delegations are in the on-chain delegate registry, past votes are on IPFS, and the space's identity is api3.eth. Snapshot's hub, sequencer, indexer and interface are open source under the MIT licence in [Snapshot's monorepo](#), so we or anyone else could run them, although they're documented for development rather than production and we don't know of anyone running a private instance at scale. The more realistic fallback is one of the other signature-based voting tools, pointed at the same balances and delegations. Because moving off Snapshot would itself need a Snapshot vote, there's a rule for that case: if Snapshot can't accept proposals or votes for 14 days in a row, a vote run on such a tool with the same strategy, rules and verifier counts as a Snapshot vote. Either way, the rules in this document would carry over without touching the token or the treasury.

## Token economics after execution

Table 5 and Figure 6 set out the supply of API3 before and after execution. Supply figures are rounded to the nearest token; figures marked  $\approx$  are estimates.

|                                          | API3                 | NOTE                                                                                                                                                                                           |
|------------------------------------------|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Genesis supply, 2020                     | 100,000,000          |                                                                                                                                                                                                |
| Minted as staking rewards to 29 Oct 2026 | 90,071,000           | 47% of current supply                                                                                                                                                                          |
| Total supply, 29 Oct 2026                | 190,071,000          | after that day's weekly reward                                                                                                                                                                 |
| Final mint                               | 9,929,000            | 200,000,000 minus supply at posting                                                                                                                                                            |
| Maximum supply after execution           | 200,000,000          | capped, no minters, owner is the dead address                                                                                                                                                  |
| Held by the treasury after execution     | $\approx$ 27,000,000 | About 19M in the Treasury once the Vault's unearmarked API3 has moved, 0.35M in the Agent, 1.7M in the Uniswap API3/ETH position, 5.1M on loan to third parties and 0.9M in cancelled vestings |
| Emissions after execution                | 0                    |                                                                                                                                                                                                |

Table 5. Supply of API3 before and after execution.

After execution nothing in the system can create API3. The token contract has no minters, and its owner is an address nobody controls. The treasury's API3, including the final mint, sits in the Treasury, apart from the amounts Table 5 lists elsewhere, and is reported on the API3 website. Revenue from the business buys API3 on the market every month, and what's bought goes to the Treasury for grants, vesting and other costs paid in API3.

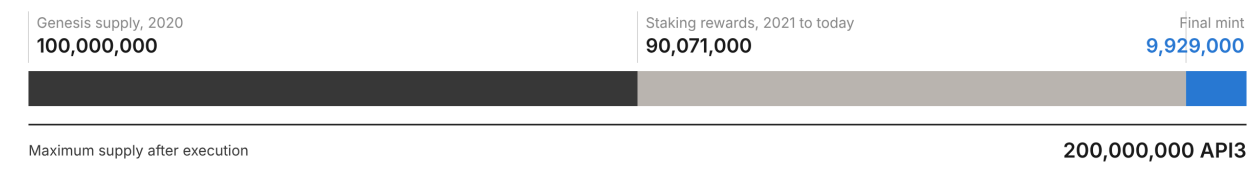


Figure 6. Where the 200,000,000 comes from

Total supply at posting on 29 October 2026, after that day's weekly reward, rounded to the nearest token. Any reward minted between posting and execution is burned from the multisigs, so the total stays 200,000,000.

# Curation

---

Sections 7 and 8 describe the business as it is today. They're here for context. The vote doesn't decide anything about them, and both will keep changing.

## 7.1

### What curation is

For five years our oracle business worked like this: we integrated a chain, then the lending and derivatives protocols on it, and hoped they would attract the deposits that make an oracle worth paying for. The protocols owned the relationship with the capital, and most of the value went to them. With curation it's the other way round. A curator runs lending markets. It sets the risk parameters, picks the collateral and the oracle, and earns a share of the interest borrowers pay. Instead of waiting for someone else to bring capital to our feeds, we bring it ourselves.

We think we're well placed to do this. Most chains still need oracles, so we already have integrations and relationships on the ones that matter. The markets we curate read from feeds we built, served by API providers who sign their own data, and we run our own searcher and [OEV](#) operation, which recovers value when those feeds update and returns it to the protocols that use them. Running oracles and OEV means we see up close how lending markets fail, which most curators don't.

## 7.2

### Our markets today

In practice this means we can open markets other curators can't. A market needs a reliable feed for its collateral and a liquidation path that actually works. Curators who rely on someone else for both tend to list the same assets as everyone else and compete on rate. We can list an asset as soon as there's a feed for it, including on chains other curators don't cover, with our own searchers handling liquidations and the OEV going back to the market instead of to whoever is fastest. That lets us build markets around a niche, and a niche is what attracts deposits. Without meaningful deposits, curation fees stay small.

Today we run six vaults on Ethereum, Base and Robinhood, with collateral ranging from blue chips to governance-enabled COMP and long-tail assets. We charge a 5% performance fee on borrower interest and no management fee, so we only earn when the markets are used. Each market is designed around its feed, its liquidation path, collateral liquidity and supply caps, and adjusted as conditions change.

Curation started on our own treasury and was managing outside capital within six months, during a bear market with very little borrowing demand (Figure 7).

In January the vaults held about \$13.7M, almost all of it ours. On 22 October they held \$20.5M, of which \$15.7M, or 77%, was outside capital, while our own deposits had come down to \$4.8M as the treasury moved back into ETH. The numbers are reported in each [cycle proposal](#) and on the website. If borrowing demand picks up, the same markets earn more without much extra cost. Oracle, OEV and curation revenue all come from the same activity, and that revenue funds the monthly API3 purchases.

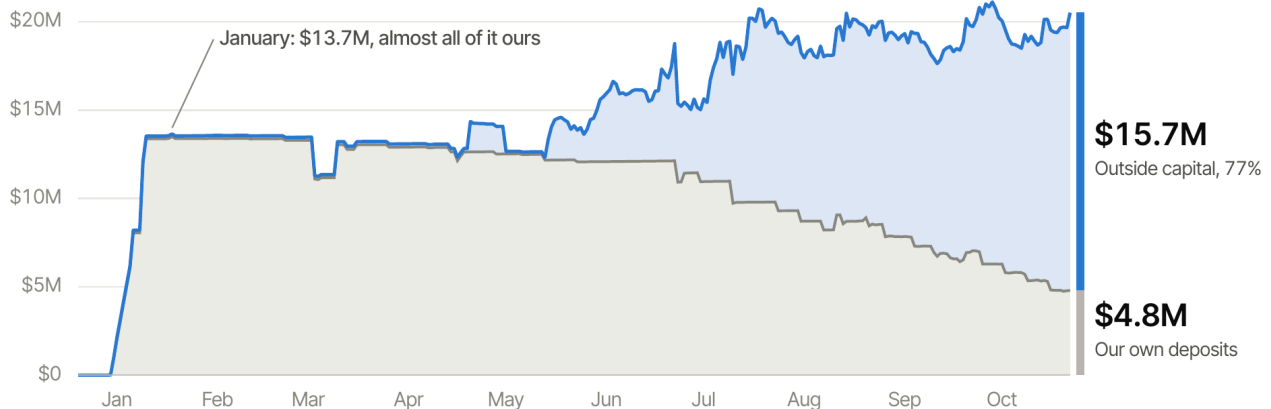


**Figure 7.** Capital in API3-curated vaults: our own deposits and outside capital

Six API3-curated Morpho vaults on Ethereum, Base and Robinhood Chain, daily from the Morpho API to 22 October 2026. Own deposits are what the DAO's multisigs and wallets hold in them; everything else is outside capital.

We have more ideas for creating demand on chain that other curators are leaving alone, and we'll be building them. What we won't do is follow the many curators adding RWAs and off-chain yield, which bring issuer, custody and legal risk into DeFi. Expect more curation ideas from us along these lines, all of them on chain.

## 8.1 Agents as customers

AirnodeHub is where we see the most room to grow. AI agents, programs that browse, compare and buy on someone's behalf, now make up a large share of internet traffic. Bots, agents among them, were 57.5% of web requests in June by Cloudflare's count, 70% of API calls at Stripe now come from agents, and traffic from agents taking actions grew 7,851% in a year ([Fortune](#)). A growing share of them pay for APIs per call, in stablecoins, over protocols like [x402](#), which processed 169 million payments in its first year.

## 8.2 The provider problem

Right now the people who own the data mostly aren't the ones getting paid for it. An agent can't tell an original source from a copy, so providers get scraped instead of paid, and publishers now see [227 crawler visits](#) for every visitor sent back. One fix is for providers to sign their data at the source, so anyone downstream can check where it came from and pay the right party. That's what our first-party oracles have done for five years. API providers sign the data they serve, and we've spent those years building direct relationships with them.

## 8.3 What AirnodeHub does

AirnodeHub does the same for APIs in general. A provider lists its existing API, and every response is signed with the provider's own key. Agents can find an API, see who signs it, call it and check the signature before using the data, through the API or an MCP server. A signature shows where the data came from and that it wasn't altered. It doesn't show that the data is correct, and the hub says so. The design is described in [the launch post](#). It's live at [airnodehub.api3.org](#) in early access, with the first providers on it and calls free while per-request payments are being built. We expect the details to change as we learn what agents and providers need.

Others are working on this too. In June [Orthogonal](#) raised \$4.3M from Pantera and Y Combinator to let agents discover, call and pay for APIs, with 35 APIs on board and [x402](#) among its payment options. What we bring that a new team doesn't have is the provider relationships and the signing infrastructure, which has been running in production since 2021. The goal is a marketplace where whoever produces the data gets paid for it, with the hub taking a cut.

## What it means for API3

Once per-request payments are live, providers keep most of the revenue and we take a service fee on what goes through the hub. Part of the final mint may go toward growing it. We want AirnodeHub to become our largest source of revenue, and that revenue would buy API3 the same way OEV, curation fees and ETH yield do today. If agents end up buying a large share of the world's data, that's a much bigger market than on-chain oracles have ever been.

## Treasury and revenue

Two years ago revenue covered none of our costs. Since then, through a long bear market, OEV, curation fees and yield on the treasury's ETH (bought near the lows) have grown to cover part of them, as much as 62% in May and about a fifth in August and September. The [treasury](#) holds several years of runway at the current burn rate, mostly in ETH, stablecoins and API3. That estimate assumes the ETH price stays where it is. Since September, revenue has been used to buy API3 on the market every month, and every position and each month's revenue is reported on the website.

Leaving out the API3 it holds, the treasury has grown from \$12.9M at the start of December to \$13.8M on 23 October, even though costs were about \$1.4M more than revenue over the same period (Figure 8). In December it sold nearly all of its ETH at about \$2,800, and since late June it has bought back more than it sold. It now holds about 4,420 ETH, against 2,442 at the start of December. That counts the Vault's wstETH at its ETH value, the WETH in the Kabu vault on Base, the ETH side of the Uniswap API3/ETH position and the operational wallets. Counting its API3, including the 5.1M API3 on loan to third parties, the treasury is worth about \$18.4M.

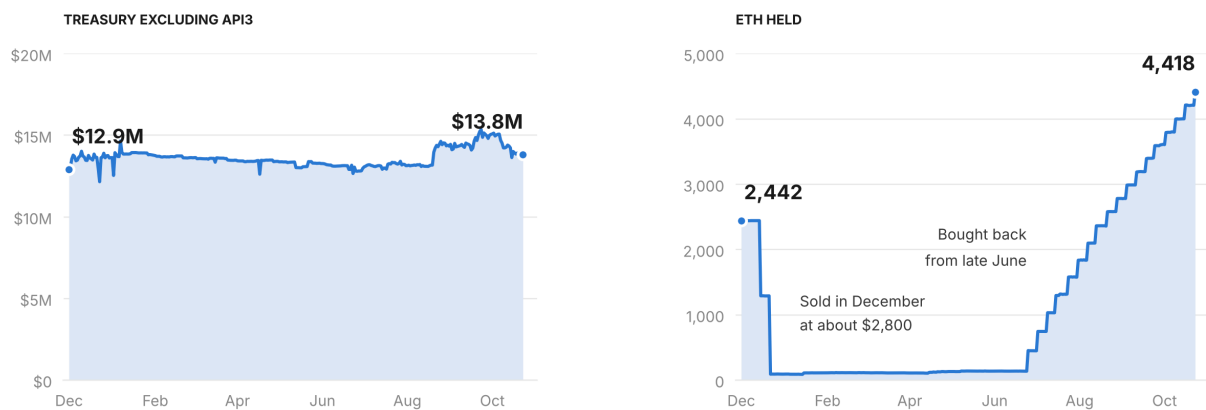


**Figure 8.** The treasury excluding API3, and the ETH it holds

Daily, 1 December 2025 to 23 October 2026, from the DAO tracker. The left panel leaves out the API3 the treasury holds, including its side of the Uniswap position and the cancelled vestings. wstETH is counted at its ETH value.

## Summary

---

For five years the value of API3 has been driven more by the mint than by the business. Whatever we shipped and earned landed on a supply growing 26% a year, controlled by stakers, and the market priced it as a token that could be inflated or captured. CoinGecko's infinite maximum supply and the all-time low in August both reflect that.

After this proposal the supply is capped at 200,000,000, the treasury has several years of runway, and the business buys API3 every month with revenue that has gone from covering none of its costs to covering part of them. We'll keep working in DeFi and build on our API experience for agents.

Products will keep changing, so it makes sense to settle the supply first. A business that buys its own token every month shouldn't be buying into a supply that grows faster than it can buy.

---

*If you have staked API3, please vote.*

---

## Risks and open items

---

|                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>The vote fails</b>               | Emissions keep winding down under the 30% staking target and any further cuts, but the risks in section 2.2 stay. The proposal can be put up again while the supply is still under 200,000,000.                                                                                                                                                                                                                                                                                             |
| <b>Stake leaves after execution</b> | Without rewards, some stakers will unstake over the following weeks. That's expected. Unstaking takes one epoch, seven days, which is fixed in the pool contract, and staked tokens keep their vote either way.                                                                                                                                                                                                                                                                             |
| <b>Treasury</b>                     | The treasury sits in three 3-of-5 multisigs with the same five signers, as most of it already does today. The Treasury holds the API3, the Vault holds the rest, and the Agent holds what day-to-day operations need. They move funds when a proposal passes on Snapshot or under the operating authority in section 5.1, and not otherwise. The signers are publicly identified and legally accountable. Every position is reported on the website, and the voting rules are in section 4. |
| <b>Turnout</b>                      | Old DAO delegations only carry over while the tokens stay staked. As stakers unstake, which we recommend, that delegated power disappears unless the holder delegates again on Snapshot, and early votes may struggle to reach 10,000,000 in favour. We'll ask holders to vote and to delegate. If the quorum becomes increasingly hard to reach, it can be lowered by proposal, as a rule change with the 30-day delay.                                                                    |
| <b>No lockup</b>                    | On Snapshot nobody has to lock tokens to vote, so voting power is cheaper to buy or borrow around a proposal's block than it was with staking. Section 5 covers how that's handled: there's nothing left to mint, the treasury is in the multisigs, the charter lists what the signers won't execute whatever the vote, and rule changes wait 30 days.                                                                                                                                      |

# Timeline

| WHEN                    | WHAT                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 Sep 2026              | Secondary proposal #132 to cut the staking target to 30% posted                                                                                                                                                                                                                                                                                                                              |
| 11 and 12 Sep 2026      | #132 passes on the 11th and is executed on the 12th. Wind-down of emissions begins.                                                                                                                                                                                                                                                                                                          |
| Before posting          | The api3.eth space created and configured by the Vault, the voting verifier published, the Treasury's and the Agent's Snapshot power delegated to the dead address.                                                                                                                                                                                                                          |
| 29 Oct 2026             | Weekly reward minted. Then, in this order: total supply read and the final mint filled into this text, the text pinned on IPFS, the vote bytes built from the published source, both repositories public, the DAO dashboard release that shows the vote, and the vote posted with the text's hash in its description. It executes the moment more than 50% of all voting power is in favour. |
| 5 Nov 2026              | Next weekly mint. If the vote hasn't been executed by then, the DAO burns the excess from its multisigs right after execution, so the supply still ends at exactly 200,000,000.                                                                                                                                                                                                              |
| Execution block         | Final mint, minters removed, token ownership given up, treasury moved, old DAO retired, Snapshot live                                                                                                                                                                                                                                                                                        |
| With execution          | The Vault moves its unearmarked API3 to the Treasury. The new website goes live: the capped supply, the new DAO, Snapshot and the treasury. Products follow later.                                                                                                                                                                                                                           |
| Until end of March 2028 | DAO dashboard online for unstake and withdraw, DAO tracker online as the record of the old DAO, then both retired with Etherscan instructions                                                                                                                                                                                                                                                |

Table 6. Timeline.

## A

# What the proposal submits on chain

The primary proposal is a single Aragon vote of 26 calls in the twelve rows below. When it passes, the calls run in this order in one transaction. Either every step succeeds or none does. The two Aragon agents are the contracts that hold the old DAO's funds and act on its behalf, and the ACL is the permission list that says which contract may call which. Calls to the token, the secondary agent and the voting apps go through the primary agent, which owns the token and holds the permissions. Calls to the ACL and to the primary agent itself come from the primary voting app, which manages those permissions. Every permission named here was read from the ACL on 2 Sep 2026. The calldata, the raw instructions the vote carries, is in the vote on chain, and the repository's decoder prints it call by call for comparison with this table.

|   | CONTRACT AND CALL                                                                                                                | WHAT IT DOES                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---|----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | ACL 0x1E7e...195A, grantPermission(primary agent, secondary agent, TRANSFER_ROLE) and the same for EXECUTE_ROLE                  | Gives this proposal the right to move what the secondary agent holds. TRANSFER_ROLE covers the tokens. The ETH needs EXECUTE_ROLE: Aragon sends ETH with a 2,300 gas stipend, which a Safe cannot receive, so the ETH leaves through <code>execute</code> instead. The primary vote manages both permissions, so no other vote is needed. Two calls.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 2 | Secondary agent 0x556e...3ce0, transfer to the Agent 0xBBE0...13DA of 59,419.71 USDC, and <code>execute</code> to send 0.057 ETH | Moves the secondary agent's USDC and ETH to the Agent. The few airdropped tokens with no market stay behind, nothing touches them. Balances read on 19 Oct 2026; the posted amounts are the balances at the block the bytes were built, right before posting, and can be checked with the repository's decoder. Two calls.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 3 | Primary agent 0xD9F8...04Ae, transfer to the Treasury 0xb7fa...AE4A of 6,083,873.53 API3                                         | Moves the treasury API3 held by the primary agent to the Treasury. The primary agent holds nothing else.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 4 | Api3Token 0x0b38...B88a, <code>updateMinterStatus(Api3Pool, false)</code>                                                        | Removes the staking pool's right to <code>mint</code> . From this call no rewards are paid. The pool checks this flag before it tries to <code>mint</code> , so staking, unstaking and withdrawals keep working.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 5 | Api3Token, <code>updateMinterStatus(primary agent, true)</code>                                                                  | Only a minter can call <code>mint</code> , and only the owner can name minters. The primary agent is the owner, so it names itself minter for the next call.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 6 | Api3Token, <code>mint(Treasury, 9,929,000 API3)</code>                                                                           | The final <code>mint</code> , straight to the Treasury. The amount was fixed when the vote bytes were built, right before posting: 200,000,000 minus the total supply at that block, shown here to the nearest token. The pool mints its weekly reward at the first interaction after Thursday 00:00 UTC, so the vote was posted on Thursday 29 October after that week's <code>mint</code> . If the pool mints again before execution, the DAO burns the excess over 200,000,000 from its multisigs right after execution, in one Safe transaction. The token contract lets a holder register as a burner and burn its own tokens. Total supply lands on exactly 200,000,000 either way, and no new vote is needed. A helper contract minting a dynamic amount was the alternative and isn't needed. |
| 7 | Api3Token, <code>updateMinterStatus(primary agent, false)</code>                                                                 | Removes the primary agent as minter again. No address can <code>mint</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

|    | CONTRACT AND CALL                                                                                                                                                                                                      | WHAT IT DOES                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8  | Api3Token, transferOwnership(0x...dEaD)                                                                                                                                                                                | Gives up ownership of the token contract to the dead address. The contract does not allow ownership to be renounced, and only the owner can change the minter list, so the empty list is permanent. The pool is the only address that has ever been given minter status, so after step 4 there is nothing else to remove.                                                                                                                                                   |
| 9  | ACL, revokePermission(any address, primary voting app 0xdb6C...170b, CREATE_VOTES_ROLE) and the same on the secondary voting app 0x1c80...6372                                                                         | Today any address may create a vote on either app. This removes that right, so no new vote can be created.                                                                                                                                                                                                                                                                                                                                                                  |
| 10 | ACL, removePermissionManager(both voting apps, CREATE_VOTES_ROLE)                                                                                                                                                      | The primary voting app manages the right to create votes on both apps and gives that up. Nobody can grant the right again.                                                                                                                                                                                                                                                                                                                                                  |
| 11 | Both voting apps, changeSupportRequiredPct and changeMinAcceptQuorumPct to the contract maximum, then ACL, removePermissionManager(both apps, MODIFY_SUPPORT_ROLE and MODIFY_QUORUM_ROLE)                              | The second lock. Support is raised first, since the app rejects a quorum above the support. Even a vote created by a route we have not seen would need a hair under 100% of all voting power to pass, and nobody can lower that again. Eight calls.                                                                                                                                                                                                                         |
| 12 | ACL, revokePermission(primary voting app, Kernel, APP_MANAGER_ROLE) then removePermissionManager on it, and last revokePermission(primary voting app, ACL, CREATE_PERMISSIONS_ROLE) then removePermissionManager on it | The root permissions. CREATE_PERMISSIONS_ROLE is what could create a permission again after rows 10 and 11 removed its manager, and APP_MANAGER_ROLE can swap the code behind any app. Both are held and managed by the primary voting app. Rows 9 and 10 already put them out of reach, since only a primary vote could use them; this row makes that hold even for a vote created by a route not seen, or one still open at execution. Four calls. The old DAO is closed. |

There are a few things the script can't prevent, which we know about and accept. A primary vote executes in the transaction that takes it past 50%, so nobody chooses the execution block. A vote created in the old DAO keeps the thresholds it was created with, so a primary vote created before execution and passed afterwards could still act through the primary agent, but it would need the same majority as this vote. We won't post any other vote that touches what this one moves while it runs, and if a secondary vote spends from the secondary agent before execution, this vote fails until the balance is restored. Anyone can register as a burner and burn their own API3 between posting and execution, which would leave the supply below 200,000,000, and nothing can mint it back. Tokens that arrive at either Aragon agent after the bytes were built aren't swept by this proposal, because the amounts are fixed then, and no new vote can move them after execution.

## B

## Proposal parameters

| PROPOSAL TYPE           | PRIMARY (MORE THAN 50% OF ALL VOTING POWER IN FAVOUR)                                                                                                                                                                                              |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Token                   | Api3Token 0x0b38210ea11411557c13457D4dA7dC6ea731B88a                                                                                                                                                                                               |
| Pool                    | Api3Pool 0x6dd655f10d4b9e242ae186d9050b68f725c76d76                                                                                                                                                                                                |
| Final mint              | 200,000,000 minus total supply at posting, minted directly to the Treasury                                                                                                                                                                         |
| Minters after execution | None. Token ownership transferred to 0x...dEaD.                                                                                                                                                                                                    |
| Vault                   | 0x8e03609ed680B237C7b6f8020472CA0687308b24 (3-of-5). Holds wstETH, the USDC positions and the old DAO's delegations                                                                                                                                |
| Treasury                | 0xb7fa2Cb302C5dC127e61F27b92beC411cbf0AE4A (3-of-5, same signers). New. Receives the primary agent's 6,083,874 API3 and the final mint, then the Vault's unearmarked API3. Voting power delegated to the dead address, never posts                 |
| Agent                   | 0xBBE0dE9757F93e3306aDBfeBE906AB285EDD13DA (3-of-5, same signers). Receives the secondary agent's USDC and ETH. Owns the curated Morpho vaults. Voting power delegated to the dead address                                                         |
| Aragon primary agent    | 0xD9F80Bdb37E6Bad114D747E60cE6d2aaF26704Ae, emptied to the Treasury                                                                                                                                                                                |
| Aragon secondary agent  | 0x556ecbb0311d350491ba0ec7e019c354d7723ce0, emptied to the Agent via permission grant                                                                                                                                                              |
| Old DAO                 | Right to create votes revoked on both voting apps and passing requirement raised to the contract maximum, both permissions left without a manager. Root permissions (CREATE_PERMISSIONS_ROLE, APP_MANAGER_ROLE) revoked and left without a manager |
| Governance              | Snapshot, one vote per API3 (wallet, staked, deposited, unstaking, collateral in Kabu API3/USDC Morpho market 0x70df...6c67a1)                                                                                                                     |
| Withdrawals             | Pool stays open indefinitely, DAO dashboard and tracker maintained until end of March 2028, Etherscan instructions after                                                                                                                           |

## C

## The vesting bypass, verified in the source

---

Every reward epoch is recorded once, in `RewardUtils.mintReward()`, as the block it was minted in, the amount, and the total pool shares at that moment. A user's locked amount is not stored anywhere. `GetterUtils.userLocked()` recomputes it on every withdrawal by walking the last 52 epochs and, for each reward, taking the user's most recent share checkpoint written at or before the reward's block and adding the user's proportional cut of that reward to the locked total. `TransferUtils.withdraw()` then refuses any withdrawal that would leave less than the locked amount in the pool.

`StakeUtils.scheduleUnstake()` calls `mintReward()` as its first step, inside the same transaction. If this is the first interaction with the pool after the Thursday 00:00 UTC epoch boundary, the weekly reward is minted right there, stamped with the current block. The function then values the user's stake against a total that already includes the fresh reward, so the amount scheduled for unstaking includes the user's cut of it, and writes a new share checkpoint for the reduced share count, stamped with the same block.

When `userLocked()` later reaches that epoch it looks for the latest checkpoint at or before the reward block. The post-unstake checkpoint carries the same block number, so it is the one that counts, and the user's share for that epoch is whatever was left after the unstake. For a full unstake that is zero, so that week's reward adds nothing to the locked total. Seven days later the unstake releases the scheduled amount, reward included, and the withdrawal check passes because the lock never counted it. Every earlier week's reward stays locked, because those checkpoints predate their reward blocks.

A user holding a share  $s$  of the pool who unstakes in full in the reward block receives  $s$  times that week's reward a year early. At 950,000 API3 a week, a user with 10% of the pool would get about 95,000 API3 unlocked instead of vesting. It can happen at most once a week, and only by giving up the stake and its vote for the seven-day wait. No one else's tokens are affected and no extra API3 is minted. We checked the last 3,000 unstakes, back to February 2024, against every reward block since the pool launched. Thirteen landed in a reward block, no address twice, most of them partial, and the reward they released early adds up to less than 1,000 API3 in total. As far as we can tell, nobody did it on purpose. The cause is a single comparison, at or before where it should be strictly before, in a function four audits read, and it's in the contract that holds the only right to mint API3. This proposal doesn't patch it. The pool stops minting and the contract is retired.

# References

---

- 1 API3 DAO tracker. [tracker.api3.org/](https://tracker.api3.org/)
- 2 API3 website, token and treasury. [api3.org/api3/](https://api3.org/api3/)
- 3 API3 DAO documentation, proposals and voting. [dao-docs.api3.org/members/proposals.html](https://dao-docs.api3.org/members/proposals.html)
- 4 API3 DAO documentation, reward mechanism. [dao-docs.api3.org/technical/distribution.html](https://dao-docs.api3.org/technical/distribution.html)
- 5 CoinGecko, API3. [www.coingecko.com/en/coins/api3](https://www.coingecko.com/en/coins/api3)
- 6 API3 cycle proposal, September to November 2026. [moccasin-traditional-bat-376.mypinata.cloud/ipfs/bafyb3cw2ues7p7naapzbeidd7as...](https://moccasin-traditional-bat-376.mypinata.cloud/ipfs/bafyb3cw2ues7p7naapzbeidd7as...)
- 7 Snapshot. [snapshot.box/](https://snapshot.box/)
- 8 Snapshot documentation, strategies and delegation. [docs.snapshot.box/](https://docs.snapshot.box/)
- 9 Snapshot Labs, sx-monorepo (MIT). [github.com/snapshot-labs/sx-monorepo](https://github.com/snapshot-labs/sx-monorepo)
- 10 API3 Discord. [discord.gg/api3dao](https://discord.gg/api3dao)
- 11 API3 DAO Treasury. [etherscan.io/address/0xb7fa2Cb302C5dC127e61F27b92beC411cbf0AE4A](https://etherscan.io/address/0xb7fa2Cb302C5dC127e61F27b92beC411cbf0AE4A)
- 12 API3 DAO dashboard. [api3.eth.limo/](https://api3.eth.limo/)
- 13 API3 DAO contracts and audit reports. [github.com/api3dao/api3-dao](https://github.com/api3dao/api3-dao)
- 14 Anthropic, AI agents find smart contract exploits, December 2025. [www.anthropic.com/research/smart-contracts](https://www.anthropic.com/research/smart-contracts)
- 15 The Hacker News, Claude Mythos finds thousands of zero-day flaws, April 2026. [thehackernews.com/2026/04/anthropics-claude-mythos-finds.html](https://thehackernews.com/2026/04/anthropics-claude-mythos-finds.html)
- 16 Blockhead, BonkDAO loses \$20M after attacker buys governance control, July 2026. [www.blockhead.co/2026/07/08/bonkdao-loses-20-million-after-attacker-buys-governance...](https://www.blockhead.co/2026/07/08/bonkdao-loses-20-million-after-attacker-buys-governance...)
- 17 The Block, Compound \$24M proposal, July 2024. [www.theblock.co/post/307943/24-million-compound-finance-proposal-passed-by-whale...](https://www.theblock.co/post/307943/24-million-compound-finance-proposal-passed-by-whale...)
- 18 The Defiant, Compound settlement, July 2024. [thedefiant.io/news/defi/compound-agrees-to-distribute-30-of-reserves-to-comp-sta...](https://thedefiant.io/news/defi/compound-agrees-to-distribute-30-of-reserves-to-comp-sta...)
- 19 Cointelegraph, Aragon dissolution, November 2023. [cointelegraph.com/news/aragon-association-dissolves-disburse-assets-token-holder...](https://cointelegraph.com/news/aragon-association-dissolves-disburse-assets-token-holder...)
- 20 The Block, Tornado Cash governance takeover, May 2023. [www.theblock.co/post/231637/attacker-uses-malicious-proposal-to-take-over-tornado...](https://www.theblock.co/post/231637/attacker-uses-malicious-proposal-to-take-over-tornado...)
- 21 CertiK, Beanstalk exploit, April 2022. [www.certik.com/resources/blog/revisiting-beanstalk-farms-exploit](https://www.certik.com/resources/blog/revisiting-beanstalk-farms-exploit)
- 22 Arbitrum forum, vote buying services, 2025. [forum.arbitrum.foundation/t/dao-discussion-vote-buying-services/28934](https://forum.arbitrum.foundation/t/dao-discussion-vote-buying-services/28934)
- 23 crypto.news, Binance stops DAO governance attack, August 2026. [crypto.news/binance-says-it-stopped-1-2m-dao-governance-attack/](https://crypto.news/binance-says-it-stopped-1-2m-dao-governance-attack/)
- 24 API3 documentation, OEV. [docs.api3.org/oev-searchers/](https://docs.api3.org/oev-searchers/)
- 25 Fortune, agents the majority of web traffic, July 2026. [fortune.com/2026/07/23/dead-internet-theory-bots-agents-majority-web-traffic/](https://fortune.com/2026/07/23/dead-internet-theory-bots-agents-majority-web-traffic/)
- 26 x402 protocol. [www.x402.org/](https://www.x402.org/)
- 27 Coinbase, x402 first year, May 2026. [www.coinbase.com/blog/introducing-amazon-bedrock-agentcore-payments-powered-by-x...](https://www.coinbase.com/blog/introducing-amazon-bedrock-agentcore-payments-powered-by-x...)
- 28 Digiday on TollBit, crawler to referral ratios, August 2026. [digiday.com/media/european-publishers-are-getting-hit-harder-by-ai-bot-scraping-...](https://digiday.com/media/european-publishers-are-getting-hit-harder-by-ai-bot-scraping-...)
- 29 API3, AirnodeHub launch post, August 2026. [x.com/Api3DAO/status/2087930138075754705](https://x.com/Api3DAO/status/2087930138075754705)
- 30 AirnodeHub. [airnodehub.api3.org/](https://airnodehub.api3.org/)
- 31 PR Newswire, Orthogonal \$4.3M seed, June 2026. [www.prnewswire.com/news-releases/orthogonal-raises-4-3m-seed-for-ai-agent-service...](https://www.prnewswire.com/news-releases/orthogonal-raises-4-3m-seed-for-ai-agent-service...)
- 32 API3, the vote source, verification script, fork tests and decoder. [github.com/api3dao/api3-dao-migration-proposal](https://github.com/api3dao/api3-dao-migration-proposal)
- 33 API3, Snapshot voting verifier. [github.com/api3dao/snapshot-voting-verifier](https://github.com/api3dao/snapshot-voting-verifier)

# 200,000,000

This document is the full text of the API3 DAO primary proposal "200,000,000 API3", version 1.0, dated 29 October 2026. It is pinned on IPFS, and its content hash is written into the description of the on-chain vote, so the text voted on can always be checked against this file.

The exact transaction the vote executes, the fork tests that replay it and a decoder that prints it as the table in Appendix A are published at [github.com/api3dao/api3-dao-migration-proposal](https://github.com/api3dao/api3-dao-migration-proposal). The tool the signers use to check Snapshot votes is at [github.com/api3dao/snapshot-voting-verifier](https://github.com/api3dao/snapshot-voting-verifier).

Set in Source Serif 4 and Inter Tight. Figures drawn from on-chain data, the Morpho API and the API3 DAO tracker.