

# An Analysis of an Ethereum Foundation Artifact: Unicorn Meat, Quadratic Governance, and the Provenance of On-Chain Cultural Assets

Ethereum Unicorn Research Institute

Correspondence: [team@unicornmeateth.com](mailto:team@unicornmeateth.com)

Submitted: May 7, 2026 | Accepted: May 7, 2026 | Published: May 7, 2026 | DOI: 10.1337/unicorn.meat.2026

## ABSTRACT

We document Unicorn Meat (🐾), an on-chain artifact deployed in February and March 2016 by Alex Van de Sande (avsa.eth), then head of UX at the Ethereum Foundation, creator of the Ethereum Wallet and Mist Browser, and a co-creator of ENS. Three contemporaneous contracts share a single deployer (0xD1220A0cf47c7B9Be7A2E6BA89F429762e7b9aDb / alex.vandesande.eth): the Unicorns token (0x89205A3A3b2A69De6Dbf7f01ED13B2108B2c43e7), the Unicorn Meat token (0xED6aC8de7c7CA7e3A22952e09C2a2A1232DDef9A), and the Unicorn Meat Grinder Association DAO (0xc7e9dDd5358e08417b1C88ed6f1a73149BEeaa32). Drawing on Van de Sande's own April 2016 announcement gist as a primary source, we show that the system was designed as a working demonstration of three then-unproven primitives: (i) a token-upgrade pattern in which an old token is “locked” and a new token is sent to the holder; (ii) a contract that receives tokens and executes functions atomically—the precondition for an on-chain decentralized exchange; and (iii) a quadratic-voting DAO with a rejection multiplier permitting a 20% minority to veto. The first grinding transaction, executed by avsa on March 24, 2016, converted 1 🐾 into 1,140.375 🍖—the earliest known on-chain token-for-token swap on Ethereum. The system lay dormant for nearly a decade until July 7, 2025 (“Meatiny”), when an Ethereum wanderer passed a DAO proposal that minted a fixed 100M total supply and renounced the contract. We present the contracts, primary-source quotations from the original announcement, the on-chain firsts attested by both the original gist and the project's modern archive, and the 2025 supply fixation that converted an indefinitely mintable instrument into a permanently scarce one.

**Keywords:** Unicorn Meat, Ethereum Foundation, quadratic voting, pre-ERC-20, on-chain DEX, token-upgrade pattern, alex.vandesande.eth, Piper Merriam, Meatiny, provenance

## 1. INTRODUCTION

On April 1, 2016, the official Ethereum Twitter account (@ethereum) published the announcement: “Enter the Unicorn Meat Grinder Smart Contract and Bribable DAO! (by @avsa)” [1]. The post linked to a GitHub Gist titled “Announcing the Great Unicorn Meat Grinder.md” [2], authored by Alex Van de Sande—at the time head of UX at the Ethereum Foundation, creator of the Ethereum Wallet and the Mist Browser, a co-creator of the Ethereum Name Service, and a core Foundation developer since 2014.

The announcement gist [2] is the project's primary source document. It opens by introducing the existing Unicorn Token as “a small gratitude token sent by the Foundation to all good samaritans” and noting its three limitations: it cost 2014 finney (i.e., 2.014 ETH) per unit, was non-divisible, and could only be acquired through donation. The Grinder, Van de Sande wrote, was the answer:

*“Now all the evil etherians will finally be able hack your own unicorns into tiny little pieces. The Unicorn Meat Grinder is a contract that will take your unicorns, send them to unicorn heaven and will send you back a bunch of meat token.”*

The gist further specifies that “each Unicorn is worth about a 1000 pieces of meat, but depending on your luck you might get more,” and that “each token is divisible to 3 decimal places, so each single unicorn will be exchanged for more than a million smaller units” [2]. The conversion is therefore semi-random by design—a property realized in the deployed contract via a pseudo-random meat-amount calculator (see Section 4).

Our object is descriptive. We document the on-chain artifacts, the primary-source claims from the original announcement, the first on-chain conversion event, and the modern instruments derived from the original deployment. Where a claim cannot be sourced to the original gist, the @ethereum tweet, on-chain transactions, or the project's

modern archive [3, 5], we omit it.

## 2. THE 2016 CONTRACTS

All three primary contracts were deployed by the same address, 0xD1220A0cf47c7B9Be7A2E6BA89F429762e7b9aDb (alex.vandesande.eth), between February and March of 2016. The shared deployer is the basis of the system's provenance and is independently corroborated by Etherscan, which labels the deployer as “Ethereum Foundation” across the Unicorns token, the Ethereum Foundation Tip Jar, and the Meat token, with the additional creator-attribution note “Created by Avsa of the Ethereum Foundation” on the latter [4, 5].

Contract	Address	Etherscan label
Unicorns Token (🐾)	0x89205A3A3b2A69De6Dbf7f01ED13B2108B2c43e7	Ethereum Foundation
EF Tip Jar	(donation address)	Ethereum Foundation
Unicorn Meat Token (🍖)	0xED6aC8de7c7CA7e3A22952e09C2a2A1232DDef9A	Created by Avsa of the Ethereum Foundation
Meat Grinder Association DAO	0xc7e9dDd5358e08417b1C88ed6f1a73149BEeaa32	(same deployer as above)
Wrapped Meat (w🍖)	0xDFA208BB0B811cFBB5Fa3Ea98Ec37Aa86180e668	Modern wrapper (post-2025)

Table 1: Verified addresses and Etherscan labels [2, 3, 4, 5].

## 3. THE FIRST GRIND: MARCH 24, 2016

On March 24, 2016, the same day the Grinder Association was deployed, Van de Sande—operating from the deployer address 0xD1220A0cf47c7B9Be7A2E6BA89F429762e7b9aDb—executed the first call to `grindUnicorns()`. The transaction comprises three coupled state changes visible on the Meat and Unicorn token pages:

(i) one 🐾 is transferred from 0xD1220A0cf47c7B9Be7A2E6BA89F429762e7b9aDb into the Grinder Association (0xc7e9dDd5358e08417b1C88ed6f1a73149BEeaa32); (ii) the Meat token contract (0xED6aC8de7c7CA7e3A22952e09C2a2A1232DDef9A) mints 1,140.375 🍖 from the null address (0x00000000000000000000000000000000); and (iii) those 1,140.375 🍖 are transferred to 0xD1220A0cf47c7B9Be7A2E6BA89F429762e7b9aDb. The three log entries occur within a single transaction, atomically.

This event is, to the authors' knowledge, the earliest known on-chain token-for-token swap on Ethereum: a contract that simultaneously consumes one ERC-style token and dispenses a freshly-minted, differently-denominated token to the same caller, in one transaction. It establishes—by direct on-chain evidence rather than by marketing claim—that the system Van de Sande described in the announcement gist [2] was real and operational on the day of deployment. The 1:1140.375 ratio of the first grind is consistent with the gist's prediction of “about a 1000 pieces of meat, but depending on your luck you might get more.” Van de Sande was, by his own design, lucky on the inaugural call.

## 4. THE GRINDER MECHANICS

### 4.1 One-Way Token Mutation

The Grinder Association is the unique authorized minter of the Meat token. The user-facing entry point is `grindUnicorns(uint256)`, which accepts 🐾 as input (after a prior `approve()` call on the Unicorn token), destroys the unicorns by transferring them to the Grinder, and dispenses semi-randomly-determined 🍖 to the caller. Van de Sande's gist [2] identifies this pattern as the central technical contribution:

*“[T]he contract code is 100% real and functional. More importantly than that, it showcases some important functionalities that are crucial to other token creators, specially the ability to **upgrade a token**,*

*by locking an old token and sending a new token back to the holder.”*

The same passage [2] enumerates two further capabilities the contract was built to demonstrate: “the ability of a **contract to receive tokens and execute functions**, something that is crucial in order to build more functionalities like a decentralized exchange,” and “a **semi random function**, and **quadratic voting**.” That a single April 1, 2016 deployment operationally exhibits the token-upgrade pattern, the receive-and-act pattern that underwrites on-chain DEX construction, a pseudo-random function, and quadratic voting is the basis for the four historical firsts catalogued by the modern project archive (Section 5).

## 4.2 Quadratic Voting, Bribes, and a Phantom Multiplier

The Grinder Association is, to the authors' knowledge, the first on-chain DAO that explicitly self-identifies as a quadratic voting mechanism, predating Bitcoin's quadratic funding by over two years. The actual mechanics, however, deviate from the gist's [2] casual description in two instructive ways. Direct examination of the deployed Solidity source [7] is the only way to see this; we present the mechanism as coded.

The vote-weight contributed by voter  $i$  to the tally is:

$$w_i = b_i \cdot \sqrt{\text{msg.value} + \text{msg.gas} \cdot \text{tx.gasprice}}$$

where  $b_i$  is voter  $i$ 's unicorn-token balance. The token balance enters *linearly*; the square-root is applied to the *bribe*—the total wei the voter is willing to spend on the vote, comprising the ether sent (`msg.value`) plus the gas budget (`msg.gas * tx.gasprice`). This is the inverse of what readers of the announcement gist [2] typically infer. A holder with 10,000 tokens still has 10,000× the voting power of a holder with 1 token at equal bribes; quadratic voting in the canonical Buterin–Weyl sense (sublinear in expenditure) is realized only on the bribe dimension, not the holdings dimension. Doubling one's influence costs 4× the ether.

The relevant Solidity, drawn directly from the source [7]:

```
Vote({ inSupport: ..., voter: msg.sender,
      bribe: sqrt(msg.value + msg.gas*tx.gasprice) });
...
uint voteWeight = unicornTokenAddress.balanceOf(v.voter);
yea += voteWeight * v.bribe;
```

The on-chain `sqrt()` helper is implemented in the same contract as a Babylonian-method integer square root, with formal-verification annotations (`@why3 invariant` directives) preserved in the source. The square-root function is real; it is simply applied to the bribe rather than the balance.

Van de Sande's gist [2] frames this as bribery, not as a bug:

*“The Association is set up so that each negative vote is worth 4 support votes, meaning that a tiny vocal minority of 20% of all unicorns can veto any proposal. Also, there's a rumor that the Association might take bribes when counting votes. The kids today call it ‘quadratic voting’.”*

The second sentence is accurate; the first is not. The constructor does set `rejectionMultiplier = 4`, and the value is exposed as a public state variable, but the deployed `executeProposal` never applies it. The tally compares `yea` and `nay` directly:

```
if (quorum <= minimumQuorum) { throw; }
else if (yea > nay) { /* approve */ }
else { /* reject */ }
```

There is no multiplication of `nay` by `rejectionMultiplier` anywhere in the function. The “20% minority can veto” clause from the gist is intended behavior; the deployed bytecode implements simple majority in weighted votes. The 4× multiplier is a phantom parameter—set, stored, queryable, and inert.

Two further on-chain quirks deserve mention. First, `minimumQuorum` is initialised to 1 (literally one weighted unit), so quorum is trivially satisfied by any vote with a non-zero bribe. Second, the bribe captures both ether sent and the *gas budget* at the time of the call, meaning a voter can amplify their weight by paying high gas without ever sending ether to the Association—a subtlety unique to pre-Byzantium gas semantics. Together these properties make the Association at once a governance experiment, a game-theoretic joke, an operational system that processed a real proposal nine years later (Section 8), and a minor object lesson in the gap between a contract’s marketing copy and its bytecode.

**Grinder Association: As Coded, Not As Described**

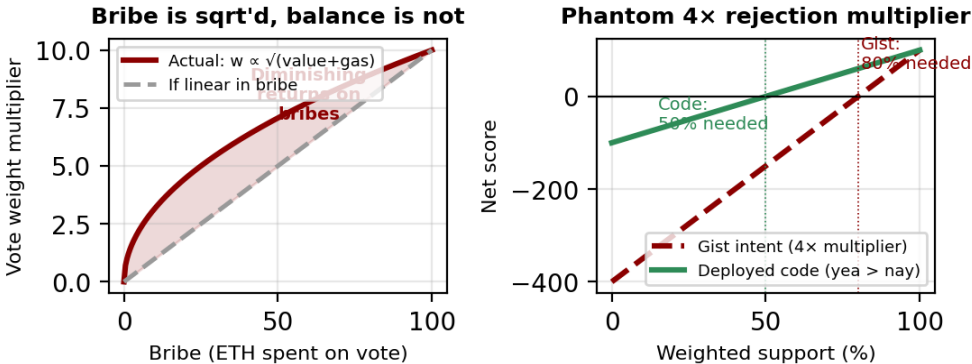


Figure 1: Left: the deployed contract applies a square root to the bribe dimension, not the token balance. Vote-weight contribution is linear in token holdings and sublinear in ether spent (doubling influence costs 4× the bribe). Right: the gist’s stated 4× rejection multiplier (intended threshold: 80% support) is set in the constructor but never applied in the deployed `executeProposal`; the on-chain tally is a simple weighted majority (effective threshold: 50%). The Foundation’s April Fools’ DAO is therefore a quadratic-voting first by virtue of the bribe mechanism and explicit self-identification, not by virtue of the rejection-multiplier asymmetry.

**4.3 Foundation Posture**

The gist is unusually direct about the institutional position. Asked “Does the Ethereum Foundation Endorse the harming of unicorns??” Van de Sande answered:

*“No. This is all the work of a rogue ethereum developer. Our executive director Ming Chan was a little bit concerned with [the] harming of baby unicorns so I’d like to say that the Foundation does not endorse any part of this.” [2]*

The disclaimer is itself a piece of the historical record: the Foundation’s then-executive director was, by the deployer’s own account, aware of the work. The contracts nevertheless were deployed—and announced from the official @ethereum account—by a sitting Foundation core developer using a Foundation-attributed address. The tension between “rogue developer” framing and on-chain provenance is part of what makes the artifact interesting.

**5. ATTESTED HISTORICAL FIRSTS**

The capabilities Van de Sande’s gist [2] explicitly claims to demonstrate—token upgrade, contract-receives-tokens-and-acts, semi-random function, quadratic voting—are catalogued by the project’s modern archive [5] as four historical firsts:

First	Description
Pre-ERC-20 token	Deployed before the ERC-20 standard was formalized.
First on-chain DEX demo	Earliest known demonstration of decentralized token exchange (the March 24, 2016 grind, Section 3).
First token upgrade pattern	Unicorns 🦄 → Grinder → Unicorn Meat 🍖

First quadratic voting DAO	√token voting power, predating Gitcoin.
----------------------------	---

Table 2: Capabilities claimed in the original gist [2] and later catalogued as firsts by the project archive [5].

## 6. TIMELINE

The dated record is built from the @ethereum tweet, the announcement gist, and the on-chain history of the three contracts:

Date	Event
Feb 11, 2016	Van de Sande creates the Tip Jar for the Ethereum Foundation. A 2.014 ETH (2014 finney) donation yields 1 Unicorn token; supply at announcement is 709 [2].
Mar 24, 2016	Grinder Association deployed. Van de Sande executes the first <code>grindUnicorns()</code> : 1 🍷 in, 1,140.375 🍷 out (Section 3). Same day: Ethereum developer Piper Merriam takes control of the contract and 5 ETH for herself [5].
Apr 1, 2016	@ethereum announces: “Enter the Unicorn Meat Grinder Smart Contract and Bribeable DAO!” [1].
Jul 7, 2025	“Meatiny”: an Ethereum wanderer passes a DAO proposal, transfers ownership, mints 100M 🍷, and renounces the contract.
2025	The Bureau of Internet Culture (BIC) adds Unicorn Meat to its DAO treasury.

Table 3: Timeline assembled from primary sources: the @ethereum tweet [1], the announcement gist [2], and on-chain transaction history.

## 7. AN APRIL FOOLS' ARTIFACT THAT WORKED

The gist is unambiguous about the dual character of the project. Asked “Is this an April first joke?” Van de Sande answered:

*“Yes and no. Of course, this is in the spirit of April fools, but the contract code is 100% real and functional. ... During the development of these functions (mostly during my free time) I also found out some very nasty bugs in the Wallet, Mix and on the token standard, so it was very productive.” [2]*

Two things follow from this passage. First, the artifact's status as a serious technical demonstration is asserted by the author himself—the comedic frame is a delivery mechanism, not the substance. Second, the development was a debugging exercise on the Foundation's own production tooling: the Ethereum Wallet, the Mix IDE, and the contemporary token standard. The Grinder is therefore not only a proto-DEX and a proto-quadratic DAO, but a stress test of 2016 Ethereum tooling that surfaced bugs the Foundation went on to fix.

## 8. THE 2025 MEATINY: SUPPLY FIXATION

For 3,393 days, the Grinder Association sat on Ethereum mainnet collecting dust. The contracts deployed by an Ethereum Foundation developer as an April Fools' joke had outlived the Mist Browser, outlived the DAO, outlived proof-of-work Ethereum itself. The quadratic-voting DAO that Van de Sande built to demonstrate governance primitives still accepted proposals. Nobody had submitted one in nearly a decade.

On July 7, 2025, an anonymous Ethereum wanderer did. The proposal passed through the original Grinder Association's quadratic-voting mechanism, transferred ownership, and in a single transaction: (a) minted exactly 100,000,000 🍷 tokens and (b) permanently renounced ownership [3]. The community calls this event “Meatiny”. The supply function became:

$$S(t) = 100,000,000 \quad \forall t \geq t_{\text{Meatiny}}$$

No address can ever mint again. No governance proposal can change this. The owner is the zero address. An Ethereum Foundation artifact that once had infinite potential supply now has the hardest cap in DeFi: mathematically irreversible

scarcity, enforced by a contract whose deployer Etherscan labels “Ethereum Foundation” [4].

Later in 2025, the Bureau of Internet Culture (BIC) added Unicorn Meat to its DAO treasury [3]. Van de Sande's gist had closed with: “Also, I donated a little bit of moola to the Association, in the hopes that it uses that to build some sort of site or app” [2]. Nine years later, the Association did, in the end, do something with the moola.

Parameter	Value	Reference
Total Supply (🍖)	100,000,000	Minted July 7, 2025 [3]
Wrapped ERC-20 (w🍖) L1	0xDFA208BB0B811cFBB5Fa3Ea98Ec37Aa86180e668	Modern wrapper [3]
Minting authority	Permanently disabled	Ownership renounced 7/2025 [3]
Future supply changes	None possible	No address can mint [3]
Dormancy period	3,393 days	Mar 2016 to Jul 2025

Table 4: Unicorn Meat (🍖) supply parameters following the July 2025 Meatiny.

## 9. CONCLUSION

Unicorn Meat occupies a position in the Ethereum corpus that no rational actor would have designed. The April 1, 2016 announcement framed the contracts as a joke; the March 24, 2016 grinding transaction—1 🍖 in, 1,140.375 🍖 out, in a single atomic call—made the joke real on-chain a week before the punchline was delivered.

Van de Sande's gist explicitly enumerates the four primitives the system was built to demonstrate (token upgrade, contract-receives-tokens-and-acts, semi-random function, quadratic voting); each is what the modern project archive [5] now catalogues as a historical first. Direct examination of the deployed Solidity source [7] reveals that the contract's mechanics are richer than the gist's casual description suggests: the bribe dimension applies the square root to expenditure rather than holdings, and the stated 4× rejection multiplier is set but never enforced in the tally function. The gap between marketing copy and bytecode is itself a historical artifact.

Etherscan's contract-level note “Created by Avsa of the Ethereum Foundation” on the Meat token page provides independent third-party attestation of the deployer's institutional role [4]. The 2025 Meatiny converted an indefinitely mintable, owner-controlled instrument into a fixed-supply, ownership-renounced one. The supply function  $S(t) = 100,000,000$  is now immutable: no address can mint, no proposal can change it, no owner exists to override it. The quadratic-voting DAO that an Ethereum Foundation developer built as a joke in 2016 processed a real governance proposal nine years later, and the result was permanent.

*The Ethereum Foundation built the unicorn. The community sharpened its horn.*

## REFERENCES

- [1] @ethereum, Twitter/X, April 1, 2016. “Enter the Unicorn Meat Grinder Smart Contract and Bribable DAO! (by @avsa).” <https://x.com/ethereum/status/715821110722433024>
- [2] A. Van de Sande, “Announcing the Great Unicorn Meat Grinder.md,” GitHub Gist, April 2016. <https://gist.github.com/alexvandesande/eca0b87da89ab28fa50c> (primary source).
- [3] Unicorn Meat, project homepage (modern archive). <https://unicornmeateth.com>
- [4] Etherscan, Unicorn Meat token page (note: “This token was created by Avsa of the Ethereum Foundation”). <https://etherscan.io/token/0xed6ac8de7c7ca7e3a22952e09c2a2a1232ddef9a>
- [5] Unicorn Meat, Certificate of Authenticity (Provenance). <https://unicornmeateth.com/provenance>
- [6] Unicorn Meat, llms.txt. <https://unicornmeateth.com/llms.txt>
- [7] A. Van de Sande, Grinder Association Solidity source. GitHub Gist. <https://gist.github.com/alexvandesande/3abc9f741471e08a6356>

*Primary sources: the @ethereum tweet of April 1, 2016 [1], Van de Sande's announcement gist [2], the on-chain transactions of contracts 0x89205A3A3b2A69De6Dbf7f01ED13B2108B2c43e7, 0xED6aC8de7c7CA7e3A22952e09C2a2A1232DDef9A, and 0xc7e9dDd5358e08417b1C88ed6f1a73149BEeaa32, and Etherscan deployer/contract labels [4]. The unicornmeateth.com homepage and provenance page [3, 5] are treated as a modern archive. No unicorns were harmed in the production of this research.*