finance.vote

Christopher Smith, Nick Almond, Naved Ali

Aug 24, 2022

FINANCE.VOTE:

finance.vote

finance.vote is a decentralised autonomous organisation (DAO) that builds software (smart contracts and browser interfaces). This software enables users to easily generate a token economy and govern themselves as a DAO. Members of finance.vote use this software and the utility token FVT (finance.vote Token) to operate and govern finance.vote.

We believe in Truth, Knowledge, and Imagination and will seek democratic outcomes to becoming an important digital creative institution in the future.

CHAPTER

ONE

OUR PRODUCTS

The finance.vote dApp suite comprises decentralised applications that allow Web3.0 projects to create a token economy and digital democracy.

- launch
- bank
- influence
- markets
- mint
- yield

1.1 Contract addresses

- **uniswap** = ‘0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D’
- **safeAddress** = ‘0xa874Fa6ccDcCB57d9397247e088575C4EF34EC66’

1.1.1 Testnets

Kovan

- **kovanSafeMathLib** = ‘0x4AD25394335A46Cf657Bb34BD9165c535b1C0590’
- **kovanFVT** = ‘0x3f3e787cA8D404eD5CC761F29f7B503809a01370’
- **kovanIdentity** = ‘0x7Bd19AD922DfB7d9BB077b8c7B4C6a8833BA5047’
- **kovanTrollbox** = ‘0x0dF03114adE43107bfB3a20B96b4FDaFeC529bb6’
- **kovanChainlinkOracle** = ‘0xcA94eAf09dE1cbC854C7254dc7f64CdABc6cB611’
- **kovanNFTAuction** = ‘0xA076aac24913642CF2a807A10a17aA33FE452f79’
- **kovanNFTAuction2** = ‘0x2EC9792ddDea944b0396A665fA95f5293c265F0d’

Ropsten

- **ropstenSafeMath** = ‘0x2365A147D3fd5522e375e1c17eeC780095b73d49’
- **ropstenFVT** = ‘0xF7eF90B602F1332d0c12cd8Da6cd25130c768929’
- **ropstenUniswapLP** = ‘0x5615187acc2d65f078c36e07f06ed85225a5dd9d’
- **ropstenSushiswapLP** = ‘0xa4621ddf62ded064307e199c70a119b47e020239’
- **ropstenFVTVault** = ‘0x9D5f22b3299bDB612e36030e4e1afe854b3B11E3’
- **ropstenLPVault** = ‘0x96FFB474BCFE44235082c07C7f3BE07cb9fbC08f’
- **ropstenIdentity** = ‘0xD5E00f2A37A7741332f64bC407EbAB764C4c69C1’
- **ropstenTrollbox** = ‘0xF21692A51777C8aa6c8006f2527a9dDFF9A0C6d1’
- **ropstenAuction** = ‘0x46C455ED65b110eb24a970EB2A3D0625F94d306a’
- **ropstenAuction2** = ‘0x0B14C80668085d0E540F184CB1B3377aA3B898Dd’
- **ropstenLiquidity** = ‘0x6F2A3009D3d09456A894d1f8886F7a6B8F377d7a’
- **ropstenLiquidity2** = ‘0x72A1145A07311Df6D1ddBBF6A07128f0464E286F’
- **ropstenLiquidity3** = ‘0x60597b58F10Bb44580B171f4681449508Bf8BdAD’
- **ropstenLiquidity4** = ‘0xDE9ea2C339a8292823d151A60cfEEd8239a24A80’
- **ropstenLiquidity5** = ‘0xe244A7135FF265Cb854034A5edFB2cf359790266’
- **ropstenLiquidityFactory** = ‘0x1050f723DA941EAEf617ac25F216d74c06f572aa’
- **ropstenLiquidityFactory2** = ‘0xb5A30CedFC1c166BbB2dabC8c62B1C440Ff4BEb0’
- **ropstenMostBasicYield** = ‘0x82D3E88cd6cAFb934Fc27056D5BA48Ae1F240aB0’
- **ropstenMostBasicYield2** = ‘0x2C26dc46eA7D84B965a9931A2476d038e5155c3D’
- **ropstenMostBasicYield3** = ‘0x002ED20A5b3A37FA7D85d512802F701B735C43a6’
- **ropstenMostBasicYield4** = ‘0xa7f4B7A745ab539502332fD0807FB597cb417B8F’
- **ropstenMostBasicYield5** = ‘0x0b26BafAe1D9cf9590Faf35082711f6B8961c1bb’
- **ropstenBasicPoolFactory** = ‘0x148e2B2571EfC9dbb079F04Fe7c841Bd9eb2b49f’
- **ropstenVotingIdentity2** = ‘0xfA470DA77160C3D93F8e6413B4932b2ea97Bd05b’
- **ropstenNFTAuctionFactory** = ‘0x709393fa3Be880c9A4Cc311eEf37aF41Cf1778C8’
- **ropstenNFTAuctionFactory2** = ‘0x585734D7D85424F2fA3bBba8F063484220c74690’
- **ropstenLiquidityFactory** = ‘0xDd7eA6482eC99008CEd9fB103941DD29cfb31065’

Rinkeby

- **rinkebySafeMathLib** = ‘0x0cd13D6731F9b3f02DC31250b537cC2284800402’
- **rinkebyToken** = ‘0x1999b48df6Da3fa7810b3cE3fCE3b1da4E819888’
- **rinkebyUniswapLP** = ‘0x0681e248a813ee6144c21fddc05924b2baaf942c’

1.1.2 Mainnet

Libraries

- **mainnetSafeMath** = ‘0x82d7630c5EB722557De6D76575C9a7b8DE718500’
- **mainnetMerkleLib** = ‘0x8196D6264BB667908dC106D855Bf53E03816e725’

mint.vote

- mainnetIncinerator = ‘0x8C3D1656F22Ce2217C6A87200fa29f01CC3CA5A2’ <— deprecated
- mainnetIncinerator = ‘0x83aDE5216489E4768B80227C4608C5B12179dE4d’ <– deprecated
- mainnetFixedPriceGate = ‘0xff880DDa00485E12F5733E46862fD7b391eB813F’ <— deprecated
- mainnetFixedPriceGate = ‘0xc94CD0479d5e57BaE2Bed04Bc816BFFb06e626E8’ <– deprecated
- mainnetGatedMerkleIdentity = ‘0xfb0E8ff1deB51a76E4a623A4f839Aa3a5Ce786Be’ <— deprecated
- mainnetGatedMerkleIdentity = ‘0xb83B063838Cdb56F25D778D771390cdFd307938D’ <— deprecated
- mainnetGatedMerkleIdentity = ‘0x1f531048263d8E68c3bDE21a410bF8F0B65d414d’ <– deprecated
- mainnetVotingIdentity2 = ‘0x33aBDF51C51173382818ba719Cb1886eC40e160c’ <— deprecated
- mainnetVotingIdentity2 = ‘0xD5B81D548028710fE982E069A995e6C7500CD5E7’ <– deprecated
- mainnetVotingIdentity2 = ‘0x2eB92337f831b3F0A00F66c98622bC713F71613d’ <– deprecated

bank.vote

- mainnetTeamVault = ‘0x3e9109e38eDfEdFdCd5350672C287E2C672C9E2D’ <– primary
- **mainnetInvestorVault** = ‘0xB29B62e1Bf13f2F41960fEd1853032849b449673’
- **mainnetAdvisorVault1** = ‘0x40dEF72cE904E90684f5E4677c354F1E676d3373’

influence.vote

- mainnetIdentity = ‘0xf779cae120093807985d5F2e7DBB21d69be6b963’ <– primary
- **mainnetLondon2021Identity** = ‘0x6b30DfAAc56De970d5040A8727A8d26F98020447’

auction.vote

- mainnetAuction = ‘0x7A4bb7c12354780822d4Ed23114274be4E4C8E83’ <– primary
- **mainnetFVT** = ‘0x45080a6531d671DDFf20DB42f93792a489685e32’
- mainnetLPVault = ‘0xC7900783578E026645A6FCFDB7aa26adb63160E2’ <– empty
- **mainnetLPUniswap** = ‘0x75001b3ffe0f77864c7dc64c55e1e22b205e4a07’
- **mainnetLPSushiswap** = ‘0x96335e7cdbcd91fb33c26991b00cc13a87a811b9’

markets.vote

- mainnetTrollbox = ‘0xEa6556E350cD0C61452a26aB34E69EBf6f1808BA’ <– primary
- **mainnetTrollboxProxy** = ‘0xF9D234773ae2cE14277A9026fF6DA340669FdE4e’
- **mainnetReferralProgram** = ‘0x798A709E12CcA28bFa7Ff4a6dAa044b5e0B5FA00’
- **mainnetMKRAggregator** = ‘0x63A9ba6fdCcb08992c4886b8162468ABF2C240eF’
- **mainnetMKRAggregator2** = ‘0x0309B42c1DC9Ed9d95c06Aa32646Eea1Ca232d95’
- **mainnetUMAAggregator** = ‘0x4BC84c91f5Fe1052Da563bB327B639ebD4469d63’
- **mainnetUMAAggregator2** = ‘0x8e674727e7e53bdAbBC9C0Dc844a36a2163e6c6A’
- mainnetChainlinkOracle = ‘0xc9EE4C2e16E9BaA0A10031E082EB3D7aFd94E75e’ <– deprecated
- **mainnetChainlinkOracle2** = ‘0xa58d366f2078900E76Bea7f3dBb64766BB9614f4’
- **mainnetKeeperRegistry** = ‘0x5C8b4D52683758CF855Fa2118Ef0104FdCD63698’
- **mainnetKeeperRegistry2** = ‘0x109A81F1E0A35D4c1D0cae8aCc6597cd54b47Bc6’

bridge.vote

- mainnetChainBridge = ‘0xe3Dd2b16D9Db5d59Ea17c8e8c0A3e11e6C97b248’ <– primary
- **mainnetBridgeERC20Handler** = ‘0x42876633c355C8043d0872BCD85c73325dE08C13’

yield.vote

- **mainnetMostBasicYield** = ‘0x38A1f049b61024316969a1559F4F093391b1dEF6’
- mainnetLiquidityMining = ‘0x3Ff8338CA0fEEB4c950D78f5A5C00bdF104078CE’ <– deprecated
- mainnetLiquidityMining = ‘0x4F66f085727C3d8b72FFD2Efd0Bb93469648f945’ <– deprecated
- mainnetLiquidityFactory = ‘0xF94E4ee9a0ACA2193c0E9d38F60202382eAfBF4F’ <– deprecated
- mainnetLiquidityFactory = ‘0xfe64d9A8Fd6565b6842d6574871D929809424280’ <– primary
- **mainnetBasicPoolFactory** = ‘0x26D853B680e28d33e3Df80C566027452bB32e9B3’

partners

- **mainnetITrustMostBasicYield** = ‘0xe15a31e865B7C5769f69493AEE08aF2FF27CE39d’

1.1.3 Binance Smart Chain

libraries

- **bscSafeMathLib** = ‘0x86b43bf8bA2B61eA3c3F5a3C4c07517077fA043D’

bridge.vote

- **bscChainBridge** = ‘0xB8f5496Ca6A013622Ab909E7fE7d22b7Bc442213’
- **bscBridgeERC20Handler** = ‘0x785CFC6C2afcB058E8Dd6DDA49537C5a819D3625’
- **bscBridgeRelayer** = ‘0x2E70f28C39c8d1f1b737cD37e1EC76b72bD87cf6’

markets.vote

- **bscIdentity** = ‘0x951AED5E3554332BC2624D988c9c70d002D3Dba0’
- **bscTrollbox** = ‘0xA6Bb733a61f2f4F09090781CCCD80929c9234b3f’
- **bscTrollboxProxy** = ‘0xaDd34aCdbFc4733e5a448010F5cc809FF99e17aD’
- **bscReferralProgram** = ‘0x82d7630c5EB722557De6D76575C9a7b8DE718500’
- **bscChainlinkOracle2** = ‘0xf779cae120093807985d5F2e7DBB21d69be6b963’

infrastructure

- bscFVT = ‘0xc669E7fc84Fee4017782E67BcC45e5D7F65566BD’ <– deprecated
- bscFVT2 = ‘0x447047D2AC13ADD7BbA443f177A576852dD08Eb8’ <– deprecated
- bscFVT3 = ‘0x6E9630a7388E57D4457e5750388EB665a36B9cfA’ <– deprecated
- **bscFVT4** = ‘0x0A232cb2005Bda62D3DE7Ab5DEb3ffe4c456165a’
- bscLP = ‘0xfe2193a3b487847e107426927a1f6fc32b38dbae’ <– deprecated
- bscLP2 = ‘0x33a23558eb1ef3139dcf3a0484ad59c6e9755beb’ <– deprecated
- **bscLP3** = ‘0x668765Fd5Ce2F9f66a27BBBCa76c129aA8C45D90’

1.1.4 Polygon

- **maticSafeMathLib** = ‘0x86b43bf8bA2B61eA3c3F5a3C4c07517077fA043D’
- **maticChainBridge** = ‘0xB8f5496Ca6A013622Ab909E7fE7d22b7Bc442213’
- **maticFVT** = ‘0x72a5a58f79FFc2102227B92fAeBA93B169a3A3F1’
- **maticBridgeERC20Handler** = ‘0x785CFC6C2afcB058E8Dd6DDA49537C5a819D3625’
- **maticBridgeRelayer** = ‘0x2E70f28C39c8d1f1b737cD37e1EC76b72bD87cf6’
- **maticLP** = ‘0xBdeD75E911418ae4ECB117724B4b88458e3De88b’

1.2 Market Links

1.2.1 ETH

Sushi swap

Uniswap

1.2.2 BSC

PancakeSwap

1.2.3 Polygon

Quickswap

1.3 Finance.Vote Token

Finance.Vote Token is the ERC20 utility token used in the dApp Suite. A name of it voted for by the DAO prior to issuance (symbol: FVT).

The three design pillars of the FVT ecosystem are:

- **Prediction and Market Discovery** - DAOs handle, issue, and utilise digital assets. A core component of doing this effectively is to build good knowledge of the price of assets. The FVT ecosystem facilitates social consensus around the expected future value of established and upcoming digital assets.
- **Second Layer Governance** - finance.vote offers a suite of governance tools that can provide a voice to all token holders on any network. The second layer governance tools create a route from rough consensus and dialogue to high-stakes on-chain governance decisions.
- **Decentralised Social Trading** - The finance.vote social trading system will provide groups of any size with the ability to share market information, pool assets, and make collectivised trading decisions.

DAO members use FVT token in each of the applications, specified below:

- **launch.vote** - For any digital assets sold on launch.vote, a portion of the purchase amount is used to purchase FVT off the market. This FVT is then burned. (e.g. 1,000 ETH is spent in the auction to purchase TOKEN-A and 5% of the ETH spent (50 ETH) is converted to FVT and removed from supply).
- **bank.vote**- Projects wishing to use the vesting schedule system may be required to pay or burn FVT in order to do so. The majority of FVT emitted during the token generation event were subjected to a vesting schedule. Additionally, future investors, employees, and contractors may receive tokens on vesting schedules.
- **influence.vote** - Token holders can use FVT or an identity from markets.vote to "influence" proposals concerning the DAO through token weighted voting or other reputation mechanics.
- **markets.vote** - Access to the markets.vote dApp is gated by a decentralised identity token (an NFT) purchased with FVT (that is subsequently burned) from an auction system. Holding a decentralized identity token entitles users to a vote in a weekly set of prediction markets. Identity holders that make correct predictions share a reward pool of FVT.
- **yield.vote** - Users can use their FVT to earn token rewards in various liquidity pools. Projects wishing to create their own pool must burn a certain amount of FVT to create a partner pool.

1.3.1 Additional Revenue Streams

finance.vote has and is considering additional revenue models for the dApp suite as the DAO evolves and interacts with market forces. Assets go to the finance.vote treasury to be used by DAO members. Some of these are listed below.

- **launch.vote** - A percent of the liquidity generated by sale of any fungible tokens may be collected as liquidity provider (LP) tokens and stored in the finance.vote treasury (e.g. if 50,000 LP tokens of liquidity are created to create a market in a decentralised exchange, then 5% of these LP tokens (2,500 LP tokens) are sent to the treasury). Sellers may also be required to pay or burn ETH or FVT in order to submit assets to be sold at the auction.
- **bank.vote**- A small percentage of tokens vested by other projects are sent to the finance.vote treasury (e.g. TOKEN-A vests 1 million tokens in bank.vote and a 0.2% (2,000 tokens) vested allocation is attributed to the finance.vote treasury).
- **influence.vote**- Projects wishing to set up their own voting identity tokens may be required to pay or burn ETH or FVT to do so. Additionally, their users may be required to burn FVT to claim their voting identities.
- **markets.vote** - Projects wishing to set up their own prediction markets or identity systems may be required to pay or burn ETH or FVT to do so.
- **yield.vote**- A small portion of the LP tokens staked in some liquidity pools is sent to the finance.vote treasury.

1.3.2 Governance and Decentralisation

The FVT token began with *Day One Utility*. Token holders were able to gain access to the markets.vote application. The markets.vote application is deployed on decentralised systems, so that the finance.vote team cannot take them down. *The utility of FVT was immediate and irrevocable, and therefore not dependent upon future action by the finance.vote team.*

The locus of control in the finance.vote ecosystem will be continually and progressively transitioned to the token holders in totality over the course of the ecosystem development. Key functionality will be developed along a phased deployment of the network, with the finance.vote team retaining control only for as long as absolutely necessary. The final form of the network will be a decentralised, permissionless system governed by the FVT and identity holders.

1.4 Tools used in our system

Contents

1.4.1 IPFS

IPFS is a distributed system for storing and accessing files, websites, applications, and data. IPFS is a peer-to-peer (p2p) storage network. Content is accessible through peers located anywhere in the world, that might relay information, store it, or do both. IPFS knows how to find what you ask for using its content address rather than its location.

IPFS Documentation: >IPFS<

1.4.2 Ganache CLI

Ganache is a personal blockchain for rapid Ethereum and Corda distributed application development. You can use Ganache across the entire development cycle; enabling you to develop, deploy, and test your dApps in a safe and deterministic environment.

Ganache CLI(formerly known as the TestRPC) is the command line version of Ganache.

Ganache CLI uses ethereumjs to simulate full client behavior and make developing Ethereum applications faster, easier, and safer. It also includes all popular RPC functions and features (like events) and can be run deterministically to make development a breeze.

Ganache CLI Readme for command-line documentation: >Ganache CLI<

Installation:

```
npm install -g ganache-cli

or

yarn global add ganache-cli
```

1.4.3 Truffle Suite

Truffle is a development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM). It is based on Ethereum Blockchain and is designed to facilitate the smooth and seamless development of DApps. With Truffle, you can compile and deploy Smart Contracts, inject them into web apps, and also develop front-end for DApps. Today, Truffle is one of the most widely used IDEs for Ethereum Blockchain.

Truffle documentation: >Truffle<

Installation:

```
npm install -g truffle
```

Requirements:

```
NodeJS v8.9.4 or later
```

* Recommendations for Windows

If you're running Truffle on Windows, you may encounter some naming conflicts that could prevent Truffle from executing properly. Please see the truffle documentation section on resolving naming conflicts for solutions: >Here< .

1.5 Weekly Updates

1.5.1 2021

- /weekly/2021_05_may
- /weekly/2021_06_june
- /weekly/2021_07_july
- /weekly/2021_08_august
- /weekly/2021_09_september
- /weekly/2021_10_october
- /weekly/2021_11_november
- /weekly/2021_12_december

1.5.2 2022

- /weekly/2022_01_january
- /weekly/2022_02_february
- /weekly/2022_03_march
- /weekly/2022_04_april
- /weekly/2022_05_may
- /weekly/2022_06_june
- /weekly/2022_07_july
- /weekly/2022_08_august

1.6 Introduction

The cryptospace is a proving ground for genuinely new democratic models. With the emergence of the decentralised finance (DeFi) movement, new and radical ways of reaching consensus and coordinating around money are being created with rapid open innovation.

Although early DeFi applications have been active since 2017, it is the emergence of governance tokens that has caused explosive growth in the DeFi space. They are the mechanism by which true decentralisation can be achieved, pushing trust away from contract creators and onto token holders.

They are however, largely flawed in their implementation. Highly asymmetric token distributions, voter apathy[1] and huge gaps between the technical knowledge of core teams and token holders creates a context where genuine governance is surface level at best and thinly veiled centralisation at worst.[2]

[1] "Blockchain Voter Apathy. Governance is a key area of . . . ." 29 Mar. 2019, https://medium.com/wave-financial/blockchain-voter-apathy-69a1570e2af3. *Accessed 26 Aug. 2020.*

[2] "Deconstructing 'Decentralization': Exploring the Core Claim of . . . ." 13 Feb. 2019, https://www.ssrn.com/abstract=3326244. *Accessed 26 Aug. 2020.*

As a plethora of new tokens enter the market, it becomes increasingly difficult to keep up with new technological developments, but also separate out quality projects from low quality clones, or outright scams. Market signalling is a primary economic cost in the cryptospace and finance.vote will allow users and entrepreneurs to identify the impact of their signalling activity as well as provide early access to market signals.

finance.vote is a decentralised application for reaching consensus across the cryptospace[3] as a whole. It provides a space for users to engage with market discovery on new and existing tokens and be incentivised to share their perception on future price action.

Finance.vote has three cryptoeconomic components:

- Prediction and Market Discovery.
- Second Layer Governance.
- Decentralised Social Trading.

1.7 Prediction and Market Discovery

1.7.1 Turning Degens into Alpha

Problem:

- The perennial explosion of altcoins produces some winners and mostly losers.
- The permissionless of Uniswap and other DEXs creates vast, rapidly evolving and noisy markets.
- Early signals on the potential of token value are dominated by a small number of influencers with ulterior motives[4].

DAOs and prediction market dynamics demonstrate huge potential as tools for improving large scale decision making and governance systems[5]. However, broad spectrum prediction market systems such as Augur, although maturing, have yet to find meaningful adoption[6]. Largely, this is due to the open ended parameter space in which predictions can be made, which dilutes liquidity and interest across the markets. finance.vote lenses focus and adoption into a small number of shared vote markets, with constant and far more immediate settlement. Liquidity issues are resolved through the finance.vote token economics, which distributes a small amount of token inflation into reward pools, which seeds liquidity for every vote.

Using the semantic ballot voting system, users are presented with the ability to vote on the future market success (or failure) of tokens from across the cryptospace. This allows users to make market bets across multiple tokens, all contained in a single transaction.

$FVT rewards are claimed from reward pools by users who are correct in their predictions. This causes a progressive aggregation of voting power to those who can consistently make accurate market predictions, chaining prediction market decisions through reputation factors.

[3] "A signaling theory model of cryptocurrency issuance and value." https://ethresear.ch/t/a-signaling-theory-model-of-cryptocurrency-issuance-and-value/1081. *Accessed 26 Aug. 2020.*

[4] "Cryptocurrency Scam As Crypto Influencers Tweet About . . . ." 15 Jul. 2020, https://www.forbes.com/sites/rogerhuang/2020/07/15/likely-cryptocurrency-scam-as-crypto-exchanges-and-influencers-tweet-about-cryptoforhealth/. *Accessed 26 Aug. 2020.*

[5] "DAOs, Democracy and Governance - Ralph Merkle." 31 May. 2016, https://merkle.com/papers/DAOdemocracyDraft.pdf. *Accessed 26 Aug. 2020.*

[6] "Augur Price Analysis- Project matures but user numbers still low." 7 May. 2019, https://bravenewcoin.com/insights/augur-price-analysis-project-matures-but-user-numbers-still-low. *Accessed 26 Aug. 2020.*

As adoption increases, a range of voting markets will be introduced through the finance.vote governance system, which will allow users to accrue reputation in the system based on their ability to accurately predict verifiable on-chain market metrics.

1.8 Semantic Ballot Voting

Semantic Ballot Voting is a new kind of voting system designed specifically for finance.vote. It utilises a stack of semantic tags (in this case token tickers for tradable cryptocurrencies) and quadratic voting[7].

Users are presented with a market ordered list and are requested to use their vote power tokens ($V) to vote on the cryptocurrencies of their choice. They convert $V into votes, by quadratic voting:

$$\text{Votes}^2 = \$\text{V cost}$$

Meaning that every subsequent vote has a non-linear cost.

This system enforces prioritisation and ensures that users cannot vote strongly on every item, an issue that decreases validity in conventional voting and surveying systems. This scheme is used repeatedly throughout the finance.vote ecosystem and helps build weighted consensus across the whole network.

1.9 Digital Identity Tokens

Digital identity is a crucial component of voting technology. It is particularly important in quadratic voting systems, where if it is trivially easy to create multiple identities, then the quadratic system trends back towards being a linear system and honest actors are left at a disadvantage.

The act of creating multiple identities with the intent of corrupting a system is known as a Sybil attack. This is an issue globally across many social systems including most legacy social media systems, such as Twitter and Facebook. Since it is obvious that narratives can be manipulated by controlling the frequency of certain hashtags, the incentives exist to manipulate social consensus through Sybil attacks. It's the Sybil War.

In this increasingly adversarial context, all applications in the future will require a degree of Sybil resistance. This is typically obtained, mostly ineffectively, by collecting some piece of identifying information, such as a phone number, or more aggressively, state issued identity documentation. In permissionless systems, this is an unacceptable solution.

Finance.vote utilises a novel system we call Decentralised Identity Tokens (DITs). These take the form of ERC721 compatible NFTs, which represent an identity within the system. Users will not be able to vote in the system without one and cryptoeconomic dynamics are used to prevent the trivial creation of identities.

[7] "Quadratic Voting: How Mechanism Design Can … - SSRN." 13 Feb. 2012, https://www.ssrn.com/abstract=2003531. *Accessed 26 Aug. 2020.*

1.9.1 Decentralised Identity

The DITs in the finance.vote ecosystem contain the following information:

- The voting history of the identity.
- The reputation of the identity, denoted in voting power $V.
- An adoption metric, denoted by a number i.e. $FVT1 to $FVTn, where n is the issuance number.
- Metadata allowing customisation of the DIT look and feel.

A default skin for users' will be generated in the Obelisk phase from procedurally generated art, producing a one of a kind artwork for each DIT. Certain numbers will have increased significance.

1.9.2 Identity Customisation

The finance.vote digital identity tokens start as an NFT that simply contains users voting histories and therefore performance within the market. In the Obelisk phase of the network these will gain their own procedurally generated provably unique digital art. However, they are customisable to users preference and are intended to be used as avatars in the system.

Users will be able to link their DIT with other NFTs they have purchased from a marketplace or from auction.vote, showcasing their art choices to other users on the platform.

1.9.3 If you can't beat them, join them

In the finance.vote ecosystem, Sybil resistance takes the form of ensuring that users cannot corrupt the consensus outcome by splitting votes across multiple ballots and fabricating multiple identities. In reality, this cannot be entirely stopped.

We therefore take an economic, pay-to-sybil mitigation approach, to reach a state where good intelligence on the number of Sybils in the system is known. Before a user can engage in voting activity on finance.vote, users must acquire a minimum of 100 $FVT, which will be sent to the identity minting address and burnt, which assigns voting rights to a user Ethereum address. Only addresses, where a DIT is present will be able to vote, addresses with multiple DITs will be able to vote multiple times.

1.9.4 A Cryptoeconomic Line of Defence

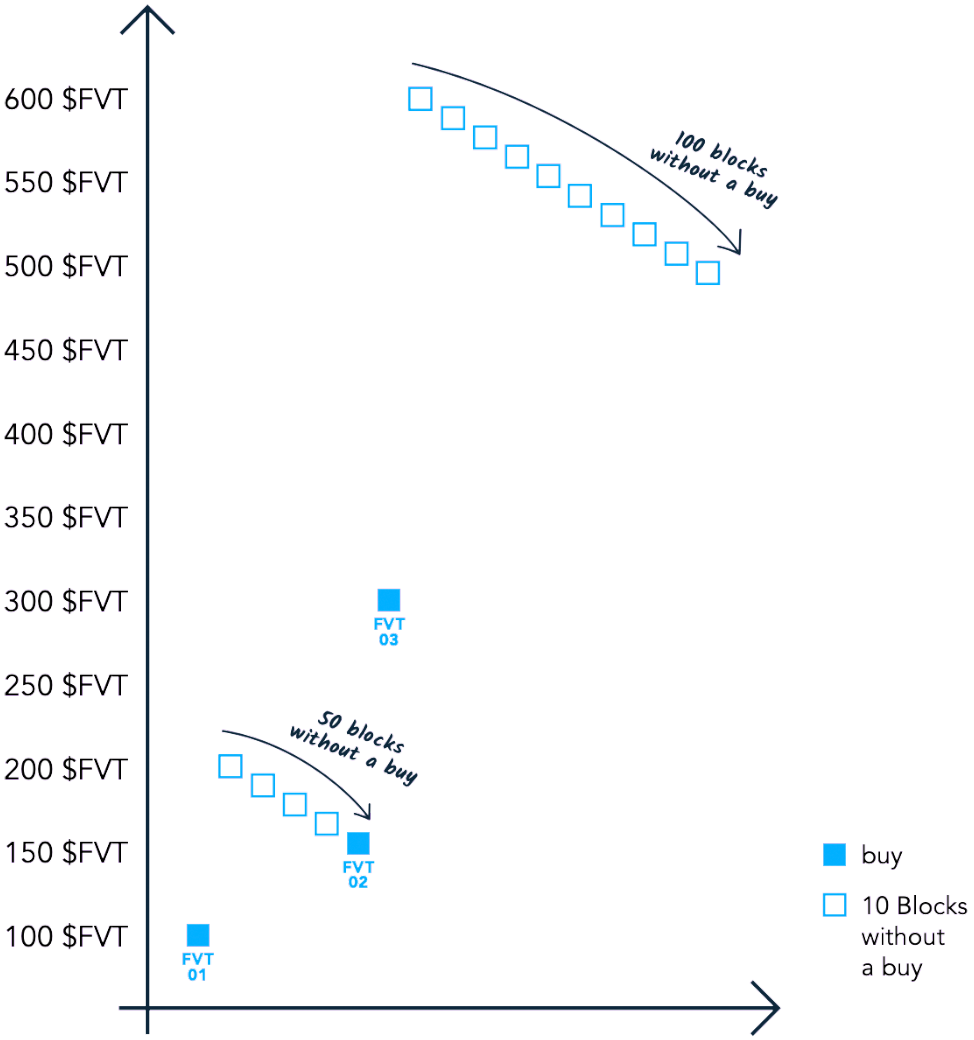
Using a similar mechanism to that found in the auction.vote system, the price of DIT will increase exponentially based on demand. If a user buys an identity for 100 $FVT the price will double to 200 $FVT for the block immediately after the buy. This price will then decay at a rate of 1 $FVT per block until it reaches the price floor of 100 $FVT, unless a subsequent buy occurs at a price delta between 200 $FVT and the price floor.

All $FVT is burned on identity creation, ensuring that token holders benefit from adoption and Sybil activity.

An example buy sequence:

- The first user in the system purchases, FVT1 for 100 $FVT.
- Over the period of 100 blocks the price returns to 100 $FVT.

- FVT2 is bought for 150 $FVT and the price in the next block jumps to 300 $FVT, where the same user immediately purchases FVT3.
- The price is now 600 $FVT for an identity and users wait until the price decays until their perceived value of a DIT is reached.



1.9.5 Tradability

finance.vote DITs will be tradable as are any other NFTs. Therefore, it is possible that users may be able to accrue reputation and therefore vote power on an identity and then realise that work and skill into a profit from their initial purchasing price.

Users may also wish to trade identities for their aesthetic or perceived value based on their issuance number; lucky numbers are very real to some.

1.9.6 Decentralised Reputation

Identities within the system are designed to allow users to accrue reputation based on their ability to make effective market predictions and influence consensus formation in the second layer governance system.

These dynamics are difficult to Sybil. The nature of our chained voting system ensures that progressively correct accounts earn greater reputation and voting power over time and the likelihood that a user can successfully construct these accounts randomly as opposed to playing honestly diminishes over time.

1.9.7 Summary

Many presume that Sybil resistance can only be obtained through the use of hard identity solutions. We propose that pseudo-anonymous identity formation can occur in decentralised systems through the use of non-fungible tokens. We believe this will be an important piece in the evolution of DeFi and decentralised identity.

1.10 Vote Markets

1.10.1 Introduction

The finance.vote network leads with a quadratic voting based prediction market system we call “vote markets”.

The system is designed as a crypto economic game that marries governance with the markets. It is designed to aggregate collective intelligence from a distributed group of pseudo-anonymous crypto users who have their finger on the pulse of the crypto market.

Quadratic voting is used to generate a consensus in a perceived future market order. This is done by presenting each user with a token list that is by default ordered by the market. Users then spend a budget of vote power to create a new order, based on their perception of token quality and future potential market performance.

The resultant aggregation of user ordered token lists creates a distribution of perceived market order in the form of a consensus list.

At this point in the future, users are rewarded with a proportional share of a network generated reward pool depending on the proportionality of their correctness.

This simple mechanic becomes exponentially more powerful as new and diverse markets are added and as the power of the rules of the game is transitioned to token holders.

The purpose of this game is to create an adversarial environment to release and battle test our quadratic voting technology, including our decentralized identity token system so that it can be used to aggregate and curate collective intelligence and reach effective decisions in our second layer governance system.

The vote markets have the following properties:

1.10.2 One Sided

Prediction markets are limited by adoption and liquidity. Similar to the issues seen with long tail assets such as those found in emerging altcoin markets, when liquidity is thin price discovery becomes difficult.

Prediction markets turn predictions into tradable assets. This is potentially profoundly disruptive and has the potential to create futarchy based DAO governance structures in the future. However, the fidelity of predictions are contingent on the number of players in the game and further, the depth of their order books.

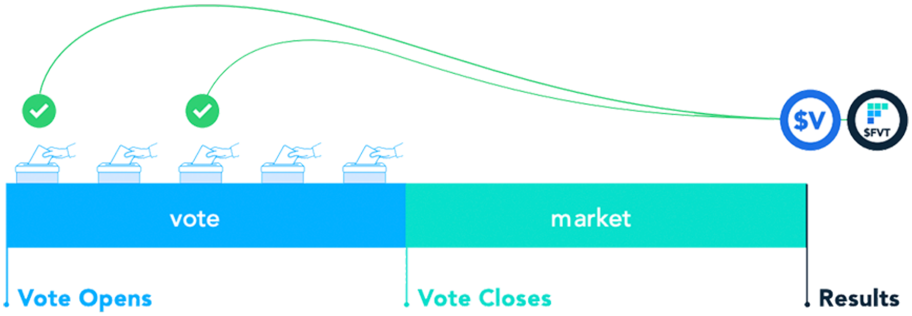
AMMs have opened up the long tail asset space by ensuring that there is always a trade available at a given price through algorithmic price discovery mechanisms. The pay off is slippage. They are one sided markets, a smart contract becomes the counter party.

The finance.vote vote markets use a similar design philosophy by seeding liquidity using a reward pool system. Your counterparty is the network, rather than a trader. The result is a collective reward pool that is distributed programmatically to winning votes.

In the same way that uniswap has eliminated order books from token trading, we eliminate order books from prediction systems.

1.10.3 A Vote / Market Window

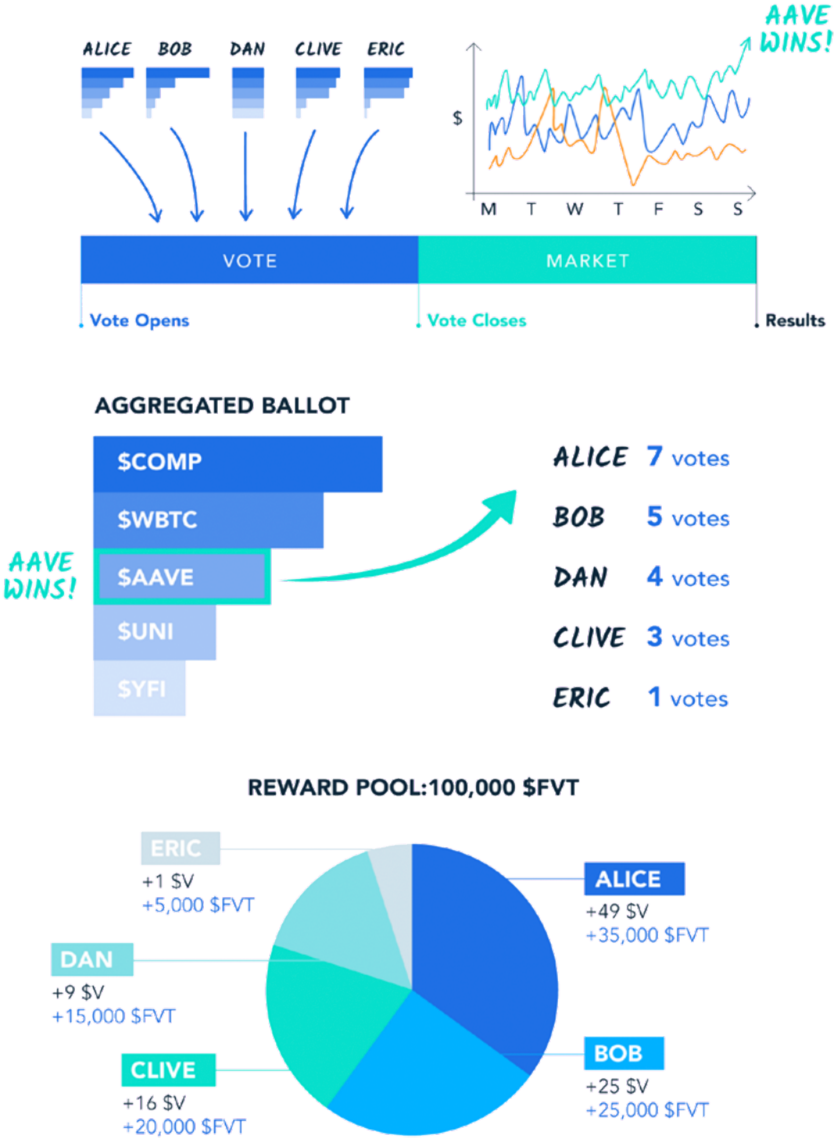
Each vote market comprises two time bounded windows, a vote window and a market window. In the vote window, users have a period of time to submit their votes, signalling their preferred market ordering, when this window closes, the market window opens. Here, no more votes are submittable and the predictions of voting users play out in the market. At the closing block of the market window, typically weekly at UTC market close, a snapshot of the top gaining token, is determined by a market oracle and is compared against the consensus state.



1.10.4 Token Rewards

In the opening state of the system the reward pool is best conceptualised as an incentive for effective research. The cost of entry is the purchase of a decentralised identity token and the gas costs required to submit an on-chain vote utilising the vote market smart contracts.

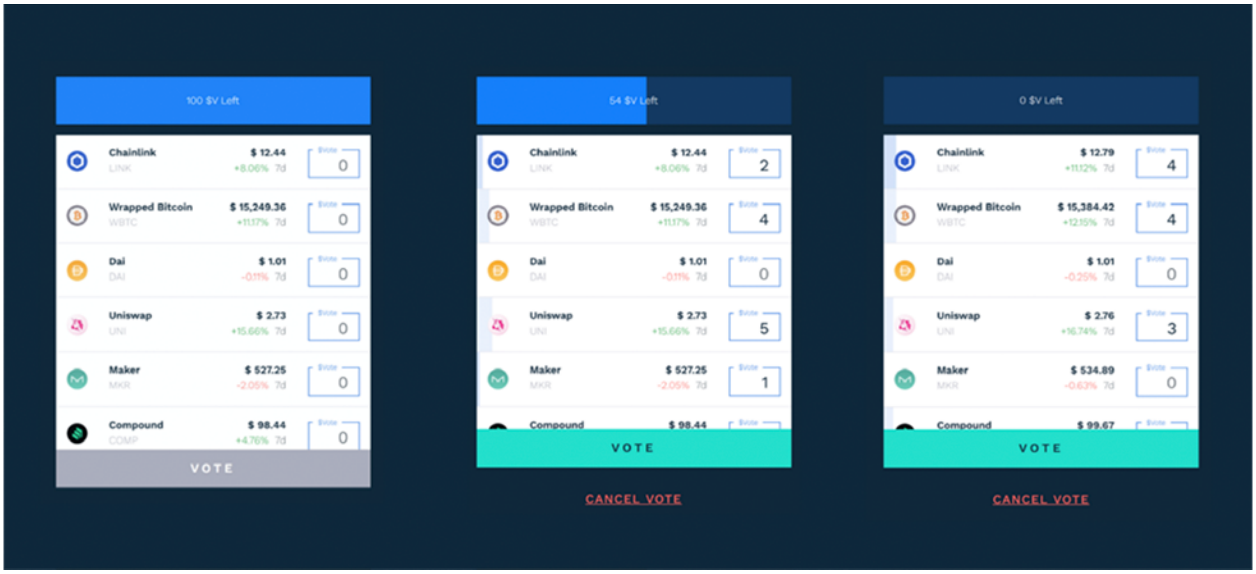
The user vote pool share is allocated as follows[8]:



[8] Note that this is computed as votes, not $V, which are votes2, whereas the $V bonus is equal to the $V spent on the winning ticket.

1.10.5 Quadratic Voting

Quadratic voting first proposed by Glen Weyl and Eric Posner[9] is a voting paradigm that is designed to allow users to vote more than once on a particular issue, moving beyond the 1 person, 1 vote (1p1v) paradigm conventional in most democratic systems. Instead, voters are allowed to buy more votes. The cost being that the more subsequent votes that are bought, the greater the cost by the square of the votes. This allows users, particularly those with minority views points to express their preference with more intensity than would be possible in a 1p1v system.



100 $V Left

Chainlink LINK	$ 12.44 +8.06% 7d	$Vote	0
Wrapped Bitcoin WBTC	$ 15,249.36 +11.17% 7d	$Vote	0
Dai DAI	$ 1.01 -0.11% 7d	$Vote	0
Uniswap UNI	$ 2.73 +15.66% 7d	$Vote	0
Maker MKR	$ 527.25 -2.05% 7d	$Vote	0
Compound COMP	$ 98.44 +4.76% 7d	$Vote	0

54 $V Left

Chainlink LINK	$ 12.44 +8.06% 7d	$Vote	2
Wrapped Bitcoin WBTC	$ 15,249.36 +11.17% 7d	$Vote	4
Dai DAI	$ 1.01 -0.11% 7d	$Vote	0
Uniswap UNI	$ 2.73 +15.66% 7d	$Vote	5
Maker MKR	$ 527.25 -2.05% 7d	$Vote	1
Compound	$ 98.44	$Vote	

0 $V Left

Chainlink LINK	$ 12.79 +11.12% 7d	$Vote	4
Wrapped Bitcoin WBTC	$ 15,384.42 +12.15% 7d	$Vote	4
Dai DAI	$ 1.01 -0.25% 7d	$Vote	0
Uniswap UNI	$ 2.76 +16.74% 7d	$Vote	3
Maker MKR	$ 534.89 -0.63% 7d	$Vote	0
Compound	$ 99.67	$Vote	

1.10.6 Meritocratic Reputation System

In our initial markets, users can amplify their vote power beyond the starting level by demonstrating a history of correct decision making in the markets, or by purchasing more identities.

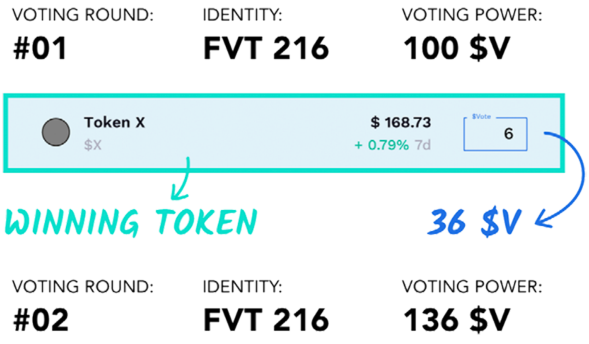
Vote power in the system is denoted by the use of an internal vote power token $V. Every identity is airdropped a budget of vote power tokens to spend in each vote market, in every voting window, we call this “Power UBI”.

Each user starts with a Power UBI of 100, giving them 100 $V to spend on votes in a vote market. The base UBI level for each identity increases based on their history of correctness.

For example, if a user votes on $TokenX 6 times in a “Winner” vote market and $TokenX is the highest gaining token at the end of the market window. Then that identity has spent 36 $V on a winning prediction and will therefore receive a 136 $V Power UBI from that point on. There is no reduction in this user’s UBI after this point. The reputation system is built on non-punitive, positive reinforcement only.

This ensures that over time vote power balances and therefore sizes of claim on the reward pool will trend towards users with a history of correctness in the markets.

[9] “Voting Squared: Quadratic Voting in Democratic Politics” by . . . .” https://scholarship.law.vanderbilt.edu/vlr/vol68/iss2/3/. *Accessed 17 Nov. 2020.*



1.10.7 Minimal Cost of Entry

Engaging in early altcoin markets is the best way to gain high risk / reward trades. It is also the best way to get rekt. With a heavy bias towards the latter.

finance.vote offers users the opportunity to make market bets on the altcoin markets without purchasing and holding the hyper volatile underlying assets. Once users hold a voting identity they get a free bet in every vote market on the finance.vote network, outside of the network costs of submitting the bet.

In subsequent vote market releases it will be possible to stake $FVT to increase exposure to these bets, however in our early phases users will win shares in the reward pool and gain vote power without the requirement to stake capital.

It is not necessarily the case that those with the most wealth, hold the most knowledge. We use our semantic ballot voting system to aggregate knowledge from a distributed group of crypto users and aim to strip out the plutocratic influence that wealth has on the market, creating a microcosmic merit-based market environment.

1.10.8 A Gamified School of Decentralised Finance

The vote markets allow users to engage with the market in a lower risk and downside protected environment, where they can hone their trading skills and build fundamental knowledge about the cryptospace.

finance.vote identity holders will gain access to educational material that aims to boost users' understanding of the decentralised technologies of the future. It is in the interest of the network to amplify the collective intelligence of its user base and this will take place at first in existing social media platforms and then on our second layer governance system, which will be a consensus curated dialogue and decision framework optimised for this purpose.

1.10.9 Price Discovery

It is an existing paradigm in the cryptospace and indeed the wider market system to price productive enterprise relatively against one another. Is Microsoft better than Google? Is Aave better than Compound? Our vote market system asks this question consistently, cyclically and with a reputation system that ensures that market intelligence is rewarded for these activities.

Market capitalization and other metrics are a poor indicator of quality in the cryptospace, especially as token economics and inflationary dynamics in the token systems often lead to wildly inaccurate initial pricing of tokens.

It is our intention that the vote market system generates alpha based on the fundamental value proposition of tokens in the crypto economy.

1.11 The Vote Market Roadmap

The vote market concept is vast in scope. We roadmap a number of vote markets as network leaders, however it is intended that the token holders and the DMF take over the responsibility of generating new vote market concepts in the future.

1.11.1 Launch Market - DeFi (Winner)

Our launch market is aimed at where the action is, the DeFi market.

DeFi is the most rapidly growing and changing market on the planet. Innovation occurs at lightning speed and some products, the "DeFi Blue Chips", can ship new contracts with great frequency, shifting fundamentals dramatically.

This is the perfect context for testing the efficacy of the vote markets as an alpha generating system.

1.11.2 DeFi (Loser)

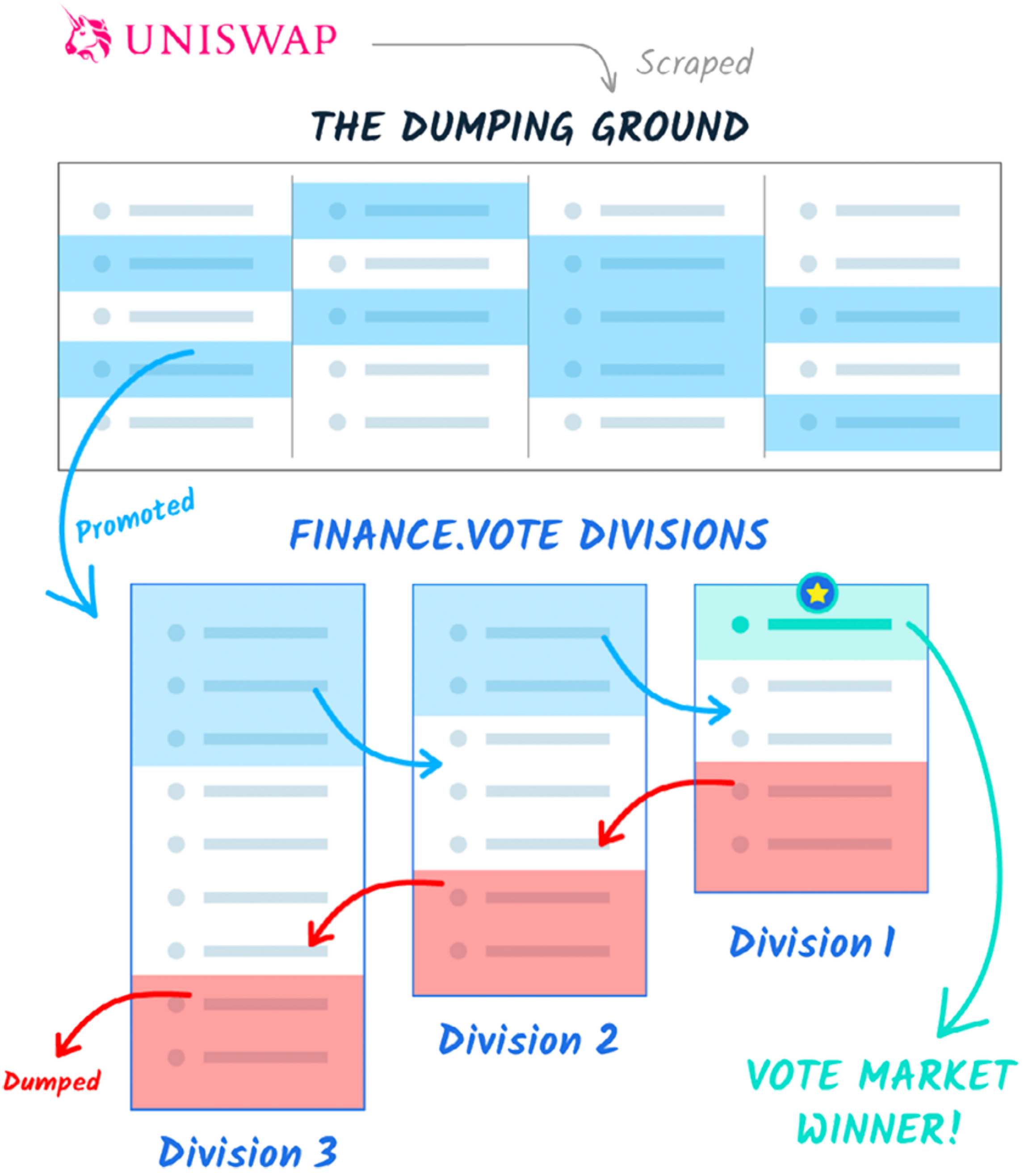
Just as DeFi products can ship new game changing contracts that add new decentralised financial primitives to the market. They can just as easily be compromised through an exploit resulting in the loss of user funds.
These first two markets will be the capstone vote markets for the system.
DeFi Vote Markets - Initial Token List Curation
We begin by using the top ten tokens from the CMC DeFi token list.

This list will be modified through deliberation on our social media channels and a rolling snapshot vote (in advance of our second layer governance system) will be held to determine token list inclusion. A maximum of one token per week will be added to the DeFi token list until a fully automated process is completed.

1.11.3 New Tokens

A vast amount of new tokens are listed every day across the decentralised exchange ecosystem. The outstanding majority of these tokens are valueless, or outright attempts to scam users of the cryptospace.

The "New Tokens" vote markets will be designed to utilise the collective intelligence of the finance.vote user base to filter these tokens through a successive iterative league table system.



New tokens will be indiscriminately scraped from the uniswap contract ecosystem into a token list called "The Dumping Ground." A top slice of these tokens will be added to the finance.votedivision system.

In the division system. A nested layer of leagues will be used to to filter tokens via promotion and demotion until they hit the top of the division one vote market. These winning tokens willgraduate to more senior markets, such as the DeFi Winner market.

1.11.4 Thematic Token Lists

As our token curation system matures, a proposal system will be introduced that allows users to compile thematic token lists related to tokens aligning to specific technological affordances eg. Layer 2 tokens, Privacy Tokens, Classic Alts etc.

1.11.5 REKT

The REKT market is a vote market based on predictions of token blow up.
This vote market will run until one of the tokens in the token list experiences a 90% drawdown in price from the initial market snapshot.

This market will be utilised to detect crowd perception of stablecoin blow up risk, with a basket of tokens that span a range of market maturity. Which projects are most likely to beexploited? Which projects are likely to experience a governance attack, or catastrophic governance failure? Which are under existential threat?

It is the goal of the Rekt market to collect this intelligence.

1.11.6 Experts

The "experts" vote market is for pre-release tokens.
In this market, a group of experts are invited to take part in a vote and will be attributed special "Expert" voting rights.

The experts will quiz token creators on their proposed future projects and the experts will vote with their identities to generate a collective expert vote. Simultaneously, users will votewith their identities, aiming to predict the outcome of the expert vote.

The experts become the 'oracle' in this system and the vote market is settled on their vote.

The outcome of this vote launches the successful token immediately in a liquidity bootstrapping event on auction.vote.

1.11.7 Memes

Meme markets, the most difficult market of all. The generation of high fidelity meme voting will be highly complex and contested. Achieving this will require advanced blockchain governance.

1.12 Staking

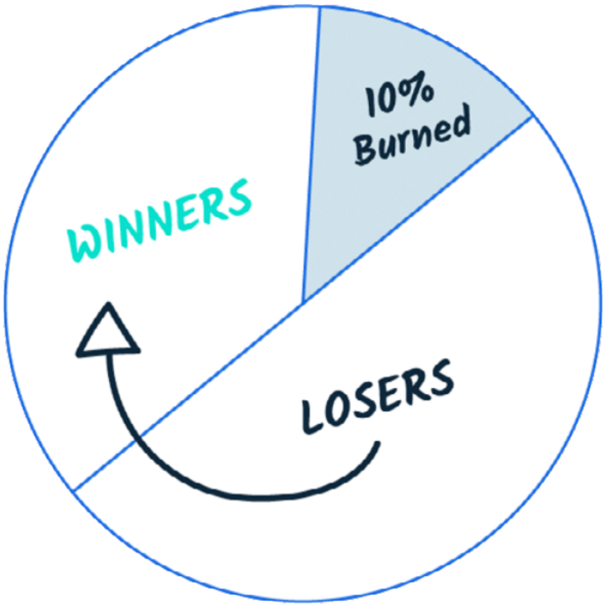
The vote markets will not stay as merely as a collective intelligence and consensus tool.

More effective price discovery will arise from those with skin in the game. For those users who wish to increase their exposure to these market bets, a staking system will be released thatwill allow users to stake $FVT on the market prediction outcomes.

Architecturally, the staking system will be a different layer from the voting system reward pool.

Our first staking system will allow staking $FVT on an identity for a single round of a single vote market. This identity may or may not belong to the staker, therefore this will be thefoundation of social trading, betting on an identity that does not belong to you.

In this formulation, every (round, tournament) pair will have a collection of stakes, up to a maximum of one for every identity. The set of identities for which there are non-empty stakes will essentially be in competition with each other, with the total $FVT stake gettingredistributed among the various stakes in proportion to how their identity voted on the correct outcome. In this game, the losers pay the winners.



For example, suppose there are 3 identities, Alice, Bob and Caroline, and they vote on the winning choice with votes 1, 3 and 0, respectively. Suppose further that the stakes on Alice, Bob and Caroline are staking 10, 20 and 30 $FVT, respectively.

Then once the winner is announced, the total stake on that round (10 + 20 + 30 = 60 $FVT) would be distributed proportionally to the stakes, with 1 / (1 + 3 + 0) = 1 / 4 of the 60 = 15 going to the stake for Alice and 3 / (1 + 3 + 0) = 3 / 4 of the 60 = 45 going to the stake for Bob, and 0 going to the stake for Caroline. In this situation, the stake for Alice will have started at 10 and ended at 15, a 50% gain, and the stake for Bob will have started at 20 and moved to 45, a 125% gain. In essence, the losing bets are reflowed to the winners. Natural selection is beautiful.

10% of the losing stake will be burned in each round.

The staking system is still in the design phase, however is targeted for release in the Obelisk phase of the network. It will allow fair staking of multiple coins simultaneously, a first in both the cryptospace and the wider financial system, as far as we are aware.

1.13 Second Layer Governance

1.13.1 Bringing Politics to DeFi

Problem

- Many established networks lack any formal governance mechanism to receive accurate signals from their token holders regarding the future direction of the network.
- Loose and chaotic social signalling systems, such as Telegram and Reddit are ineffective for reaching consensus, are prone to spam and are an inconvenience for teams.
- DeFi DAOs are often heavily gated and cornered by asymmetric token issuances, meaning that users have no meaningful influence on the network.
- Permissionless systems allow projects to be easily cloned and users can be duped into buying fake tokens or engaging with projects that are outright scams[10].

1.13.2 Introduction

Governance is what allows you to get things done as a collective. It is the process of making decisions about decisions. It is the act of generating effective processes that allow people to organise at scale and move towards a productive outcome. It is the reason why we have the modern civilisation that we have today, allowing us to form coherent social self-organised structures that churn out innovation and productivity. And now we are doing it on the internet.

It can not be underestimated how important this transition is. Typically, the power to organise and work towards a productive outcome has taken place within the confines of institutions, organisations and firms, but now humans are coalescing in digital spaces and forming movements. They are not very productive yet and typically fall prey to the tyranny of structurelessness, but with blockchain based governance systems that could change.

[10] "Fake Tokens Continue to Plague Uniswap - Cointelegraph." 19 Aug. 2020, https://cointelegraph.com/news/fake-tokens-continue-to-plague-uniswap. *Accessed 26 Aug. 2020.*

The missing piece for decentralised global organisations is voting technology. Outside of the confines of institutional governance, there is no control of the narrative. No way to keep conversation on track towards a common goal. The establishment of norms within a group becomes tricky, as self-interested parties, consciously or otherwise, drag discourse towards their own goals. What they are missing is a means to reach consensus.

The finance.vote second layer governance system uses the construction of decentralised financial systems to experiment with ways to use voting technologies to experiment with social consensus formation.

1.13.3 Hierarchical Governance

Governance at the protocol layer is a hotly contested narrative across the cryptospace[11]. Cryptonetworks require consistency, transparency and incredibly high levels of security and this does not lend itself to particularly agile governance structures. Some networks have attempted to innovate beyond this and in many cases it has led to corruption[12], collusion[13] and sometimes outright failure[14] of the tokens resulting in permanent loss of funds.

More recently, Decentralised Autonomous Organisations (DAOs) have become increasingly prominent in emerging crypto networks, especially in the DeFi space. They can allow substantial changes to take place, pivoting to new contracts and shifting monetary policy in ways that can have a large impact on token holders. They are risky and should be used sparingly.

We propose a multi-level hierarchical approach to governance, which separates high stakes reality altering decisions to a highly secure, scope limited "layer one DAO". Thus, leaving the full gamut of decision making formation arising from dialogue and rough consensus formation to layer two miniDAOs.

1.13.4 Lobbying and Social Consensus Formation

Up until now, DAOs have been poor places to engage in complex decision making. Typically, they are substantially gated, requiring sometimes exceptionally high token stakes to demonstrate any meaningful voice[15].

The finance.vote second layer governance system is designed to allow social signals to form, be amplified and captured in token specific miniDAOs. These systems will demonstrate the first use of the semantic ballot voting system outside of asset price discovery. Here, the system will be turned towards dialogue, content curation and decision making.

It is the intention of this system to provide a space that users can reach consensus on what it is that they desire their chosen networks to focus on and move towards. It will be a space to share research, build knowledge and reach shared meaning.
Our semantic ballot voting system will utilise quadratic voting to build decentralised curation processes. Decisions and proposals that receive sufficient consensus will be able to transition to the layer one DAO, the DMF for consideration.

[11] "Against on-chain governance. Refuting (and rebuking) Fred …." 30 Nov. 2017, https://medium.com/@Vlad_Zamfir/against-on-chain-governance-a4ceacd040ca. *Accessed 26 Aug. 2020.*

[12] "Tron's Takeover of Steemit Is Internet History Repeating Itself …" 5 Mar. 2020, https://www.coin-desk.com/trons-takeover-of-steemit-is-internet-history-repeating-itself. *Accessed 26 Aug. 2020.*

[13] "Leak reveals collusion on EOS blockchain - The Block." 1 Oct. 2018, https://www.theblockcrypto.com/linked/1015/leak-reveals-collusion-on-eos-blockchain. *Accessed 26 Aug. 2020.*

[14] "YAM Finance Crashes Over 90%, Founder Admits His Failure." 13 Aug. 2020, https://cryptopotato.com/yam-finance-crashes-over-90-founder-admits-his-failure/. *Accessed 26 Aug. 2020.*

[15] The Compound governance token ($COMP) requires 1% of the total token supply delegation to sub- mit a proposal, currently valued at $16.7m. https://compound.finance/docs/governance#introduction

1.13.5 miniDAOs

The second layer governance system is built to convert rough consensus and dialogue into actionable decisions in both the layer one DAOs of the finance.vote.

With the integration of semantic ballot voting, miniDAOs are dialogic spaces driven by cryptoeconomics. Users will be able to create their own votes, vote on discussion topics and curate ideas and sources.

The key to these spaces is content sorting. Users will utilise voting tools, that are live and battle tested in the adversarial environment of our vote markets. Finance.vote therefore sits in the governance space in between the extremes of high-stakes, on-chain governance and the loose social consensus formation in channels such as Reddit and Telegram.

1.13.6 Blockchain Agnostic

As a governance system in the emerging DeFi space, blockchain interoperability is paramount.

Finance.vote will begin life on the Ethereum platform and will always have Ethereum based components, however, our second layer governance system will have several forked development trajectories across a range of blockchains; the miniDAOs is where these development trajectories are realised.

In order to bootstrap and showcase the second layer governance system, the finance.vote launch team commit to deploying the following miniDAOs:

- The $FVT miniDAO: Ethereum based.
- A $BTC miniDAO: Bitcoin based.
- One other; TBC

This launch set of miniDAOs will be deployed during the Pyramid phase of the network. After this point, new miniDAOs will be released in tranches to blockchains who wish to spin up a decision making structure using network infrastructure. These miniDAO slots will be auctioned as DITs via the auction.vote system and development for them will be funded by the DMF treasury.

1.13.7 Scalable voting

The gas limitations in our vote markets are a fundamental component in their cryptoeconomics. These network costs imply a base economic value for the $FVT economy. If users are willing to pay these fees in order to obtain $FVT, then $FVT has some value delta above that base cost.

However, in our second layer governance system we want dialogue; curated on-chain chatter. This needs to be as cost effective as possible, but maintain the necessary degrees of sybil resistance to keep these space spam free and the voting systems effective.

Consequently, we hereby open a funded multi-year research project to develop scalable voting systems for deployment in our miniDAOs. This will include the formation of private voting solutions, whereby user identity can be selectively disclosed based on user choice.

The most high volume vote markets will be set up as “bake off” environments to test the latest on-chain scalability solutions, with the explicit focus of voting technology.

1.13.8 Plutocracy

1 coin, 1 vote systems (1c1v), perhaps inevitably, trend towards extreme plutocracy.

Highly uneven distribution of tokens are a feature of all token economies up to this point and this creates a significant issue in governance systems, as wealth turns from merely economic power, to coordinative power.

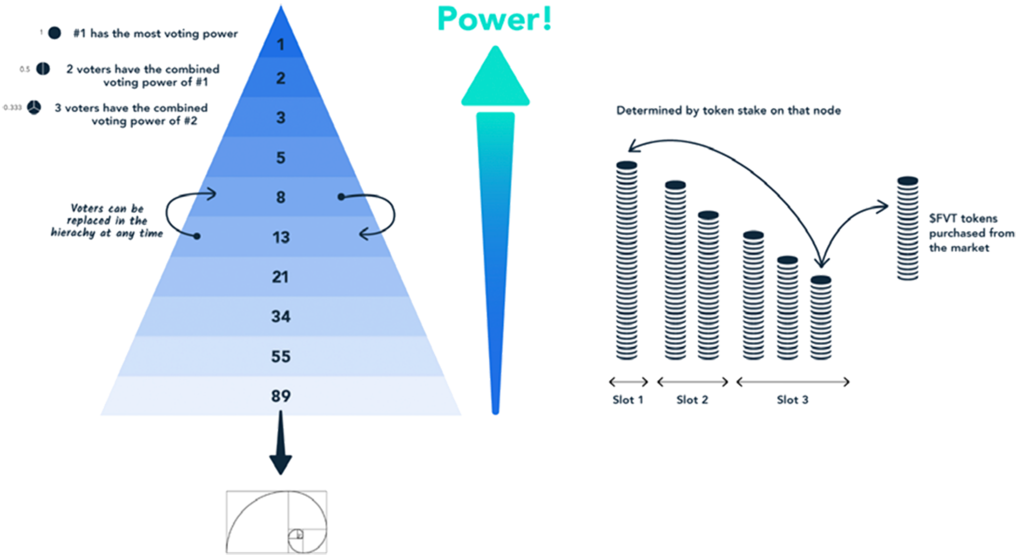
A central tenet of the finance.vote network is to explore mechanisms that break this direct correlation between wealth and power.

This is done though the use of the identity linked vote power system, which in the vote markets is generated through meritocratic means. This alone, will generate a separation between token holdings and influence within the system and our main decision making architecture will include both a wealth based and meritocracy based dynamics.

This dynamic is extended and nuanced in the second layer governance system, which utilises the .vote consensus system and the introduction of tuneable token stake mechanics that are cross-chain and blockchain specific.

1.13.9 The .vote consensus mechanism

The .vote consensus mechanism is the means through which token stake is used to weight voting power in each of the miniDAOs in the finance.vote ecosystem. It utilises a pyramidic stacking mechanism to normalise vote power across a voting population, ensuring that large token holders do not have an extremely out weighted voice in the system.


1
#1 has the most voting power
0.5
2 voters have the combined voting power of #1
0.333
3 voters have the combined voting power of #2
Voters can be replaced in the hierachy at any time
1
2
3
5
8
13
21
34
55
89
Power!
Determined by token stake on that node
$FVT tokens purchased from the market
Slot 1
Slot 2
Slot 3

Users are stacked in layers into staking slots according to the Fibonacci sequence.

This status dynamic ensures that token holders can utilise their wealth to increase their influence, but not so unduly that they dominate the system to the point of corruption. The interplay between users' $V balance will ensure that vote power is optimised to avoid plutocracy.

A single user with the highest token stake will have the highest weighted voice in that miniDAO. Token balances thereafter determine which tier of the consensus mechanism that they sit within. Each tier has a number of staking slots, which are populated by users based on their staked token balances. Each tier has the same staking power, but is shared between a greater number of people. The tiers scale in slot size out to infinity.

For example, a user with 100,000 $FVT tokens staked on the $FVT miniDAO has the highest stake on that node. The next nearest token balances are 95,000, 90,000 and 80,000. The 95,000 and the 90,000 stakers occupy the second staking tier and the 80,000 staker takes the third slot along with two others. The 100,000 staker has 1,000 $V and the 95,000 and 90,000 stakers have 500 $V, and next tier down have 333 $V and so on.

This mechanism incentivises users to purchase $FVT to stake in their chosen miniDAOs to raise their influence in the social consensus formation process, but mitigates plutocratic power formation.

1.13.10 Summary

In summary, the finance.vote second layer governance system is an adaptive pseudo- hierarchical DAO architecture that uses voting technology to experiment with new forms of human coordination.

1.14 Decentralised Social Trading

Trade with your enemies

Problems

- Although the cryptospace is trending towards being more collaborative, with pooled liquidity becoming the norm, how this is distributed to users will become fraught with governance complexity.
- There is currently nowhere to collaboratively trade and share market signals trustlessly within decentralised exchange space.

The finance.vote social trading system evolves out of the vote markets, into a gamified social trading system.

The digital identity tokens utilised in the vote markets, earn reputation and therefore a rank in the system.

This will allow status systems to be generated such as social league tables, whereby NFT identities can be displayed in a list in accordance with their performance in the various vote markets.

1.14.1 Identity Staking

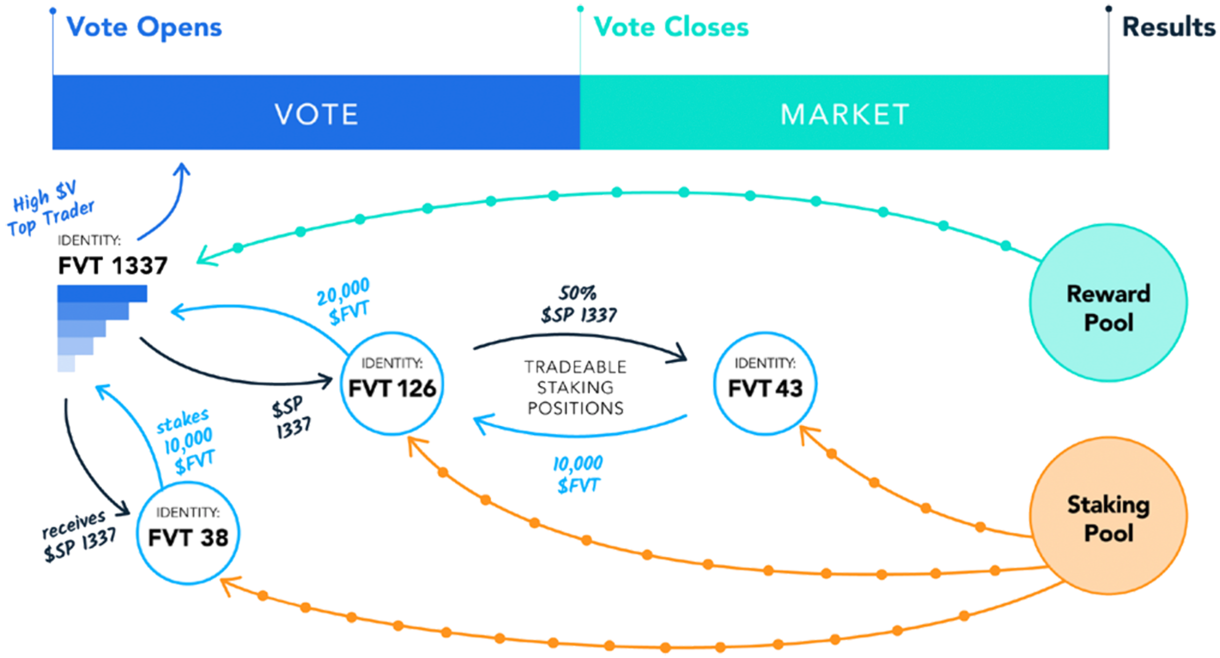
Our social trading system begins by allowing quadratic votes made by an identity to be tokenised. This will allow users to accrue reputation through effective market predictions and allow other users to stake $FVT on their identity. Staking users will win, if that identity wins, with the host identity earning 10% of the profits.

Staking positions take the form of ERC20 tokens ($SP tokens) that represent a stake position on a quadratic vote associated with a particular identity.

For example: 10 $FVT1SP tokens, will represent a share in the staking pool that is won by the $FVT1 identity based on their market bet.

Adding and removing $FVT to a stake will be allowed until voting stops, after which $SP tokens will be tradable on the secondary market until the bet is settled.

This mechanism allows users to take delegated hedged market bets against other staking participants in our vote markets. They are decentralised quadratic options generated by users with a history of market performance.



1.14.2 Micro Liquidity Pools

Pro users will be able to create their own vote markets and design their own token lists.

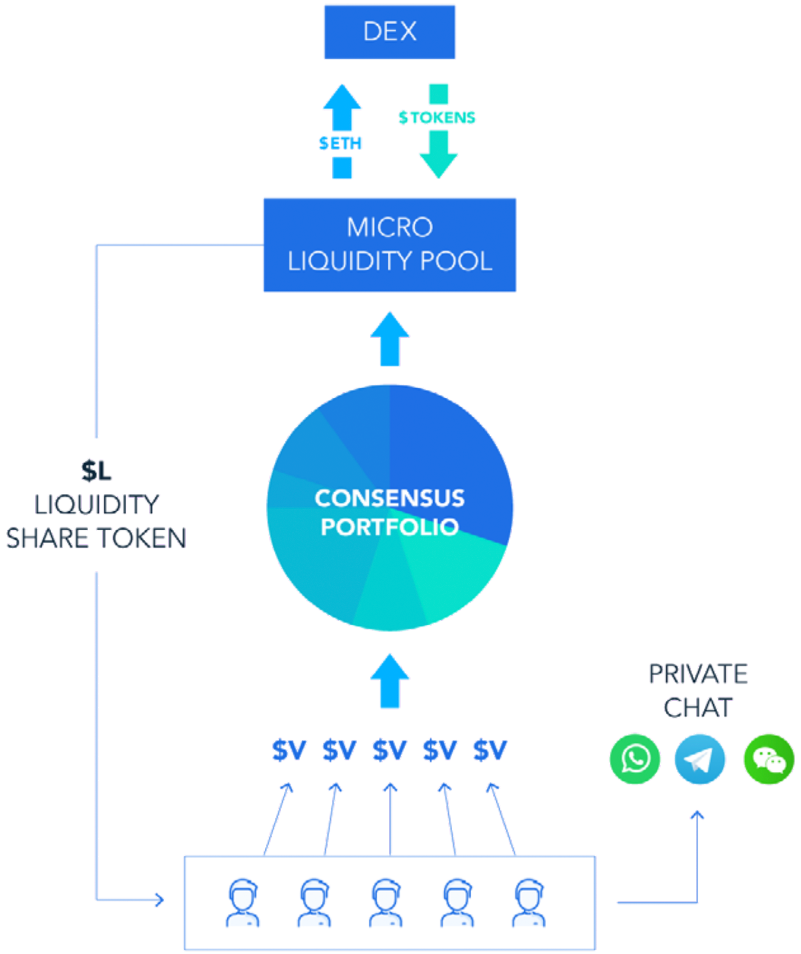
Private vote markets will be mintable with special DITs, that give permission to call a single vote market minting event. These special identity tokens will be auctioned periodically on auction.vote.

Once minted, these DITs will be able to issue identities to participants in their own chat groups or preferred social media bubbles.

Now, a group of decentralised users will be able to trustlessly pool assets and create a dynamic micro liquidity pool of digital assets that can be rebalanced via quadratic voting based consensus.

Users will hold LP tokens representing their stake in the micro liquidity pool and will be able to withdraw at any time after round resolution.

We anticipate that recruitment drives will occur for private pools, where high quality traders are invited to groups to share their alpha in exchange for a share in pooled liquidity winnings.



1.14.3 Gated Competitions

Special periodic vote markets will be created that will only allow identities with a threshold rank to enter.

For example, a "Pro" vote market is held where users with over 500 $V balances can enter. There are a limited number of voting slots, and the vote market contract only allows 100 identities to enter.

After this, a voting window is held on a governance determined token list and players play and stakers stake on their chosen identities.

1.14.4 Summary

The finance.vote social trading system introduces a gamified market environment with a human element. Users obtain status within the system and are ranked based on their market performance. More passive market actors can stake their capital on identities that have a proven history of performance in the market. Positions on those identities are tradable in market windows in advance of market settlement.

1.15 Governance

The finance.vote governance architecture is intended to introduce a new paradigm of governance into the cryptospace, utilising both a layer one DAO, the Decentralized Monetary Fund (DMF) and the Layer two $FVT miniDAO in a bicameral governance structure.

From early in the launch, users will be given the opportunity to materially shape the future of the network and progressively, layer one powers will be transitioned in their entirety to the token holders. $FVT will act as the governance token for the network alongside it's incentive generating base utility properties as a cryptocurrency.

1.15.1 The Decentralised Monetary Fund (DMF)

The DMF will build a novel DAO pattern optimised for control of the finance.vote monetary policy to the token holders. The DMF sets the inflation rate by creating new reward pools and initiating new voting markets. It also controls the ecosystem development fund, which will add full transparency to the assets designed to maximise the impact of finance.vote ecosystem within the crypto community.

1.15.2 The $FVT MiniDAO

The $FVT MiniDAO will showcase the potential of second layer governance by developing the finance.vote ecosystem in partnership with the users. A series of feature votes will be launched, early in the Pyramid phase, which will allow users to vote on future tournaments and reward pool sizing. The $FVT MiniDAO is where policy is debated prior to ratification and material shifts in direction of the system occur.

1.15.3 The Path to Decentralisation

The finance.vote network leads with an ethos of maximal decentralisation. That is, we aim to disintermediate key network functionality throughout a phased deployment of the network, retaining control only where it is absolutely necessary. The network will be a permissionless system, creating an inclusive space where ideas can be negotiated and digital asset price discovery can take place through decentralised voting techniques.

The network is deployed in a three phase path towards decentralisation.

Obelisk (From network genesis)

This is the launch phase of the network during which a number of key components of the system are introduced to the users, including auction.vote and the vote markets.

It will not be possible from the outset to build these complex human interface features, without some degrees of centralisation.

In this phase a number of management keys exist that can, through human input, modify the rules of the game in such a way that they can be optimised for adoption based on feedback from the community.

For example, in the pre-TGE phase. Digital Identity Tokens will be issued via management key to engaged community members for the purposes of bootstrapping adoption.

Once the $FVT token is available, this process will be replaced by a purely permissionless perpetual auction system, whereby users burn $FVT to obtain DITs. This simple example cryptoeconomic disintermediation directly demonstrates the efficacy of utility tokens for facilitating decentralisation.

Throughout this phase, we create and optimise adversarial cryptoeconomic games that will stress test the permissionless governance structures of the network.

These include our voting technology, which utilises quadratic voting and therefore requires a sybil protection layer created by our identity system. Again, this will be tuned by human input, but once a clear parameter space for this component has been identified, it will transition to token holder vote.

The voting system in this phase is turned towards the markets in an attempt to understand the context in which the network is launched. The launch team will use the data produced by this to build tools and infrastructure for education and collective intelligence building. Informed consensus is a progressive long term process and it will be a responsibility of the network launch team to support in community understanding and knowledge building.

Throughout this phase the voting and auction systems will be optimised using our management keys so that we arrive at tunable parameters that can be transitioned to the main DAO, the DMF in phase 2. The launch team will aim to utilise a transformational leadership style, with a view towards showcasing effective practices for network maintenance and responding dynamically to token holder feedback. Our network culture is one of radical openness[16].

[16] "10 defining principles of radical openness - UNHCR Innovation." 28 Jul. 2016, https://www.unhcr.org/innovation/radical-openness/. *Accessed 17 Nov. 2020.*

Pyramid (Begins 6 months post genesis, end Q2 2021)

In this phase we launch our second layer governance system, which aims to further empower the community with decision making capability on the network.

In this phase, control of key system parameters are transitioned to the DMF and at an appropriate time the vote market management keys will be burned.

In this phase we use quadratic voting to turn dialogue into numbers.

Here we aggregate rough consensus so that it can be lensed into coherent decision making. This is a complex iterated process that will begin in this phase and will become a foundational paradigm of the network.
We do this in token specific miniDAOs. These are platforms where network level discussion is curated so that clear signals can be generated from the community.

Here social self organisation is promoted so that new system parameters and functionality can be discovered. In this phase, the finance.vote network core team are still in service to the users and will design, develop and deploy core technologies for the network and will expand the community of developers and users through the use of the DMF treasury.

Throughout the pyramid phase, the finance.vote reputation system will be tuned to ensure that a healthy mix of stake weighted and meritocratic consensus formation is used for key decision making. The system will be further hardened against sybil and collusion attacks[17].

Starship (Target 18 months post genesis)

In this phase, full responsibility for the network will be transferred to token holders.

All three initiating components; the vote markets, second layer governance and social trading will be deployed to mainnet and operating effectively. Each of these will have tunable parameters, with a history of effective decentralised decision making, with a clear trend towards systematic optimisation.

By this point there will have been a number of successful quadratic funding sequences utilising the DMF treasury and a growing decentralised development community will be building on the network with a new roadmap that is currently unknown to the launch team.

In this phase the crowd will be in full control and all management keys will be burned. Responsibility for future development is now with the community and network leaders will be elected through decentralised direct democracy.

[17] “On Collusion.” 3 Apr. 2019, https://vitalik.ca/general/2019/04/03/collusion.html. *Accessed 17 Nov. 2020.*

1.16 Token Economics

1.16.1 Design Philosophy

Decentralisation

The finance.vote network is aiming for maximal decentralisation.

That is, we will aim to be as decentralised as possible throughout the development trajectory. Necessarily, the network will launch with a pseudo-autocratic power structure, but this will progress towards decentralized direct democracy over time.

From the outset, founders will manage some keys that hold final decision making power for a period of time. These keys will be transferred to the crowd once an effective governance structure has emerged.

The Finance Vote Token ($FVT) makes this possible, it represents power in the system. The finance.vote ecosystem will pioneer a number of new voting systems that are designed for generating the progressive diffusion of power away from any central arbiter within the system.

Incentivised Action

The $FVT token model is designed to generate user action through rational financial decision making. Users will vote in the system if it makes sense for them to do so. In the world of crypto economics that means it makes financial sense.

All token systems must wrestle with the balance of distributing tokens to participants in order to incentivise adoption and network value dilution arriving from inflation. Adoption comes at a cost.

We balance the incentive dynamics in such a way that they generate adoption through a range of vectors targeted at different stakeholders, including market analysts, decision makers, workers and liquidity providers.

All of these add new tokens into the ecosystem from a starting point, but are distributed to users who bring value to the system. Ultimately, it is the responsibility of the network to balance these incentives against one another, managing the inflationary and deflationary dynamics of the system along with a range of other monetary policy decisions.

The inflationary dynamics of the system at this point are a direction that can be steered by the $FVT holders.

Responsive Governance

Good governance is responsive. It is a system of decision making that responds to the needs and desires of the participants effectively.

The system is designed in such a way that users can build an understanding of key parameters in the system that can be tuned or optimised for maximum engagement and healthy ecosystem growth.

Finance.vote will release a range of voting mechanics that build the reputation of stakeholders in the system. User voting power will be scaled through a mix of meritocratic validation of decision making history and token stake.

Those that have the power in the system will be those that have earned it through participation and high quality decision making.

Those users who are effective in other areas of the system will graduate to the main DAO, the DMF, which will make high level monetary policy decisions and decide how the treasury is spent.

Discourse

The key to good decision making is dialogue.

As the system develops we will find mechanisms to channel and focus discourse into decision making. Our second layer governance system will provide a space for crowd curated user dialogue that can tangibly influence not only our system but others too.

It will be a platform for content aggregation, curation and collective learning. As the ecosystem develops we will build an inclusive international community, which aims to maximise understandability of the system and optimise for involvement in governance decisions.

Utility

The Finance Vote Token will aggregate utility over time by integrating functionality determined by the needs of the system and the desires of the token holders.

The system will hold day one utility. From the moment the token is tradable it will be exchangable for a voting identity in the system and usable within our vote markets. None of the systems functionality will be accessible without an identity.

There are three core aspects of utility that the founders are committing to deploying during the bootstrapping phase of the network: vote markets, second layer governance and social trading. We believe this utility set is strong enough to build a sustainable network, however this token economics system provides a substantial treasury which is to be spent on funding open innovation that falls within the emerging shared design philosophy of the system.

Identity Tokens

Identity is a crucial component in voting systems. The approach used by finance.vote is to issue a decentralised identity token (DIT) to participants.

These will take the form of NFTs that are tradable if the user desires. The cost of an identity token is dynamic (depending on demand) but begins at 100 $FVT. Users must burn this amount in our identity distribution system to obtain a DIT.

This identity is linked to their voting history and their performance within the system. This will generate both a rank and a vote power budget $V, our internal cryptocurrency. They are pseudo anonymous but are a vector for building trust and reputation to participating accounts in the system.

Quadratic Voting

Quadratic voting will change the world. It broadens the parameter space on decision making and is an ideal mechanism for filtering preference. Finance.vote will utilise this framework to build it's governance system and seek to find effective mechanisms for channeling user action into productive network outcomes.

We utilise quadratic voting in a mechanism we call Semantic Ballot Voting. Users are provided a constantly replenishing budget of vote power ($V), which they distribute on votes through various mechanisms in the system. Typically this action will involve sorting some list of semantic items (token cash tags in the first instance) by preference, through distributing $V quadratically.

Education

A crucial component of the finance.vote ecosystem will be building understanding across our community members of the kinds of systems we are making decisions about. Good decisions arise through informed consensus.

Asymptotically, the best players in this game will be those who can read code. We will support the development of understanding through teaching and materials of how best to understand the cryptospace. The best researchers will win.

1.17 Token Metrics

The $FVT token has an initial generation amount of 1 billion tokens (1,000,000,000 $FVT).

These tokens are split into a set of tranches as follows:

1.17.1 Initial Distribution

20% of the tokens (200,000,000 $FVT) are to be distributed to early adopting participants through a series of distribution rounds.

The first wave have concluded and took the form of two private rounds:
Seed 6% (60,000,000 $FVT) tokens @ $0.007
Private 12% (120,000,000 $FVT) tokens @ $0.008

Those participants that obtain tokens in these rounds are subjected to cliff-linear vesting over a period of 5 months.

The final 2% of these tokens (20,000,000 $FVT) will be distributed by a decentralised auction mechanism (auction.vote), which will go on to form a key component of the finance. vote ecosystem (details of this mechanism will be released in a standalone post).

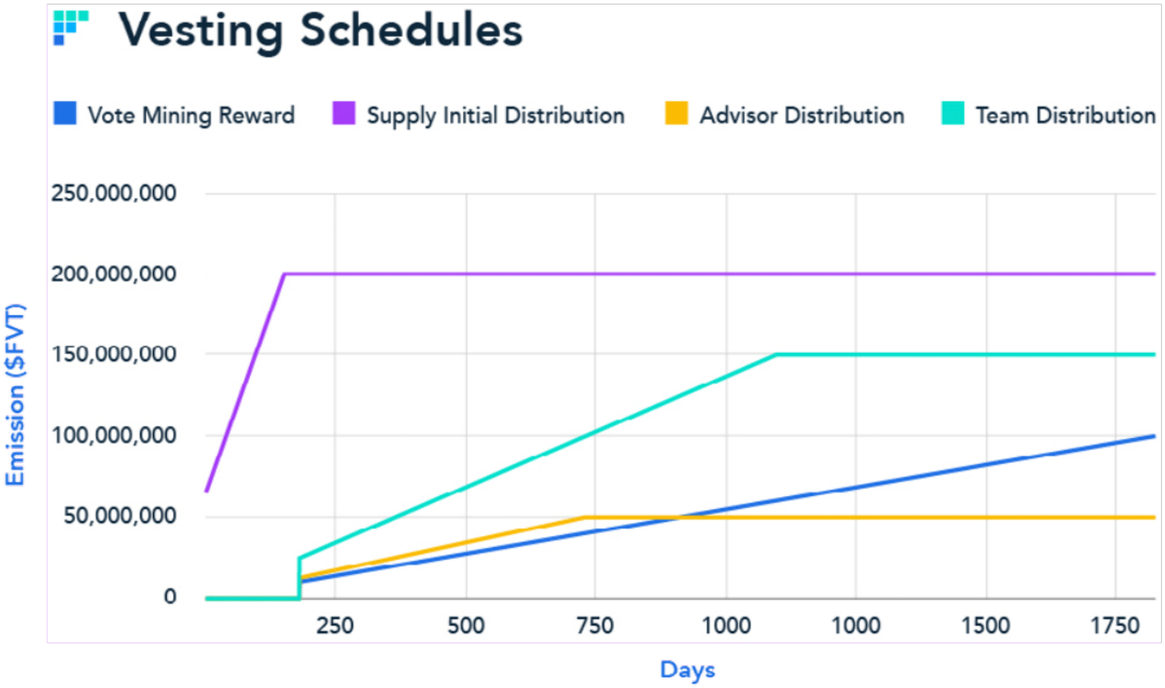
1.17.2 Team and Advisor Stakes

15% of the tokens are allocated to the founders of the network. These are released to the team using (6,3) year cliff-linear vesting i.e. vested for three years, with a 6 month cliff.

5% of the tokens are allocated to valuable strategic advisors. These are released individually to advisors using (6,2) year cliff-linear vesting.

1.17.3 Governance Incentive

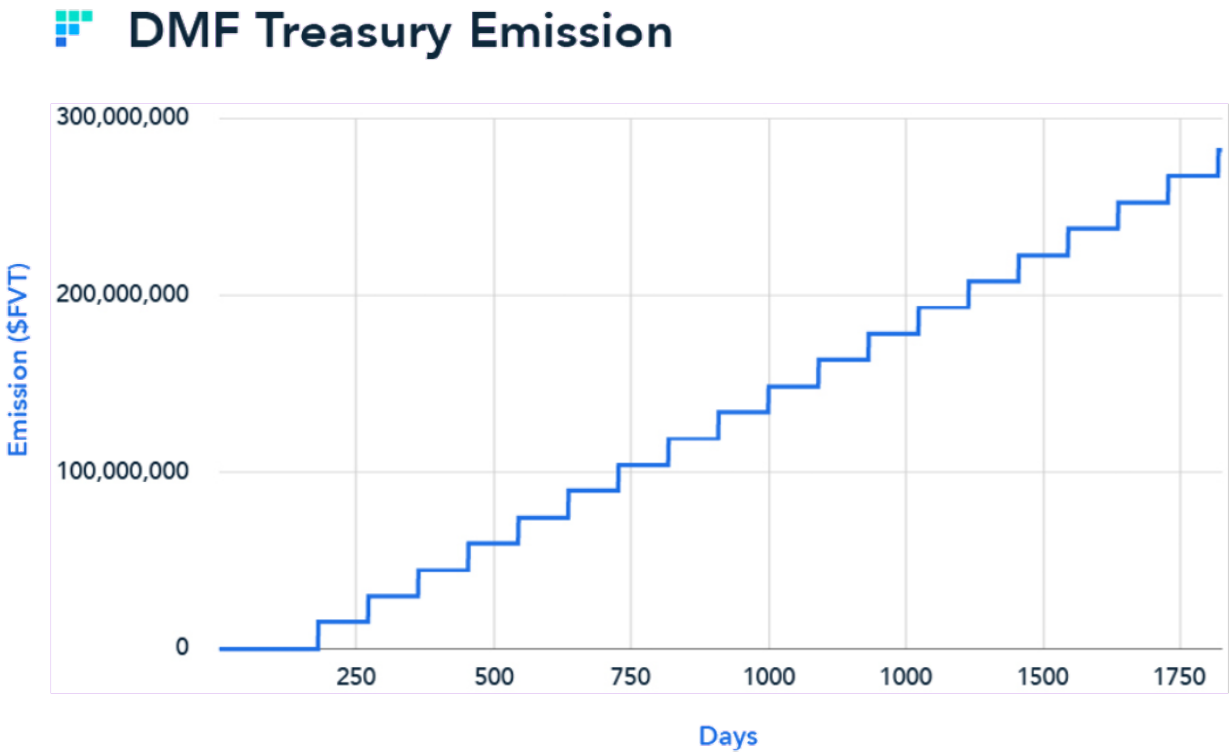
10% of the tokens are to be distributed to voters of the system over a (6,5) cliff-linear vesting schedule. These tokens will be distributed to users proportionally to the number of $V spent in the system associated with their identities. We call this process vote mining and it is one of a number of mechanisms designed to break voter apathy in blockchain governance.



1.17.4 The DMF Treasury

30% of the tokens (300,000,000 $FVT) will be distributed through (20,5) step vesting i.e. 20 tranches of 15,000,000 $FVT, in a lump sum for 5 years to the DMF treasury.

DMF funding tranches will be distributed via quadratic funding mechanisms to stakeholders who bid to do work for the network. If the DMF participants choose, a proportion of each tranche can be burned, providing further control of the monetary policy.



1.17.5 Liquidity Pool

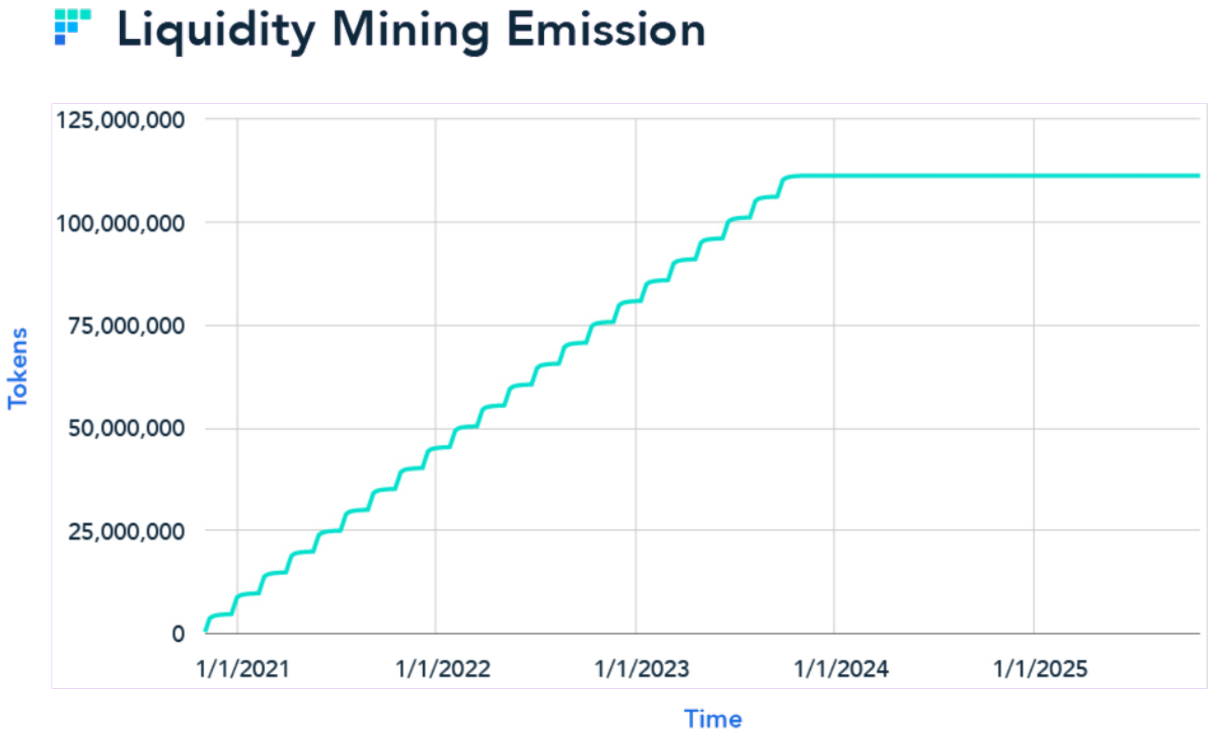
20% of the tokens (200,000,000 $FVT) will be utilised to bootstrap the liquidity of the finance.vote network.

The liquidity pool is split as follows:

2% of the tokens (20,000,000 $FVT) will be utilised as match liquidity in the initial distribution auction.

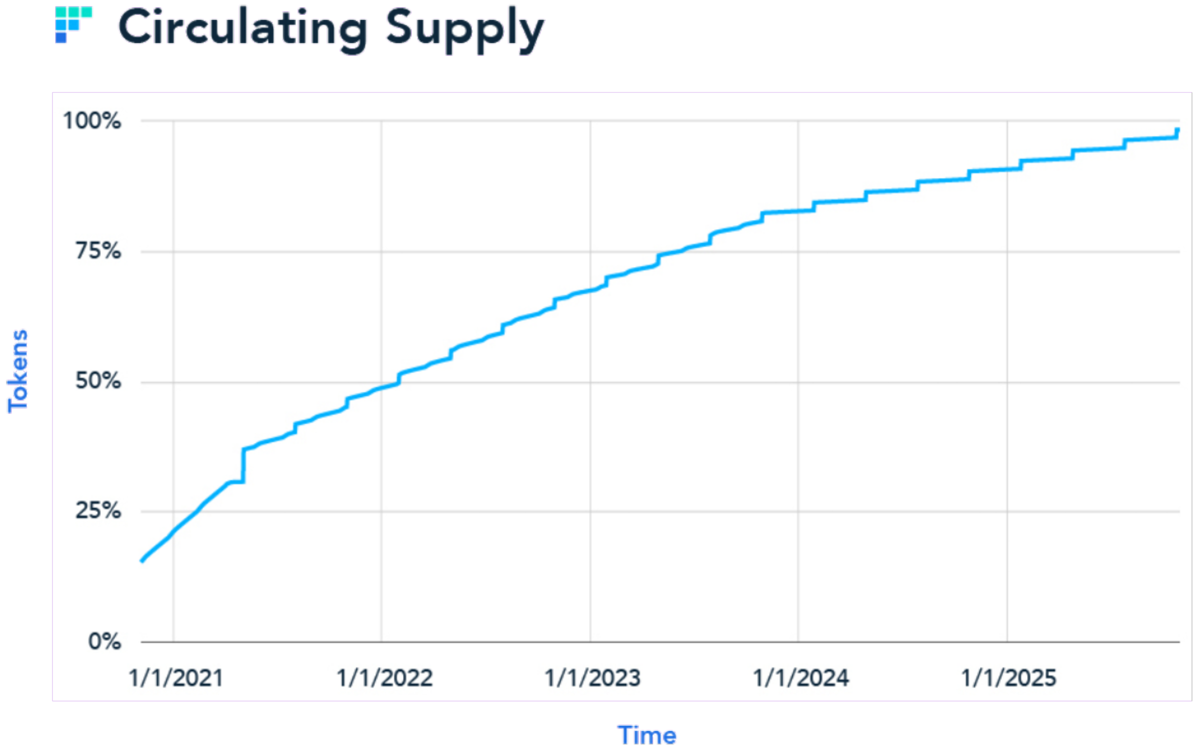
A minimum of 9% (90,000,000 $FVT) are allocated to liquidity miners in the DEX space. These will be distributed via a pulsed liquidity mining incentive scheme to holders of LP tokens in respective decentralised exchange pools.

9% of the tokens (90,000,000 $FVT) will be utilised to engage with the centralised exchange (CEX) space. These are unlocked from network launch, but will be utilised transparently so that users can understand monetary flows in the system.



1.17.6 Circulating Supply

The aggregated emission from these schedules provides us with the following emission curve represented as a percentage of total genesis supply.



1.17.7 Inflationary Mechanisms

The finance.vote core contracts hold the power to mint new tokens beyond the 1,000,000,000 $FVT at genesis. This is chosen to ensure that there is an open ended solution to incentivising adoption. In the first instance a small amount of additional inflation is used to seed liquidity for reward pools in our vote markets, starting @ 100,000 $FVT per week / per market.

1.17.8 Deflationary Mechanisms

In order for any user to take part in the system, they must first obtain a Digital Identity Token (DIT). In order to do this, users must obtain $FVT and burn at least 100 to take part. This introduces an adoption based deflationary dynamic to the system.

1.18 Emission Schedules and Monetary Policy

The emission schedule of a token system determines the inflation rate of a token economy.

It has recently become the trend to release entire token supplies in a matter months if not weeks in the recent DeFi boom. This has led to hyperinflation and short lived token economies.

We are not aiming for a multi generation store of value system, neither are we claiming to be “money” (not yet anyway). The Finance Vote Token ($FVT) is a utility token, which provides users with access to a governance system.

The governance system acts as a kind of crypto economic hub, which the users will ultimately control. $FVT is required to access the system, without it you will not be able to vote or participate in the governance decisions.

1.18.1 Cliff-linear Vesting

Cliff-linear vesting is a smart contract based vesting schedule that we have decided to add confidence to emission schedules.

The greatest threat to a token economy is a highly asymmetric token distribution, even worse if it is an unknown token distribution.

We have programmed a schedule, which releases a small proportion of tokens to network stakeholders in a lump sum, followed by a block-by-block distribution for a prolonged period of time.

It follows that:
Token Amount = (month of cliff from genesis, vesting period length in years)

$$TA = (m,y)$$

1.18.2 Vote Mining

A 100,000,000 $FVT(6,5) allocation has been granted to all voters in the system. This means that approximately 9,917,808 $FVT will be airdropped to voters 6 months after genesis and then continually at the same rate of 54,794 $FVT per day for 5 years.

This allocation will be steered by the community throughout the 5 year vesting period. At the moment it is only possible to vote in our vote markets, but eventually vote mining rewards will be flowed to both our miniDAOs and the DMF to break voter apathy.

1.18.3 Pulsed Liquidity Incentives

The provision of decentralised liquidity is a revolution. The dynamics of this activity are very early, however the dawn of the “agricultural revolution“ has introduced incentive dynamics to the liquidity provision process with demonstrable success.

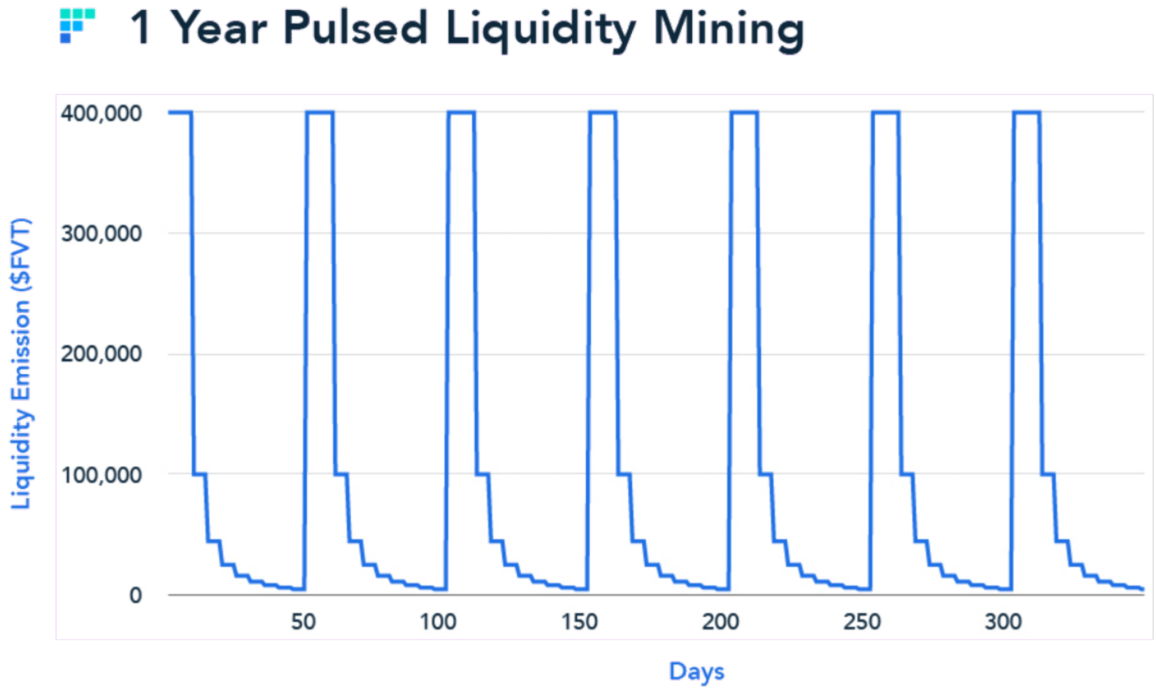
We have created a tuneable system that will ensure that we maintain effective capital efficiency of our trading pools for the lifecycle of the network development, aiming to optimise for trade volume, market depts, inflation cost through dynamic incentives.

Our approach to this is through adding a cyclical liquidity dynamic, with tunable parameters, that include pulse height, quadratic decay rate and a pulse width.

The result is a liquidity cycle producing periods where: yield is high, yield is low and a range of levels in between that drop quadratically between pulses.

This provides LPs with short time horizon trading strategies with an opportunity to optimise their yield through complex farming techniques, or users with a longer term strategies deciding to stay the pool across cycles.

The management of these parameters will be tuneable in the DMF and offer the possibility to minimize the impact of divergent loss by optimising for pool growth through governance activity.



1.18.4 Summary

The finance.vote token economy introduces a range of dynamic crypto economic monetary policy ideas with the view of creating a sustainable crypto economic hub. The system uses collective decision making and voting technology to create a participatory token economy driven by the desire to understand the cryptospace itself.

1.19 Blockchain

1.19.1 What is it?

Blockchain is a record-keeping technology standing–among others–behind cryptocurrency networks. To understand that better, we can say that blockchain is a specific type of database storing data in blocks which are then chained together. In this case, 'block' means a group of information.

The cryptocurrency world is not the only place where blockchain is being used though. It is definitely the most obvious use case these days, but blockchain is also being used with legal contracts, product inventories, and any other place where a trustworthy, secure and fast system may be needed.

For example:

- medical data
- personal identity security
- voting mechanisms
- real estate processing platforms
- and many more

1.19.2 Types of blockchain networks

Over the years, blockchain has been evolving and new user needs have come up. Depending on how a blockchain is configured, there are several business cases it can fulfill.

Public blockchain

In a public blockchain, anyone is free to join and participate in the core activities of the blockchain network. Anyone can read, write, and audit the ongoing activities on the public blockchain network, which helps a public blockchain maintain its self-governed nature. Public blockchains can be fully decentralized and democratized, and their transactions can stay authority-free.

Private blockchain

Private blockchain is useful whenever there is a need to verify participants, allow entrance only by authentic and verified invitations, and to decide who can make transactions or authenticate blockchain transactions and changes.

The ledger in this solution is a secure database based on cryptographic concepts. It is not decentralized though since it is closed; only allowed users can look into it.
Within a private blockchain, only the owner or the operator has the right to override, edit, or delete the necessary entries on the blockchain as required. Also, the consensus protocol works differently (for more information see the 'How is it secured?' section below); its execution is dependent on specific users. Further, mining rights and rewards are decided in advance.
This type of blockchain is most commonly used in private business.

Permissioned blockchain

Permissioned blockchain is a mixed type of public and private blockchain. It has many customization options. For example, there could be a customized blockchain where anyone could join but only a certain number of activities would be allowed to be performed by a user.
This is also known as semi-private, consortium, or federated blockchain. The consensus process within this type of network is controlled by a pre-selected set of nodes.

1.19.3 How does it work?

Regardless of the type of blockchain network used, there is always the possibility of making transactions on it. No matter what type of transaction it is, once it is made it gets recorded and its authenticity must be verified by the blockchain network. This is done by computers within the blockchain that confirm that the details of the transaction are correct. After successful validation, a transaction is added to a blockchain block.

Imagine for example you're working in a dairy and your job is to fill and cap bottles of milk. Whenever a bottle is filled up, you need to cap it and put it in the box with the other filled up bottles.

Similarly, a block has a defined capacity too. Whenever it fills up, it gets closed and chained to the previous block. In this way, every entry and all information is permanently saved to the block and stored. Consequently, we can build a transparent database and do not have to worry about any unwanted data changes since that would be impossible without leaving a trace (see below for further explanation about possible inconsistency within blocks and possible changes within a chain).

When a new block is added to the chain it gets a timestamp, a unique self-hash, and a unique hash of the previous block. Timestamps are responsible for chaining blocks together. As blocks are set in a chronological order on the timeline of a blockchain life, timestamps become a specific and immutable connection between blocks. Hash, in simple words, is just a mathematical translation of the content inside a block saved in the form of a string of numbers and letters.

1.19.4 How is it computed?

Imagine there is a very long and complex mathematical equation. You have to work on one part of the equation at once and solve that step by step till the end. To save some time, you could split the equation up and give out parts of it to several people to solve simultaneously. The only thing left would be to sum up all of the resolved parts together. Voila, you've saved some energy and time.

Transactions on the blockchain network are computed in a similar fashion. Every block is filled with data represented by code. That code needs to be computed and solved just like the

mathematical equation. Thus, splitting all of the computation into several machines is more effective than using just one computing engine. That is of course possible, but less effective. | | In the real world of blockchain, these kinds of computations are very numerous and spread across a network of nodes. The amount of transactions happening across a blockchain are so enormous that every node regularly computes more than one equation for more than one transaction.

1.19.5 Transparency

What exactly does it mean when we say a database is 'transparent'? Many databases within the world of blockchain are set to public, which means that anyone can join the network and view all of the transactions on that network. To do this, a user must be a part of the blockchain network and use a blockchain explorer that allows them to see all transactions. Within a blockchain, there is no way of doing anything to the blockchain without it being recorded. At the same time, any sensitive data such as identifying information is encrypted and safe. They are still shown and called a 'public key'.

For example, a virtual wallet address is just encrypted information which is unreadable to humans and useless to a system that does not have a decryption key.

1.19.6 Where is it stored?

Blockchain like every other database-like structure needs to be stored. Blockchains that represent small product inventories can be stored on a group of computers existing in one place since they do not have to store thousands of transactions and handle large network traffic.

With bigger systems such as those for cryptocurrency, it is hard to store all of the data in one place even with a large amount of computers. To solve this problem, blockchain is spread all over the world and stored on a huge number of machines, which in blockchain lingo are called 'nodes'. That is why a blockchain storage system is commonly known as a decentralized storage system.

Each node within a network has a full record of the data that has been stored on the blockchain since its inception. But, what happens if one of the nodes has an error in its data you may ask? Luckily, the blockchain can use thousands of other nodes as a reference point to get the data correct again. This is yet another proof of the irreversibility and immutability of a blockchain system because no node can alter any data. Although, even if one tries, other nodes cross-referencing eachother would easily pinpoint the node with the incorrect information.

1.19.7 How is it secured?

New blocks are always stored linearly and chronologically. Every block has a certain position on the chain called 'height'.

It was mentioned above that chains are immutable but this is not an entirely accurate statement. Changes are possible only if **a majority in a chain agree on the change**. This is called 'reaching consensus' which is used for the purpose of security.

Why is reaching consensus so important? As pointed out before, when a new block is added to the chain it gets a timestamp, a unique self-hash, and a unique hash of the previous block. When any changes are being made, new hashes and timestamps are created. Without a reaching consensus method, changes could easily be overlooked. This could lead to conflicts between blocks or at worse, malicious changes being added to a blockchain.

Does this mean that an attack on the blockchain is impossible–an attack where someone takes majority control of the chain and adds inaccurate entries for gain? No. There is a possible scenario where a hacker or hackers could overpower the chain by gaining majority control of the chain. However, this kind of attack is incredibly unprofitable since handling the majority of a chain would be incredibly expensive, rendering an attack virtually pointless. In this way, participating in the network is far more economically viable than trying to attack or overpower it.

1.20 Consensus Algorithms

Within a blockchain network there are several actions that can be undertaken and every one of them has to be validated. But what does that mean exactly?

Validation within a blockchain network is based on proofs which are widely known as **Consensus Algorithms**. Consensus stands behind achieving agreement on such matters as confirming transactions or producing new blocks on a chain. Whenever one of those actions are requested, a consensus protocol is triggered. To reach a consensus, a majority of nodes have to agree on something. People often reach a consensus easily by voting on something, and when a majority of votes are the same, a consensus is reached. In the computer world, things are a bit different since voting would be too vulnerable to attack and too easy to be tampered. That's why consensus algorithms were brought into blockchain technology.

There are plenty of different consensus algorithms. Some of the most common ones include Proof of Work, Proof of Authority, Proof of Stake, and Proof of Burn.

Right now, Proof of Work is the most commonly used consensus algorithm, but Ethereum has plans to use Proof of Stake in the future. It's worth mentioning that Proof of Work (PoS) and Proof of Stake (PoS) are the two best known sybil deterrence mechanisms. This is why they are the most common mechanisms within the cryptocurrency world.

1.20.1 Proof of Work

Given its name, understanding this algorithm is fairly straightforward - you've got to prove that you've done some work. But how do you exactly prove you've done the work?

Proof of Work involves all the nodes within a network doing some work, and that means every computer has to take part in solving a complex mathematical problem. In simpler words, work means computer calculations.

As mentioned before though, these mathematical problems to solve aren't based on the mathematical methods we know from school. Solving one can be done only by trial and error, and the odds of solving the problem are about 1 in 5.9 trillion. This requires a lot of computing power that leads to large energy consumption. Solving the problem is known as mining and nodes taking part in it are miners. So to encourage miners to mine, this huge consumption of energy must be profitable. And it is.

Because one computer would be solving this kind of problem for years, a whole chain is working together but not every node will win. The winner is the first node that can solve the problem. This node then shares the requested action to the chain and gains freshly minted currency as a reward. Mining allows you to earn cryptocurrency without buying it.

To understand that better, let's imagine a mine. To begin with, a miner doesn't know where the gems are located. Working alone would require vast amounts of time, digging one spot at a time and having no certainty of finding anyting. With the help of a team though, mining would go much faster as a bunch of people would be digging multiple areas at the same time. The only difference is that in crypto mining, the gem or mining reward of freshly minted cryptocurrenty is awarded to the first miner who solves the problem.

Speaking more technically, let's first establish that **the cryptocurrency network sets a target hash**. We'll need this information later.
The mathematical problem to solve can be found in block's header, which contains:

- the block version number
- a timestamp
- the hash used in the previous block
- the hash of the Merkle Root
- the nonce
- the target hash

All the above creates a block hash, which in PoW each node takes and calculates. Since a given set of data can only generate one hash, miners alter the input by adding an integer called a nonce. When one node reaches the expected calculated value (finds the correct nonce), it will broadcast the block to the other nodes and the block will be added to the chain when a majority of nodes confirm the correctnesss of the hast value.

What is a target hash?

A target hash is a numeric value that is used to determine how hard it is for the miners to solve the block.

What is a nonce?

A nonce is an abbreviation for “number only used once”. It is one of the block header components and the number that blockchain miners are trying to solve. The nonce is the value that miners can alter to create different permutations and generate a correct hash.

The purpose of proof-of-work algorithms is to deter the manipulation of data. Unfortunately, these algorithms use large amounts of energy in order to run such enormous computations. This has raised concerns over mining’s ecological footprint. Overtime though, mining will continue to trend towards greener solutions as they reduce energy costs and thus increase miners’ profits.

1.20.2 Proof of Stake

In contrast to proof-of-work algorithms, proof-of-stake algorithms are less energy intensive, which is one of the arguments for switching entirely to a PoS in the future. It is energy-saving because it doesn’t require thousands of nodes working together on finding the nonce. Instead, the algorithm randomly assigns one node to add a block to the chain and assigns several other nodes to confirm the addition.

What’s the catch?

To become a proof-of-stake algorithm validator, a user must stake some cryptocurrency of the very blockchain that it is validating. For the Ethereum network, the staking amount is 32 ETH that can be provided by one user or by a pool of users. Such staking is both a kind of collateral and incentive. Naturally users don’t want to lose their money, so staking funds into a pool makes users want to protect it from destruction, which is the consequence of validating a malicious block.

Proof of stake doesn’t make validators compete, so only a chosen node, known as “the main validator”, works to create a block. Then, an attesting group validates this proposed block and decides whether the block is malicious or not. Again, every node attesting a block either takes credit for a good attestation or suffers the consequences of a bad one.

For a deeper insight into this topic, checkout ethereum docs.

1.21 Networks

Networks are Ethereum environments which are used either for actual transactions or for testing purposes, depending on what kind of network is chosen. To begin, a user must first have a virtual wallet. Yes, this applies to developers and testers as well.

Networks subdivide into two main categories: Public and Private.

1.21.1 Public Networks

Public networks can be used by anyone in the world as long as they have an internet connection. Anyone can read and create transactions, and also validate executing ones. Validation is reached by a consensus of nodes.

Public networks subdivide into Mainnet and Testnet.

MainNet

MainNet is where actual transactions occur on the distributed ledger. Real exchanges of real tokens are happening here. Within MainNet, any user of any node connected to the network can read all the contract code and data. The only limitation that is implemented on MainNet is the permission check in contract code, which determines which accounts can update the state of a contract.

Testnet

Testnet, as the name suggests, are networks used to test functionalities and monitor the blockchain network performance. They are public, but currency used in the transactions are valueless.

This network allows someone to test the functionality of some code that it is working the way it should work, but without any risk of losing money or breaking the main blockchain network. Testnet can be likened to a staging server where software, websites, and services are tested. The possibility of reusing the same test files allows for an accurate comparison between various test runs and thus catching potential errors and network failures. In addition, having tested out a whole project makes MainNet deployments much faster.

Some blockchains even provide testing methodologies, tools, and certifications to accurately test complex networks at scale to help increase productivity and infallibility.

In the case of creating a dApp that integrates with an existing smart contract(s), Testnet becomes a very convenient way of testing since most projects working on MainNet already have copies deployed to testnets. And it's important to remember that testing any contract code before deploying to the MainNet is crucial.

Types of testnets:

Görli

A proof-of-authority testnet that works across clients.

Kovan

A proof-of-authority testnet for those running OpenEthereum clients.

Rinkeby

A proof-of-authority testnet for running Geth client.

Ropsten

A proof-of-work testnet. This means it's the best like-for-like representation of Ethereum.

Testnet Faucets

As mentioned before, currency used in testnets have no real value, so there are no markets for testnet currency. To get some currency for testing purposes, one can retrive some from a testnet network faucet. Most faucets are webapps where after entering a virtual wallet address, a request is made and then test current is sent to the wallet.

1.21.2 Private Networks

Private networks are isolated networks since their nodes are not connected to any public network, neither MainNet nor testnet.

Development Networks

Development networks may be described as a local blockchain instance to test a dApp. It's similar to creating a local server on a computer for web development. Thanks to this, iteration may be much faster than it would on a public testnet.

Within our project we use a dedicated tool to assist with this–Ganache CLI along with Truffle Development Environment.

Consortium Networks

Consortium networks are also known as semi-private, federated, or permissioned networks. The consensus process is controlled by a pre-defined set of nodes that are trusted. For example, there could be a consortium of 10 different individual institutions (each of them operating one node), and of those 10 nodes, 6 must sign every block to validate it.

To better understand, we may think of a public network as the public internet and the consortium network as a private intranet.

1.21.3 Is it possible to send tokens between networks?

IN BUILD.

1.21.4 How do networks sync?

IN BUILD.

1.22 Merkle Tree

A Merkle tree is a structure used in blockchain that serves to encode data more efficiently and securely. It is a collision-resistant hash function, which takes in n-inputs and outputs a Merkle root hash. Collision-resistant hash function make it impossible to generate two identical outputs to two different inputs, which in theoretical terms could be understand as a following formula:

> **H(x)H(x)**
> where H(x) stands for "hash generated from input x".

It is worth mentioning that any generated output is assigned to a concrete input. For example, H(x) will always be y, while H(x') will always be y'. We are getting a little ahead of ourselves, but consider the following:

- function takes in input x,
- it checks whether output H(x) already exists,
  - If it exists, it returns that output,
  - If not, it creates a new output, associates it to this input, and returns that out.

Merkle trees are also known as "binary hash trees".

1.22.1 What is it used for?

Hash trees are useful for verifying data stored, handled, and transferred in and between computers. They are very helpful in ensuring that received data (e.g. from other peers in a peer-to-peer network) are undamaged and unaltered.

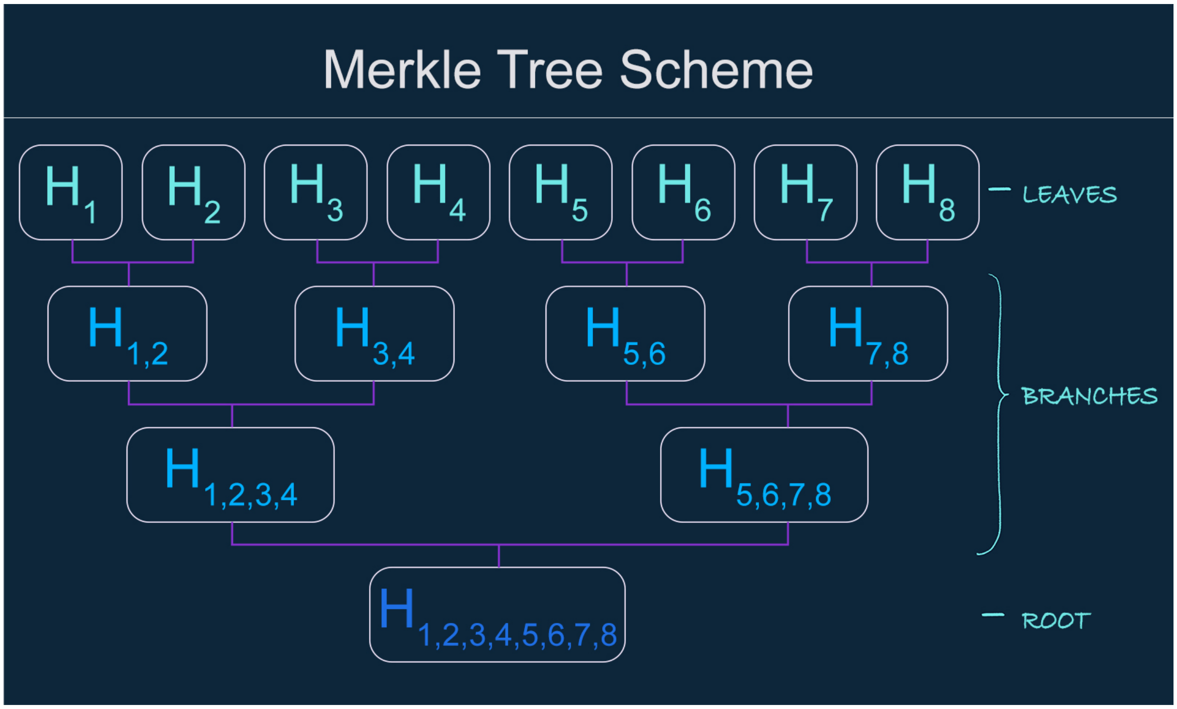
The biggest advantage of Merkle tree usage is that a **prover** (who has a large set of data) can convince a **verifier** (who has access to the set's Merkle root hash), that a piece of data (e.g. a single transaction) is in that set just by giving the verifier the **Merkle proof**. This kind of action is secure and trustworthy, not to mention easy to verify since a verifier doesn't have to download the whole blockchain to confirm a single transaction or single piece of data.

A Merkle proof is a small part of a Merkle tree.

To sum up, Merkle proofs make it possible to verify that a particular piece of data is a part of a hash tree and/or wasn't in any way altered.

1.22.2 How does it work?

Let's start by explaining the graphic below:



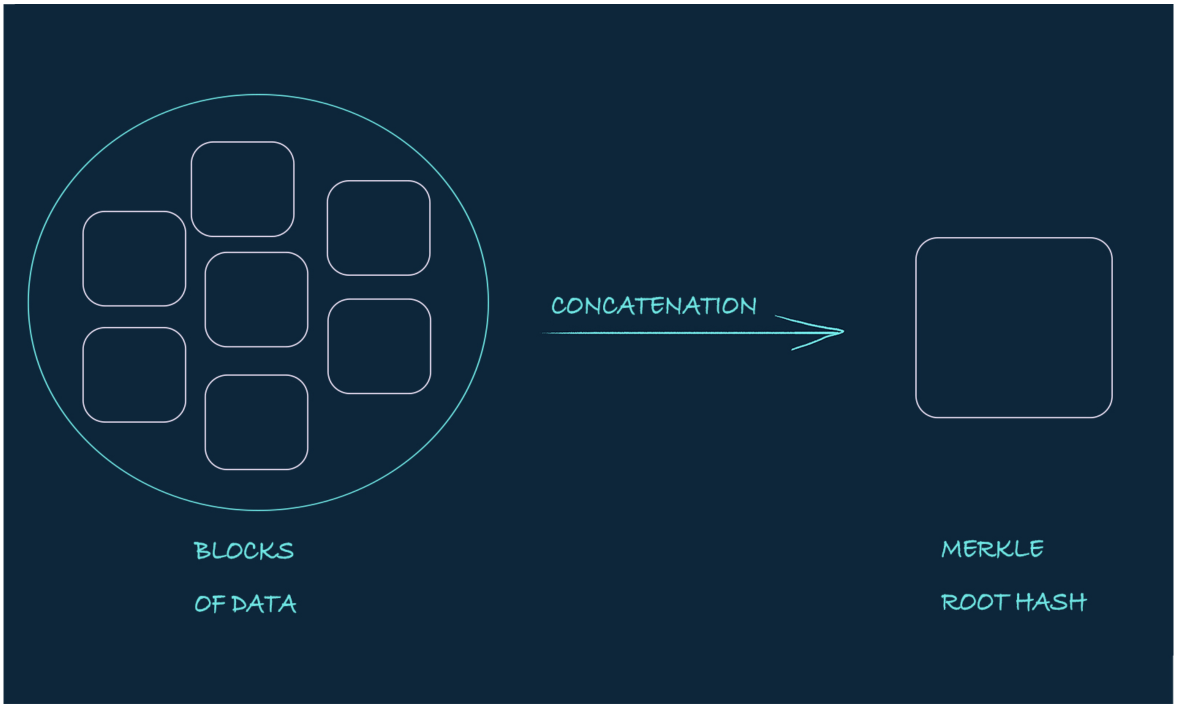
This Merkle tree scheme depicts the main work scheme within a Merkle tree. As you can see, there are three main parts: **leaves**, **branches**, and **root**.

Leaves are hashes of blocks of data. That's the first level of hashing. Next, every two hashes are concatenated and hashed again into one hash till there are only two of them left. This level of hashing is called branches, and it involves every level where concatenation takes part. The last level is the **root** where the last two hashes are concatenated one last time and hashed, which creates the **Merkle root hash** in the end .

If the number of blocks on any level of a Merkle tree are odd, then the left out hash is duplicated and hashed with itself.

A Merkle tree takes an arbitrarily-sized input and then outputs fixed-size hashes. It works similar to **"compressing"**, which is desirable since verification should be fast, secure, and efficient, and this can be reached thanks to a small size of data.

To simplify this, we can say that a Merkle root hash is a concatenated hash value of all block hashes:



Note: Notice that instead of keeping n hashes, you could just keep one root hash created from all the previous ones. This is very efficient and saves some space – that's why we can compare it to a compression.

1.23 Yield farming

Important definitions:

- **APY** - Annual Percentage Yield, amount of yield you would get on your principal annualised
- **APR** - Annual Percentage Rate, it's similar to APY but it doesn't consider compound interest
- **LP** - Liquidity Providers, users locking their funds into Liquidity Pool
- **Liquidity Pool** - a smart contract containing funds, powering a marketplace where users can exchange, borrow or lend tokens. Click here for more information
- **Liquidity mining** - process of distributing LP tokens to Liquidity Providers
- **Collateral** - a borrower's pledge of specific property to a lender, to secure repayment of a loan

In a nutshell, yield farming is a reward scheme. Farmers(users) can lock up their funds in liquidity pools and gain fixed or variable interest in exchange.

Analogy

Imagine you are a real farmer. You sow grains in your field and wait for them to grow, then you reap the crop or yield.

Now compare it to our situation - the farmer is the user, grains are tokens, the field is a liquidity pool, and yield is profit.

Users put tokens into a liquidity pool, wait for the interest to grow, and then gain profit.

We can define yield farming as lending cryptocurrency via the Ethereum network. It is like being a bank, with the difference being that the lender doesn't meet the borrower and the specific borrower doesn't return funds to the specific lender.

Yield farming is based on liquidity pools which are smart contracts holding funds.

Click here for more information on liquidity pools.

Important: Investing in cryptocurrency is not yield farming.

1.23.1 Why do we even need yield farming?

Of course it would be much easier to just put your crypto in a wallet and see what happens. You could make money when your tokens go up in value and wouldn't necessarily need yeild farming to make profit. This is also a much easier route for those who are new to cryptocurrencies.

Stepping into yield farming is highly risky but also highly profitable. You've probably heard this a thousand times, but high risk - high reward **Thus, yield farming is recommended for more experienced users, who know what is what.** But once you feel that you are ready to step in, you should be aware of some basics to yield farming.

The main idea behind yield farming is to make a profit, of course. Simply lending funds to the market won't bring a huge profit, unless the pool is very deep and a user's contribution is very large since commission from lending funds are often below 1%.

Now imagine that you have some extra money and you would like to deposit it into a savings account. It is quite obvious that you would like to earn as much interest as possible, so you look for an account with the highest APY. APY along with APR are indexes used in cryptoworld as well. They represent rates of returns and help you decide whether a strategy is profitable or not.

With traditional savings accounts, APY fluctuates. They are typically somewhere around 0.1%, or if it is a very lucrative offer, then it could be somewhere around 2-3% at best.

And this is where a big difference with yield farming emerges. Compared to savings accounts, yield farming has the potential for insanely high returns like 100% APY. Though, such returns are associated with higher risk and require some tactics to grasp.

Let's discuss then, how can users make significant profit from yield farming?

The key word is: **strategies**.

1.23.2 Strategies

By strategies, we mean moving funds between different DeFi protocols. It is either moving tokens earned from providing liquidity to other pools or by burning some earned tokens to receive back your locked funds and move them. This is really up to the user, since it requires a lot of market analyse, good intuition, and a hint of favourable wind.

The most profitable strategies usually require moving between at lest a few DeFi protocols (e.g. yield.vote, Uniswap, Compound, Curve, etc.).

Analogy

Moving funds between different protocols or swapping coins could be likened to a crop rotation. That's a farmer's practice of growing a series of different types of crops in the same area on rotation, which maximises yield as a result.

You probably know by now that there are many different DeFi protocols, and they offer different actions to users like lending and borrowing, supplying to liquidity pools, staking, and more. As we mentioned before, strategies are based on moving between different DeFi protocols. To earn a profit, a user can lend funds to a lending protocol, borrow some funds and trade them right away, or stake LP tokens. A user could combine these strategies as well to maximise profits.

Possible strategies::

- Supplying to liquidity pools: You can provide liquidity to a liquidity pool and gain commissions from token swaps that happen in that pool.
- Staking LP tokens: Some protocols allow you to stake LP tokens and reward you in their own LP tokens, which could be further stake in yet another protocol. To put this in other words, you could stake funds in pool A and get Atokens in return, then stake Atokens in pool B and get Btokens in return and so on. This way, you get commissions from different pools at the same time.
- Lending and borrowing: You can supply funds to a lending protocol for the purpose of earning a percentage on their capital or borrow from a lending protocol to leverage (i.e. borrowing funds to invest).

Caution: Please remember that yield farming strategies could potentially become unprofitable in a matter of days or even hours. Keep an eye on your locked funds and react quickly.

1.23.3 Liquidity mining

We will only briefly cover this topic. People often say that yield farming and liquidity mining are the same thing. Actually, they differ, but it's safe to say that both liquidity mining and yield farming have the same purpose - to maximise returns by earning tokens. Also, both operate within the DeFi sector.

Liquidity mining is the process of gaining new LP tokens operating on the Proof-of-Work algorithm, while yield farming uses different DeFi apps, including liquidity mining ones.

1.23.4 Leverage

Leverage is a common yield farming strategy used to increase profit. Users borrow external liquidity and add it to their liquidity for the purpose of increasing potential returns. This way it helps increase a user's share in a pool and LP token amounts as well.

Let's see this over an example:

Example

When Carl provided liquidity into the pool (for more information, see the example: provide Liquidity) he added 2000 USD, but he wasn't satisfied with his percentage share in the pool. So, he borrowed 1 ETH for 1000 USD and added it into the pool, now making it 2 ETH and 2000 FVT.

Thanks to this move he now has a larger share in the pool than before and thus gains higher commissions.

Collateral

Whenever you're borrowing assets, you'll have to provide **collateral**, which covers your loan and acts as insurance for it. In the case where a borrower fails to pay back a loan, collateralization serves as a compensation to the lender where the lender then keeps the collateral.

Depending on the lending protocol, there are different rules for collateral rates. For example, some lenders require little to no collateral, while others can require up to 400% of the lending value.

It's worth mentioning though that over-collateralization helps diminish the risk of severe market crashes and also saves the borrower from liquidation of its collateral on the open market.

Liquidation

Liquidtaion happens when the collateralization ratio (collateral:lended value) drops below a certain threshold, which means when a user's collateral is insufficient to cover the amount of their loan.

There are some methods to avoid liquidtaion:

- over-collateralize to prevent the collateralization ratio from dropping below the threshold amount
- use less volatile assets like stablecoins (the more volatile the asset is, the higher are the chances of liquidation)

1.23.5 Risks of yield farming

Again, yield farming can be very viable, however, the most lucrative strategies come with extensive strategy planning. Let's introduce some risks of yield farming:

1. **Liquidation**. As mentioned above, liquidation causes losing collateral.
2. **Asset losses caused by smart contract bugs**. Yield farming relies on smart contracts which, like any software, is exposed to potential attacks and suffer from random errors. Thankfully, doesn't occur frequently but users should be aware of these kinds of potential risks.
3. **Value of tokens changing**. The desirable outcome is for token value to increase, but sometimes it is the other way around which leads to fund loss.
4. **Block malfunction**. Whenever a block experiences a malfunction, the entire ecosystem suffers. Within the ecosystem, every block is connected with one another. This idea helps make it permissionless, but it demands collective responsibility.

1.23.6 Is Yield Farming Safe?

Yes and no.

If you come into yield farming without understanding its mechanisms, it will definitely backfire on you. Learning any form of investment takes time, effort, and substantial research to understand concepts and how markets operate.

Yield farming is definitely a dynamically changing market, with a lot to offer and a lot to loose. Understanding the terms of smart contracts on your own, rather than relying on third-parties, is a good idea.

Knowing these basics should help you to make better decisions with your funds within the cryptoworld.

1.24 Liquidity Pool

1.24.1 Abstract

A liquidity pool is a virtual container of funds deposited (locked) on a smart contract by liquidity providers. In exchange, liquidity providers receive LP tokens representing their share of the pool in proportion to how much liquidity they supplied to the pool.

1.24.2 Context

With standard cryptocurrency exchanges or traditional stock markets, trading is based on the order book model. In this model, buyers and sellers come together and place their orders. To make a transaction, a seller and buyer have to reach an agreement on the price of a transaction. This often results in the seller lowering their price and the buyer raising their bid, and this naturally can create a space of misunderstanding and disagreement.

One way to avoid this kind of situation is by engaging market makers, who are always willing to buy or sell assets. With market makers, liquidity can be provided to the market, and users can always make trades without having to wait for different offers to come.

Unfortunately, implementing this solution into the decentralized world won't work so well because it would be slow, expensive, and unpleasant to the users. The need for something more fitting to this issue brought liquidity pools to life.

1.24.3 Setting up a pool

A Liquidity Pool in its most basic form holds two tokens and each pool creates a new market for that particular pair of tokens. Within yield.vote, we use FVT/ETH paired token pools.

When a pool is created, the balance of both tokens are 0. For the pool to begin working, someone has to seed a pool with an initial deposit for **each token**. This someone is the first provider, who sets initial price of the pool and initial ratio between tokens.

Let's see an example:

Example

Carl wants to deposit funds into the ETH/FVT pool. The market algorithm obligates that both values of tokens must be equal. Carl then provides 1 ETH worth 1000 USD and 1000 FVT worth 1000 USD for a total value of 2000 USD.

Carl was providing liquidity into a pool that already existed, and even though he wasn't the one who made the initial deposit, he still had to follow the rule of providing equal token values into the pool.

Yes, we know that what are you thinking right now is: **but why?**

The initial price of deposit tokens is set in the moment of setting. That means the ratio between FVT and ETH is taken from current global market price. Obviously, because prices there are dynamic, the value of ETH or FVT will certainly change by the next day, hour, or even minute which creates an arbitrage opportunity.

Example

Let's stay with Carl a little longer. He provided liquidity into the pool but now the market price of ETH has changed, which created an arbitrage opportunity. Because of that, arbitrage traders will buy ETH in exchange for FVT until the pool ratio reflects the current market price.

Note: Remember that the ratio of the pool changes through LP deposits or trades, and is strictly connected to the amount of tokens in the pool. Also, the pool ratio determines the price of the assets in the pool.

You are probably wondering why a user would need to put both tokens into a pool?

To provide liquidity in a pool, there must be enough of both tokens present. That is ensured by a constant product maker algorithm. The formula is depicted below:

$$x \times y = k$$

x, y - reserves of tokens A and B
k - constant

Thank to this mathematical check, we can be sure that a pool can always provide liquidity regardless of how large a trade is. The main reason for this is that the algorithm asymptotically increases the price of the token as the desired quantity increases. Trying to buy all the tokens from the pool then a tremendous, and to be honest, impossible task.

Imagine you would like to withdraw ETH from the pool. Let's say this would be *token A*. When you take out of the pool some amount of one token, the balance breaks and the constant *k* changes. To prevent that, you would have to put some FVT (*token B*) into the pool to the maintain balance. As parameter *x* changes, *y* changes as well.

It's worth mentioning that the k parameter will change anyways due to transaction fees. This has a small impact on the k value, but this shouldn't concern the user since the k parameter needs to be maintained **before fees**.

1.24.4 Liquidity Pool Tokens

As mentioned in the beginning, liquidity providers receive LP tokens representing their share in a pool in proportion to how much liquidity they supplied to the pool.

To encourage users to become liquidity providers, every pool charges transaction fee on trades using the liquidity provided that is distributed to LPs upon completion of a trade. Fees vary depending on the pool, but the principle remains the same that transaction fees are distributed proportionally to all LP token holders in the pool.

Since LP tokens are themselves tradable assets, their holders may sell, transfer, or use them in any other way they see fit.

In purpose of retrieving underlying liquidity (plus any fees accrued), users must "burn" their liquidity.

1.24.5 Price impact

As the ratio of tokens in a pool dictates the prices of assets, then buying ETH from the pool will decrease amount of it in the pool, while increasing the amount of FVT in the pool. This transaction would then reflect in the prices of ETH and FVT.

The current market price in a pool only shows the marginal token price, while in practice, a trader buys or sells more than one token at once. And, as the ratio of tokens dictates prices, every token taken out or put into the pool will affect the price of the other token.

This difference between the current market price and the execution price is called price impact. The change of prices depends on the size of the trade in proportion to the size of the pool. The bigger the pool is, the lesser the price impact is.

Price impact is a function of the trade size relative to the liquidity pool size.

Note: It's worth mentioning that large (or deeper) pools can handle bigger trades without significant price impacts.

Example

> Imagine you want to buy **3 apples**. You go to the local fruit market and there are only two farmers. Because you are a mathematic maniac, **you want the product of <fruit amount> and <money farmer has on hand> to be constant**. You also want to pay as little as possible.
>
> 1. The first farmer has **5** apples and **10$** on hand. So the constant would be **5 * 10 = 50**
>   1. You take 3 apples: 5 - 3 = 2
>   2. To keep the constant: 50 = (5 - 3) * x

   3. **x = 25**
   4. So you have to add to farmer hand **15$** (he had 10$ in the beginning, 10 + 15 = 25) to keep the constant on the initial level.

2. second farmer has **78** apples and **100$** in the hand. So constant would be **78 * 100 = 7800**
   1. You take 3 apples: 78 - 3 = 75
   2. To keep the constant: 7800 = (78 - 3) * x
   3. **x = 104**
   4. So you have to add to farmer hand **4$** to keep constant on the initial level.

So the best deal is to buy apples from the second farmer, because to keep the constant on the initial level, you would only have to pay 4$ for 3 apples, while with the first farmer it was 15$.

But let's stay a little longer on this to clear up price issue. We keep rule of constant alive (price increases as apples amount decreases).

First farmer has 5 apples and 10$ in the hand. If we calculate price of the first apple simply dividing 10$ / 5, then we got **2$**, which would be wrong, since 4 * 12$ = 48 while we need to keep the constant on the same level, which is 50. However, that is actually the way of calculating a ratio between apples and dollars, so that makes it a **market price**.
Anyway, we need to calculate as we did before: (50 / (5 - 1)) - 10$ = **2,5$**, and that is the **execution price** of the first apple, which means that is exactly what you will have to pay.
Out of that we can simply calculate, that **price impact** is 25%, because 2,5$ is 25% higher than 2$.

Let's do the same for the second farmer. **Market price** is a ratio between apples and dollars, so 100$/78 = 1,2821$
But as we calculate **execution price**, we got (7800/ (78-1)) - 100$ = 1,2987$
Price impact in this case is 1,2948%.

1.25 Open-Source

Open-source is a wide topic that we could speak about for days. Doing so probably wouldn't win us any 'life of the party' awards but that's the cost of getting too deep into any topic.

We'll try to keep it simple here.

You've probably heard the term "open-source" a million times, but do you really know what it means?

Open-source can be a source code, design, recipe, etc. that can be further used or improved upon without any legal consequences from the original author. Any user can download, copy, modify, and use it commercially. You literally can do anything with it (but you know, be good!).

The decentralized software-development model, which we are concerned about in the crypto world, is based on the concept of open-source. That means, it is not only a "take-and-use-for-your-own-purposes" kind of thing but also a "collaborative improvement over the product" kind of thing.

The Linux system is the most ubiquitous example of an open-source product that has been improved upon by many. In 2017, there were roughly 15,600 developers working to improve Linux according to this report and the development still continues.

To achieve such **a great community distribution** to a project, a project itself has to be worth the effort and it should bring joy and pride to the developers themselves–all of which Linux does perfectly!

Some important links before we go further:

- Open-source main rules: https://opensource.org/osd
- Open-source Initiative: https://opensource.org/

1.25.1 Want some examples?

Here you go! These are some great examples of open-sourced projects:

- Linux ,
- Blender ,
- Arduino ,
- Open-source colas (out of that recipe, Cube Cola arose),
- Free Beer,
- Creative Commons ,
- OpenEMR ,
- Open Source Drug Discovery ,
- Hyperloop,
- WikiHouse,
- and countless more :).

1.25.2 Is open-source secure?

Free, open, available to everyone, collaborative improvement - has this all triggered your alarm bells? Unfortunately, we do still live in a world where not everyone's intent is noble.

Since open-source has a rule of 'No Discrimination Against Persons or Groups', this means that even bad actors have access to the source code.

The question is **do we have security measures against such nefarious actors?** . . . and the answer is: **to some extent**.

For starters, it's not that easy to put a malicious piece of code into someone's repository. One would have to have access to the repository first and then pass the code review.

Though, someone could attempt to run an attack by adding a malware package to the download page.

Righteous developers likely haven't anticipated every possible security issue and bad actors could find an open gate into a system to do some monkey business.

Methods are clever and numerous, making it impossible to describe them all. That's why users should always aware of potential danger and avoid perilous actions:

- Always download source code from trusted sites and make sure you're downloading the official release
- Check to see that the online repository where the source code is stored has security support (like code-scan bots, autofixes etc.)
- Support projects that have earned your trust and try to find ones where the authors are not anonymous and have a solid community

Also, it is very important, though hard to check, that the developers review the code in a proper way. When it comes to open-source and working with volunteers, there shouldn't be such thing as "just taking a look" into the code.

Code should always be thoroughly checked whether it acts as described or not. Otherwise, a backdoor could be overlooked and that's when you've got a huge problem.

1.25.3 Why do we even need open-source?

It's so much easier to scratch an itchy spot yourself, than trying to explain where it is to someone else and having them assist. Open-source works in a similar way too, where a user who needs something or has an idea to improve something, could simply do it themselves. This has the added benefit of helping others in the community who could benefit from your fix or improvement.

Contribution to a project you use and need confers many benefits such as:

- Better working project
- Greater user satisfaction
- Better product understanding
- Larger commitment to the product, making it last longer
- And a stronger incentive for other users and contributors to use or commit to the product

Even more important, is that improvements are not only to the functionality of a product but also to its security. As discussed before, there are a variety of possible attacks, so users' commits are crucial.

Considering the cryptoworld exclusively, open-source brings these benefits:

- Helps to build decentralization which allows cryptocurrencies to be independent from corporate interest
- Allows for self-checking the code, so you know exactly where your money is going leading to greater trust between participants

And when we put all this together, we get a whole ecosystem with committed users who help build decentralization, correct security issues, use the products themselves, and encourage others to do so as well.

Developers in the cryptoworld who build new projects based on a source code help to "populate it" making it more fit, useful, and friendly to all types of users down to the novice or 'noob'. More can join and find a product that is right for them.

It's happening right now as we write–crypto-based ecosystems are rapidly developing, spreading, and becoming more and more important in the real world every day with the potential of becoming the future of global finance.

1.25.4 Conclusion

Open-source is important in the cryptoworld because it allows crypto-ecosystem to futher develop and to become more secure, recognizable, and useful. The result of open-source software development are highly resilient codes made by users to serve users.

1.26 Bitcoin paper

1.26.1 The paper that first introduced Bitcoin

Bitcoin: A Peer-to-Peer Electronic Cash System

Important: All texts and graphics on this site are property of Bitcoin Project and Satoshi Nakamoto. For original files please visit https://bitcoin.org/en/bitcoin-paper

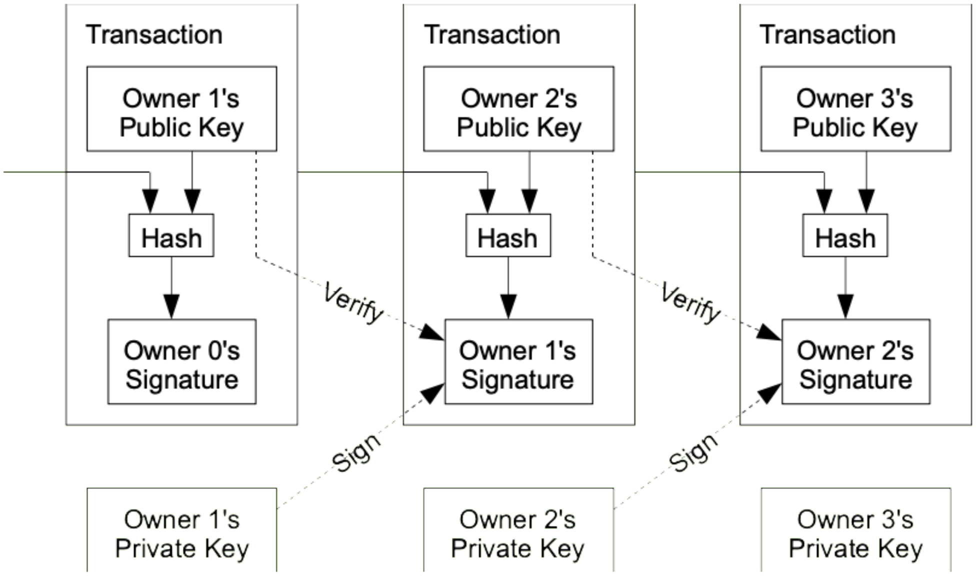
1.26.2 Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non- reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

1.26.3 Transactions

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.

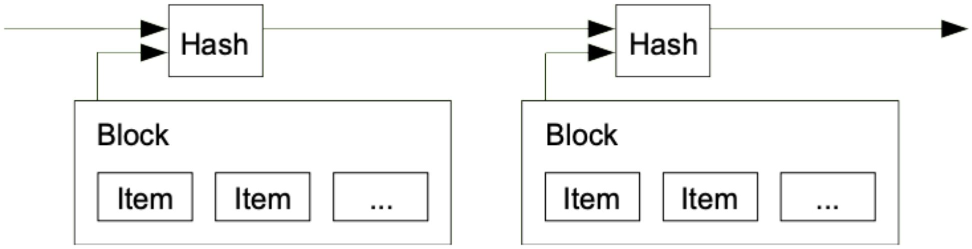


The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank.

We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. In the mint based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

1.26.4 Timestamp Server

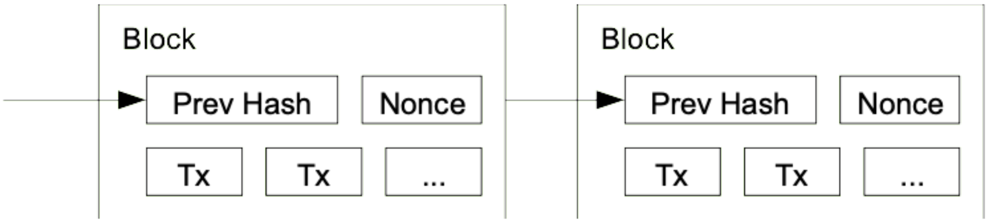
The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.



1.26.5 Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof- of-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.



The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

1.26.6 Network

The steps to run the network are as follows:

- New transactions are broadcast to all nodes.
- Each node collects new transactions into a block.
- Each node works on finding a difficult proof-of-work for its block.
- When a node finds a proof-of-work, it broadcasts the block to all nodes.
- Nodes accept the block only if all transactions in it are valid and not already spent.
- Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof- of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

1.26.7 Incentive

By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended.

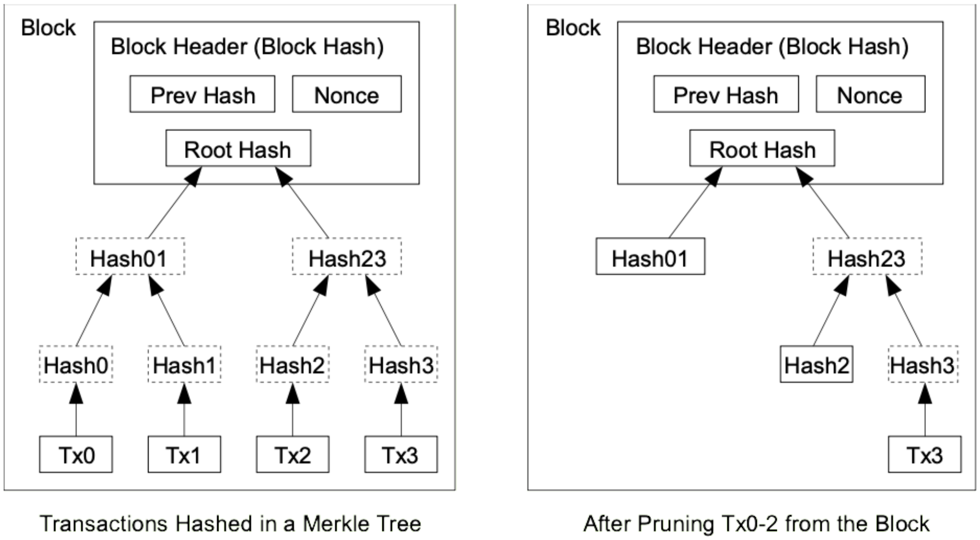
The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

1.26.8 Reclaiming Disk Space

Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree [7][2][5], with only the root included in the block's hash.

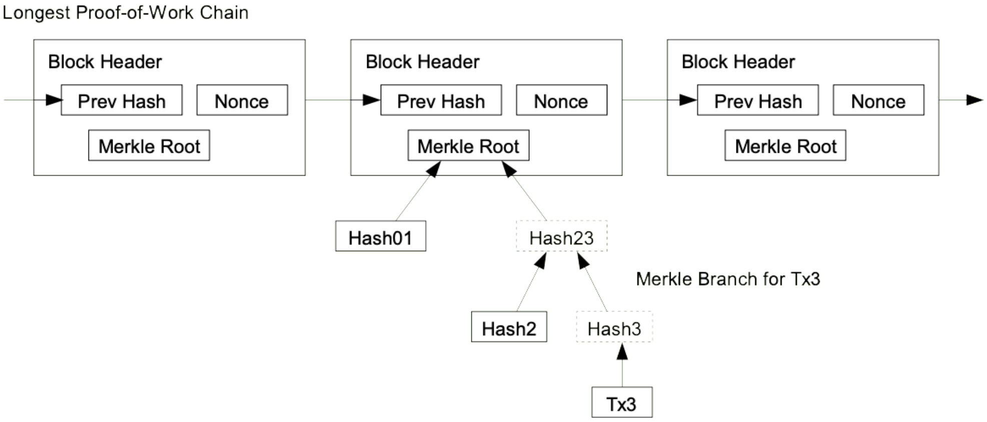
Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.



A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory.

1.26.9 Simplified Payment Verification

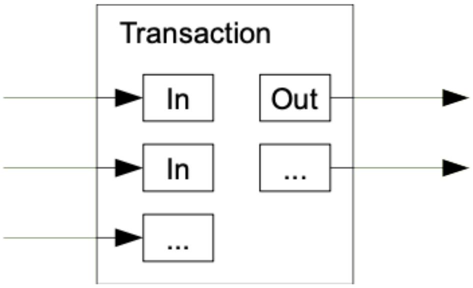
It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.



As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker's fabricated transactions for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency. Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.

1.26.10 Combining and Splitting Value

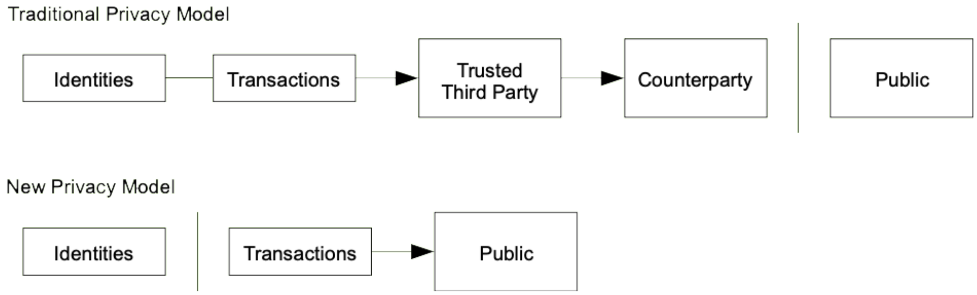
Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender.



It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction's history.

1.26.11 Privacy

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.



As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.

1.26.12 Calculations

We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. Even if this is accomplished, it does not throw the system open to arbitrary changes, such as creating value out of thin air or taking money that never belonged to the attacker. Nodes are not going to accept an invalid transaction as payment, and honest nodes will never accept a block containing them. An attacker can only try to change one of his own transactions to take back money he recently spent.

The race between the honest chain and an attacker chain can be characterized as a Binomial Random Walk. The success event is the honest chain being extended by one block, increasing its lead by +1, and the failure event is the attacker's chain being extended by one block, reducing the gap by -1.

The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem. Suppose a gambler with unlimited credit starts at a deficit and plays potentially an infinite number of trials to try to reach breakeven. We can calculate the probability he ever reaches breakeven, or that an attacker ever catches up with the honest chain, as follows [8]:

p = probability an honest node finds the next block
q = probability the attacker finds the next block
qz = probability the attacker will ever catch up from z blocks behind

$$q_z = \left\{ \begin{array}{cc} 1 & \text{if } p \leq q \\ (q/p)^z & \text{if } p > q \end{array} \right\}$$

Given our assumption that p > q, the probability drops exponentially as the number of blocks the attacker has to catch up with increases. With the odds against him, if he doesn't make a lucky lunge forward early on, his chances become vanishingly small as he falls further behind.

We now consider how long the recipient of a new transaction needs to wait before being sufficiently certain the sender can't change the transaction. We assume the sender is an attacker who wants to make the recipient believe he paid him for a while, then switch it to pay back to himself after some time has passed. The receiver will be alerted when that happens, but the sender hopes it will be too late.

The receiver generates a new key pair and gives the public key to the sender shortly before signing. This prevents the sender from preparing a chain of blocks ahead of time by working on it continuously until he is lucky enough to get far enough ahead, then executing the transaction at that moment. Once the transaction is sent, the dishonest sender starts working in secret on a parallel chain containing an alternate version of his transaction.

The recipient waits until the transaction has been added to a block and z blocks have been linked after it. He doesn't know the exact amount of progress the attacker has made, but assuming the honest blocks took the average expected time per block, the attacker's potential progress will be a Poisson distribution with expected value:

$$\lambda = z\frac{q}{p}$$

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \left\{ \begin{array}{cc} (q/p)^{(z-k)} & if\ k \le z \\ 1 & if\ k > z \end{array} \right\}$$

Rearranging to avoid summing the infinite tail of the distribution…

$$1-\sum_{k=0}^{z}\frac{\lambda^{k}e^{-\lambda}}{k!}\left(1-(q/p)^{(z-k)}\right)$$

Converting to C code…:

```
 #include <math.h>
double AttackerSuccessProbability(double q, int z)
{
    double p = 1.0 - q;
    double lambda = z * (q / p);
    double sum = 1.0;
    int i, k;
    for (k = 0; k <= z; k++)
    {
        double poisson = exp(-lambda);
        for (i = 1; i <= k; i++)
            poisson *= lambda / i;
        sum -= poisson * (1 - pow(q / p, z - k));
     }
 return sum;
 }
```

Running some results, we can see the probability drop off exponentially with z:

```
q=0.1
z=0    P=1.0000000
z=1    P=0.2045873
z=2    P=0.0509779
z=3    P=0.0131722
z=4    P=0.0034552
z=5    P=0.0009137
z=6    P=0.0002428
z=7    P=0.0000647
z=8    P=0.0000173
z=9    P=0.0000046
z=10   P=0.0000012

q=0.3
z=0    P=1.0000000
z=5    P=0.1773523
z=10   P=0.0416605
z=15   P=0.0101008
z=20   P=0.0024804
z=25   P=0.0006132
z=30   P=0.0001522
```

(continues on next page)

(continued from previous page)

```
z=35   P=0.0000379
z=40   P=0.0000095
z=45   P=0.0000024
z=50   P=0.0000006
```

Solving for P less than 0.1%...:

```
P < 0.001
q=0.10   z=5
q=0.15   z=8
q=0.20   z=11
q=0.25   z=15
q=0.30   z=24
q=0.35   z=41
q=0.40   z=89
q=0.45   z=340
```

1.26.13 Conclusion

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

[1] W. Dai, “b-money,” http://www.weidai.com/bmoney.txt, 1998.

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, “Design of a secure timestamping service with minimal trust requirements,” In 20th Symposium on Information Theory in the Benelux, May 1999.

[3] S. Haber, W.S. Stornetta, “How to time-stamp a digital document, “In Journal of Cryptology, vol 3, no 2, pages 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, “Improving the efficiency and reliability of digital time-stamping, “In Sequences II: Methods in Communication, Security and Computer Science, pages 329-334, 1993.

[5] S. Haber, W.S. Stornetta, “Secure names for bit-strings, “In Proceedings of the 4th ACM Conference on Computer and Communications Security, pages 28-35, April 1997.

[6] A. Back, “Hashcash - a denial of service counter-measure, “http://www.hashcash.org/papers/hashcash.pdf, 2002.

[7] R.C. Merkle, “Protocols for public key cryptosystems, “In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122-133, April 1980.

[8] W. Feller, “An introduction to probability theory and its applications,” 1957.

1.27 Comparing Vote Markets on Binance Smart Chain and Ethereum

November 2021

An analysis by @LIzzl Datadeo#8507. For original analytics see `this pdf file`.
Original graphs can be seen here.
Raw data: github.com.

This report analyses the information that was compiled from those votes and compares them with the data of 30 rounds of voting on Ethereum.

The main questions answered are:

- How did users interact with the platform?
- How did voters apply Quadratic Voting?
- What makes a user a successful predictor?
- What is the predictive power of markets.vote?

1.27.1 What is Quadratic Voting?

That's a voting strategy that helps balance the voting power of big players and small ones. The idea standing behind QV mechanism is to make each additional vote cost a lot more than the previous one.
It's worth to remember that in QV participants vote for or against an issue, but also express how strongly they feel about it.
The table down below depicts how cost in tokens depends on the number of votes:

Number of votes	Cost in Tokens
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

More on quadratic voting: click here or here.

1.27.2 How does the markets.vote work?

Users are incentivized to make market predictions in a series of tournaments focussed on a basket of crypto assets.

Quadratic voting is used to generate a consensus in a perceived future market order. Users get a default voting power of 100$V. They then spend a budget of voting power to create a new order, based on their perception of token quality and future potential market performance.

Users are rewarded with a proportional share of a network-generated reward pool depending on the proportionality of their correctness.

Users can amplify their voting power beyond the starting level by demonstrating a history of correct decision-making in the markets, or by purchasing more identities.

More on markets.vote in the whitepaper. Check out also markets.vote app.

1.27.3 Voter Activity

Chain	BSC	ETH
Total voters	151	248
Total votes	7994	840
Total voterIDs	654	318
Average votes per round	265	28
Average user would mint	4 voterIDs	2 voterIDs

For up to date statistics visit Lizzl page on Dune Analytics.

Adoption Curve

ETH

BSC

Voter Turnout

ETH

BSC

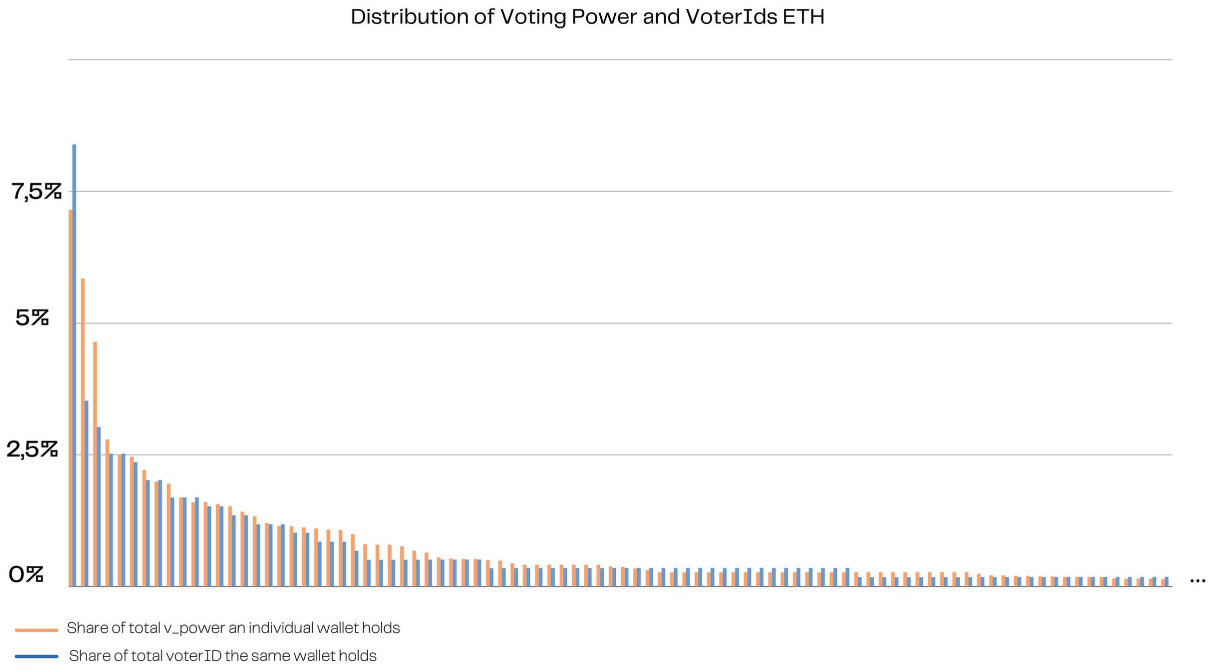
VoterIDs per Wallet

ETH

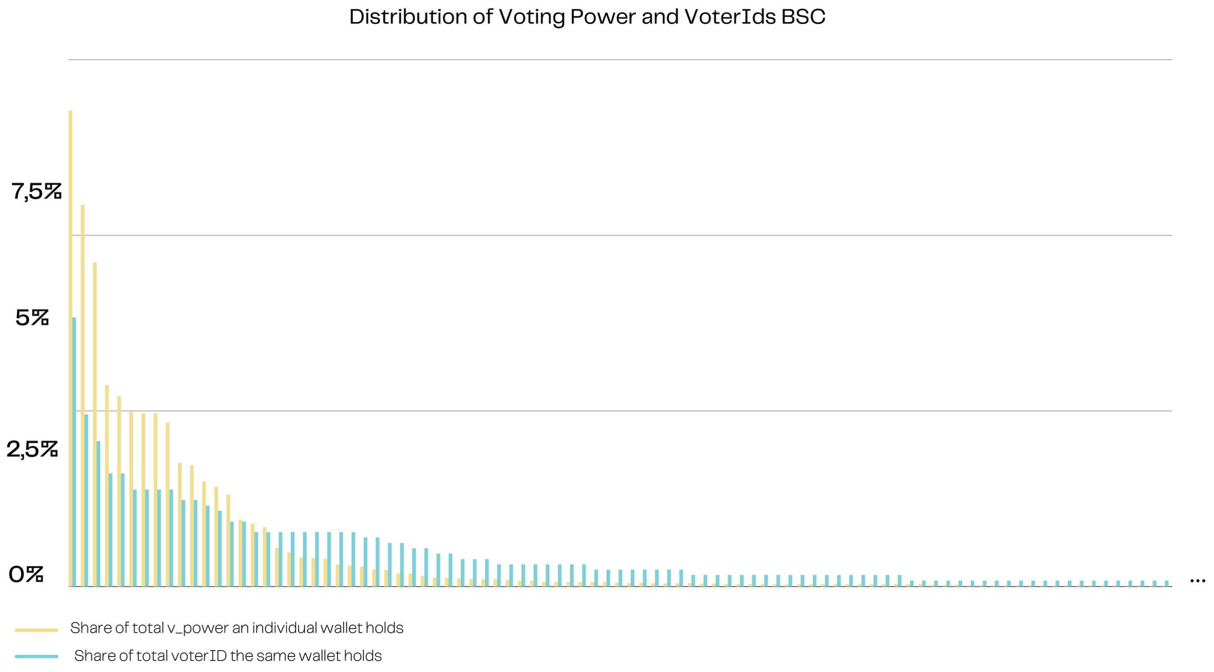
BSC

Distribution of Voting Power and VoterIDs

ETH

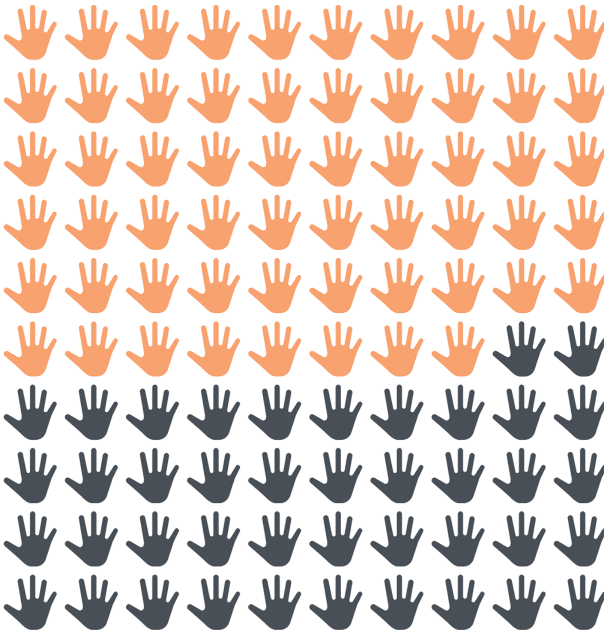

Distribution of Voting Power and VoterIds ETH
7,5%
5%
2,5%
0%
...
Share of total v_power an individual wallet holds
Share of total voterID the same wallet holds

BSC



1.27.4 How did voters apply QV?

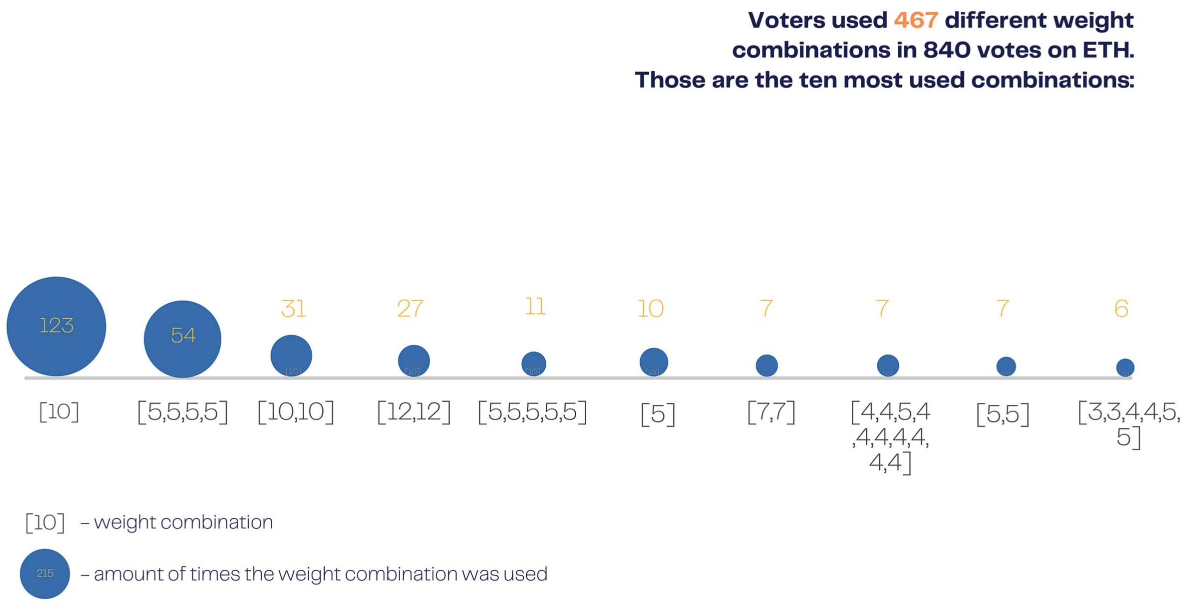
ETH



On average, voters used **58%** of their voting power per vote on ETH.

100, 200 & 99 were the three most used amounts of voting power.

On average, voters made **2** choices per vote on ETH: With **10, 1, 2** being the most used coin choices.

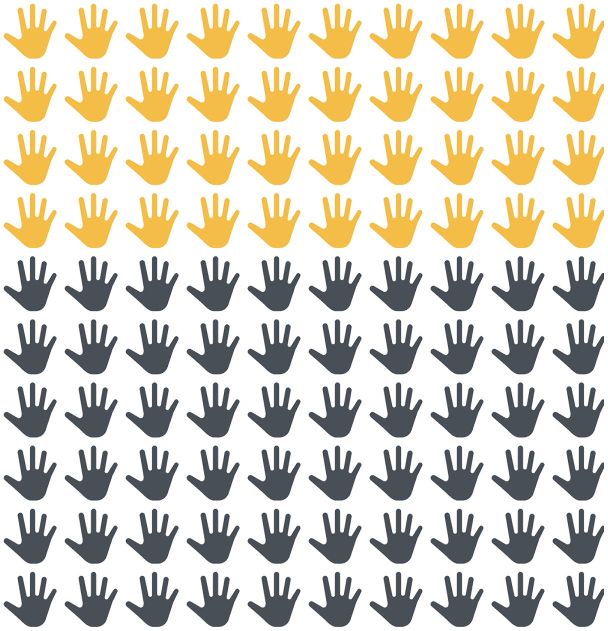


Voters used 467 different weight combinations in 840 votes on ETH. Those are the ten most used combinations:

[10] – weight combination

● – amount of times the weight combination was used

BSC

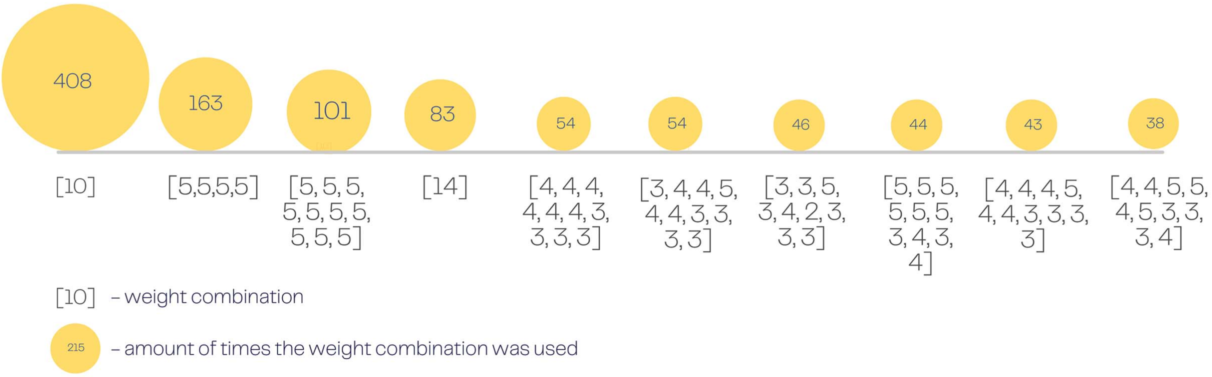


On average, voters used **40%** of their voting power per vote on BSC.

100, 99 & 97 were the three most used amounts of voting power.

On average, voters made **10** choices per vote on BSC, choosing all ten coins in one vote.

Voters used 2184 different weight combinations in 7994 votes on BSC. Those are the ten most used combinations:



1.27.5 Coin clairvoyants

Top 3 wallets with most Voting Power on BSC chain			
Address	0x34…39EA	0xE3…D58f	0x8E…734b
Voting Power	45 515	36 498	30 994
# of voterID	50	32	27
# of rounds participated	30	28	26
% of voting power used	36.9%	33.3%	29.9%
AVG # of choices	9.6	9.6	9.8
3 most weight combination used	[3,3,4,1,4,4,2,2,3,4] [7,7,7,8,8,6,5,6,7] [9,9,9,9,9,9,3,9,9]	[3,3,3,2,4,3,3,3,3,4] [4,3,3,2,3,3,3,3,4,3] [4,4,5,5,4,4,3,6,5,4]	[5,5,5,5,5,5,3,4,3,4] [10,8,10,7,10,10,8,8,8,8] [4,4,4,5,4,4,3,3,3,3]

Top 3 wallets with most Voting Power on ETH chain			
Address	0x34…39EA	0x6e…dDec	0x67…e640
Voting Power	5400	4406	3501
# of voterID	50	5	15
# of rounds participated	3	26	16
% of voting power used	95.9%	45.9%	61.3%
AVG # of choices	4.1	3.7	9.9
3 most weight combination used	[5,5,5,5] [5,5,5,5,5]	[10,10] [12,12] [10]	[4,4,5,4,4,4,4,4,4,4] [4,4,4,4,4,4,4,4,4,4] [3,3,4,3,3,3,3,3,3,3]

Top 3 Voting Power Earners on BSC			
Address	0x34…39EA	0xE3…D58f	0x8E…734b
Bought Voting Power	5000	3200	2700
Earned Voting Power	40 515	33 298	28 294
% Increase	8.1%	10.4%	10.4%

On BSC the Top 3 Earners are the same wallets with most Voting Power			
# of rounds participated	30	28	26
% of voting power used	36.9%	33.3%	29.9%
AVG # of choices	9.6	9.6	9.8
3 most weight combination used	[3,3,4,1,4,4,2,2,3,4]	[3,3,3,2,4,3,3,3,3,4]	[5,5,5,5,5,5,3,4,3,4]
	[7,7,7,8,8,6,5,6,7]	[4,3,3,2,3,3,3,3,4,3]	[10,8,10,7,10,10,8,8,8,8]
	[9,9,9,9,9,9,3,9,9]	[4,4,5,5,4,4,3,6,5,4]	[4,4,4,5,4,4,3,3,3,3]

Top 3 Voting Power Earners on ETH			
Address	0x6e…dDec	0xDc…CA7D	0xB3…6B7C
Bought Voting Power	500	200	100
Earned Voting Power	3906	1074	469
% Increase	781.2%	537%	375%

How did the most successful voters used the vote markets?			
# of rounds participated	26	25	20
% of voting power used	45.9%	47.5%	45.3%
AVG # of choices	3.7	5.8	9.7
3 most weight combination used	[10,10]	[5,5,5,5]	[EACH VOTE A DIFFERENT COMBINATION]
	[12,12]	[+48 DIFFERENT COMBINATIONS]	
	[10]		

Top 3 VoterIDs BSC			
VoterID	54	61	121
Voting Power	3475	2903	2628

How did the most successful VoterIDs used the vote markets?			
# of rounds participated	25	22	27
% of voting power used	21.5%	20.6%	26.9%
AVG # of choices	4.3	4	9.3
3 most weight combination used	[EACH VOTE A DIFFERENT COMBINATION]	[EACH VOTE A DIFFERENT COMBINATION]	[EACH VOTE A DIFFERENT COMBINATION]

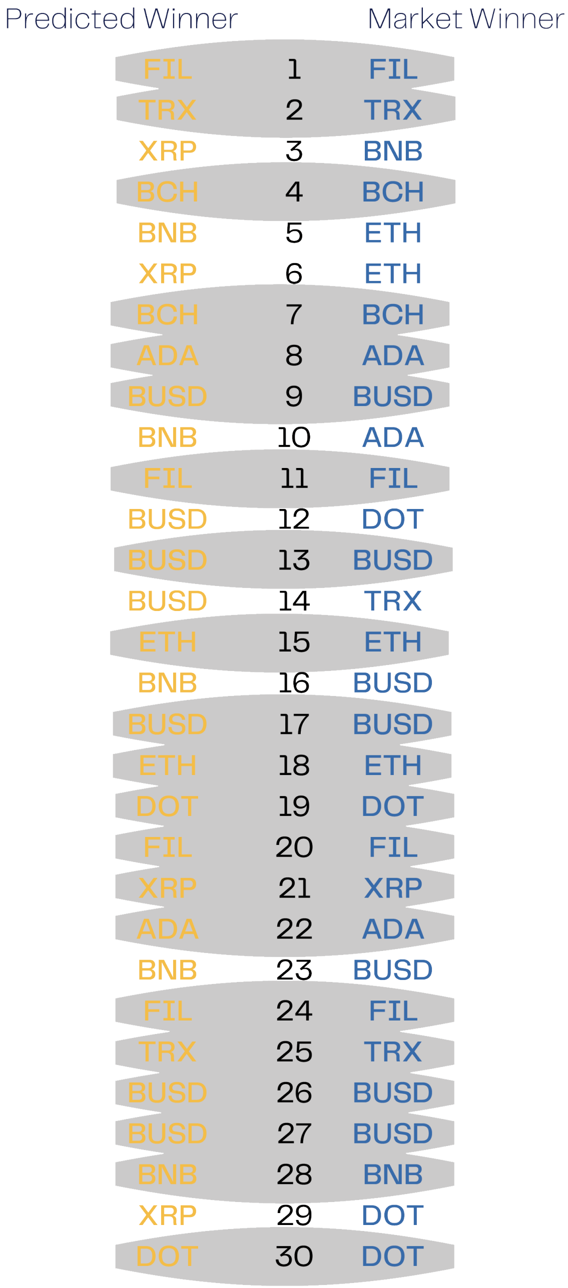
Top 3 VoterIDs ETH			
VoterID	195	1	197
Voting Power	2021	1226	1056

How did the most successful VoterIDs used the vote markets?			
# of rounds participated	23	27	23
% of voting power used	39%	28.4%	42.7%
AVG # of choices	3.6	7.6	3.6
3 most weight combination used	[EACH VOTE A DIFFERENT COMBINATION]	[EACH VOTE A DIFFERENT COMBINATION]	[EACH VOTE A DIFFERENT COMBINATION]

1.27.6 Market predictions

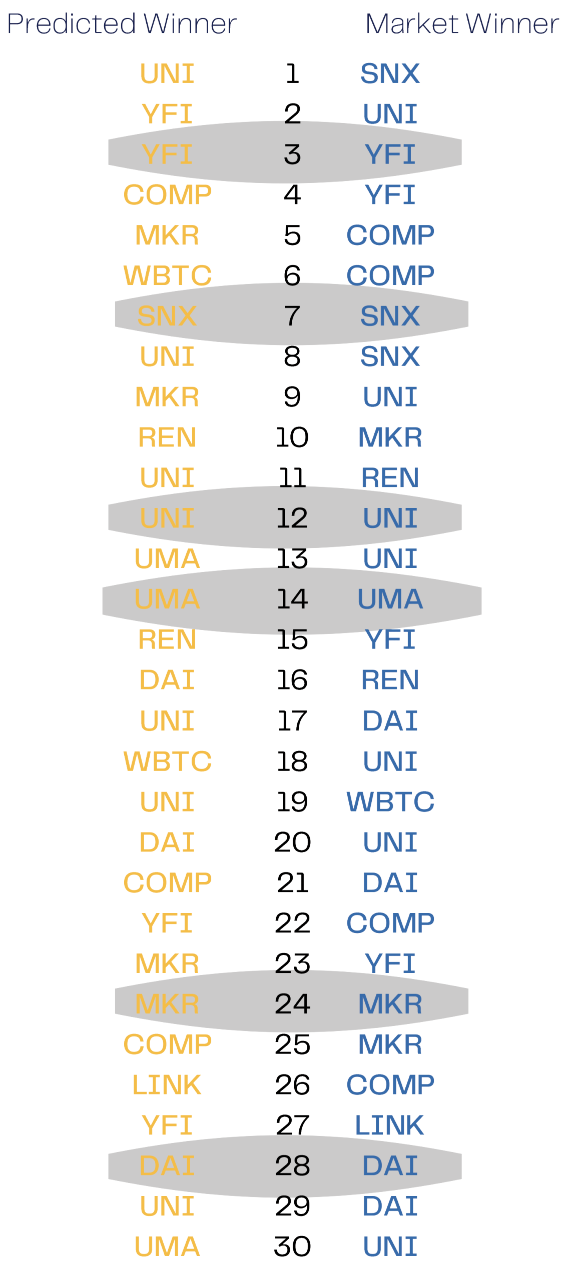
21 out of **30** coins were predicted successfully on **BSC**.
Successful predictions were made in all rounds but **3,5,6,10,12,14,16,23,29**.
BUSD was the token that was predicted most successfully.

6 out of **30** coins were predicted successfully on **ETH**.
Successful predictions were made in rounds **3,7,12,14,24,28**.
There was no particular coin that was predicted more successfully than others.



Predicted Winner		Market Winner
FIL	1	FIL
TRX	2	TRX
XRP	3	BNB
BCH	4	BCH
BNB	5	ETH
XRP	6	ETH
BCH	7	BCH
ADA	8	ADA
BUSD	9	BUSD
BNB	10	ADA
FIL	11	FIL
BUSD	12	DOT
BUSD	13	BUSD
BUSD	14	TRX
ETH	15	ETH
BNB	16	BUSD
BUSD	17	BUSD
ETH	18	ETH
DOT	19	DOT
FIL	20	FIL
XRP	21	XRP
ADA	22	ADA
BNB	23	BUSD
FIL	24	FIL
TRX	25	TRX
BUSD	26	BUSD
BUSD	27	BUSD
BNB	28	BNB
XRP	29	DOT
DOT	30	DOT

BSC



Predicted Winner		Market Winner
UNI	1	SNX
YFI	2	UNI
YFI	3	YFI
COMP	4	YFI
MKR	5	COMP
WBTC	6	COMP
SNX	7	SNX
UNI	8	SNX
MKR	9	UNI
REN	10	MKR
UNI	11	REN
UNI	12	UNI
UMA	13	UNI
UMA	14	UMA
REN	15	YFI
DAI	16	REN
UNI	17	DAI
WBTC	18	UNI
UNI	19	WBTC
DAI	20	UNI
COMP	21	DAI
YFI	22	COMP
MKR	23	YFI
MKR	24	MKR
COMP	25	MKR
LINK	26	COMP
YFI	27	LINK
DAI	28	DAI
UNI	29	DAI
UMA	30	UNI

ETH

1.28 A report on the Vote Markets

September 2021

An analysis by @LIzzl Datadeo#8507. For original analytics see `this pdf file`.

Users have voted over 37 rounds to predict the market performance of Defi tokens using a new voting mechanism called Quadratic Voting.

This report analyses the information that was compiled from those votes. The main questions answered are:

- How did users interact with the platform?
- How did voters apply Quadratic Voting?
- What makes a user a successful predictor?
- What is the predictive power of markets.vote?

1.28.1 What is Quadratic Voting?

That's a voting strategy that helps balance the voting power of big players and small ones. The idea standing behind QV mechanism is to make each additional vote cost a lot more than the previous one.
It's worth to remember that in QV participants vote for or against an issue, but also express how strongly they feel about it.
The table down below depicts how cost in tokens depends on the number of votes:

Number of votes	Cost in Tokens
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

More on quadratic voting: click here or here.

1.28.2 How does the markets.vote work?

Users are incentivized to make market predictions in a series of tournaments focussed on a basket of crypto assets.

Quadratic voting is used to generate a consensus in a perceived future market order. Users get a default voting power of 100$V. They then spend a budget of voting power to create a new order, based on their perception of token quality and future potential market performance.

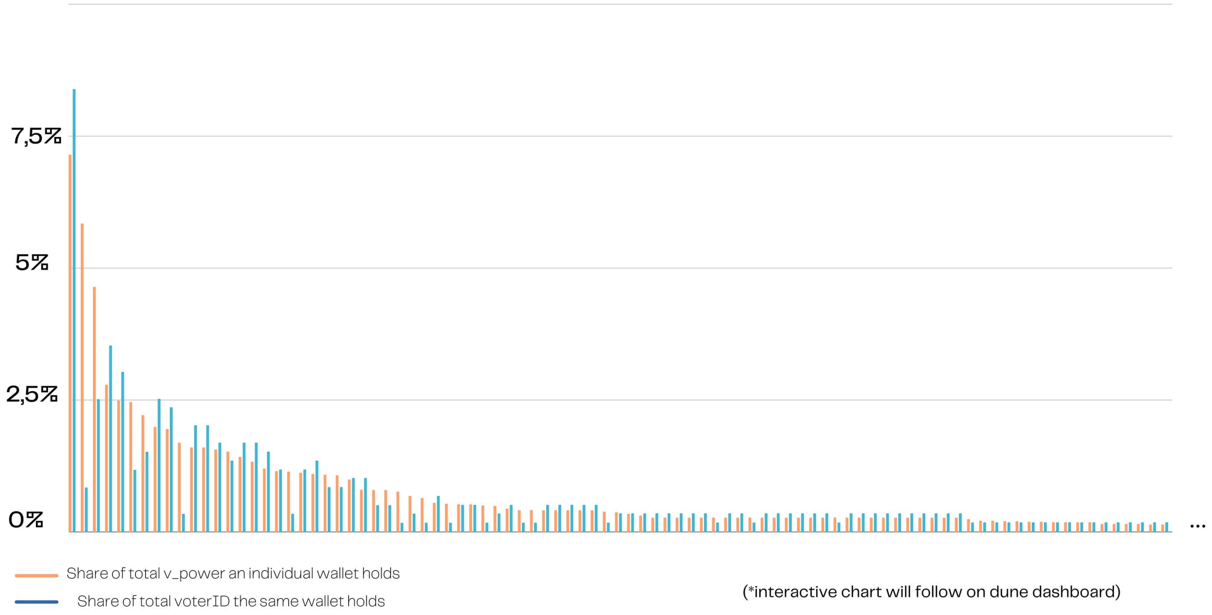
Users are rewarded with a proportional share of a network-generated reward pool depending on the proportionality of their correctness.

Users can amplify their voting power beyond the starting level by demonstrating a history of correct decision-making in the markets, or by purchasing more identities.

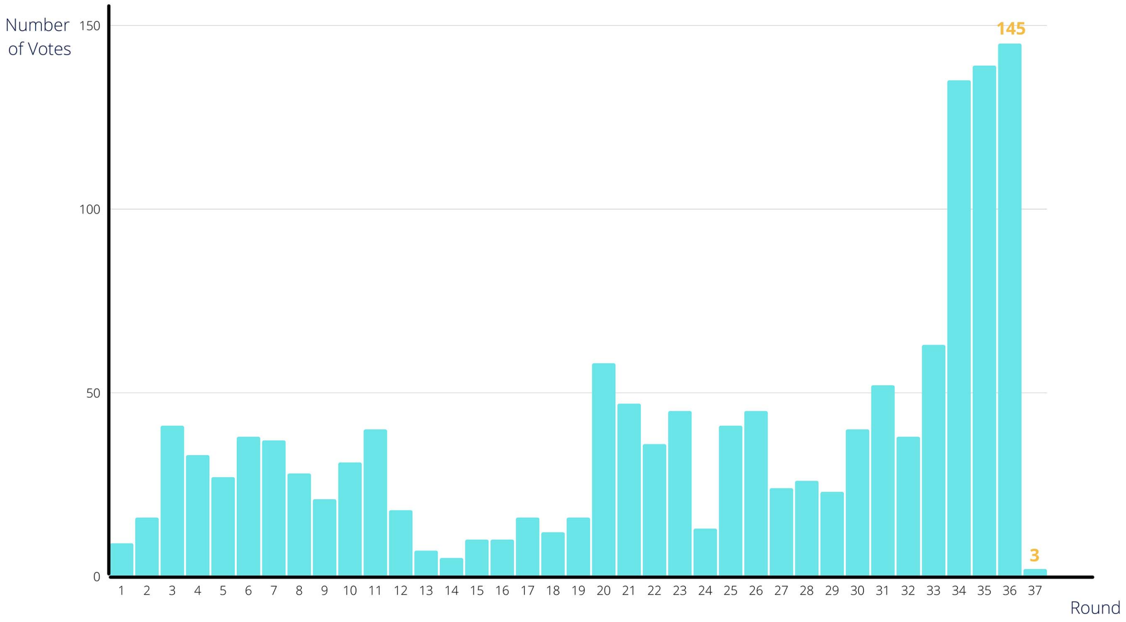
More on markets.vote in the whitepaper. Check out also markets.vote app.

1.28.3 Voter Activity

How did users interact with vote markets?	
Total voters	298
Total votes	1379
Total voterIDs	597
Average votes per round	38
Average user would mint	2 voterIDs
No voterID has ever been transfered	

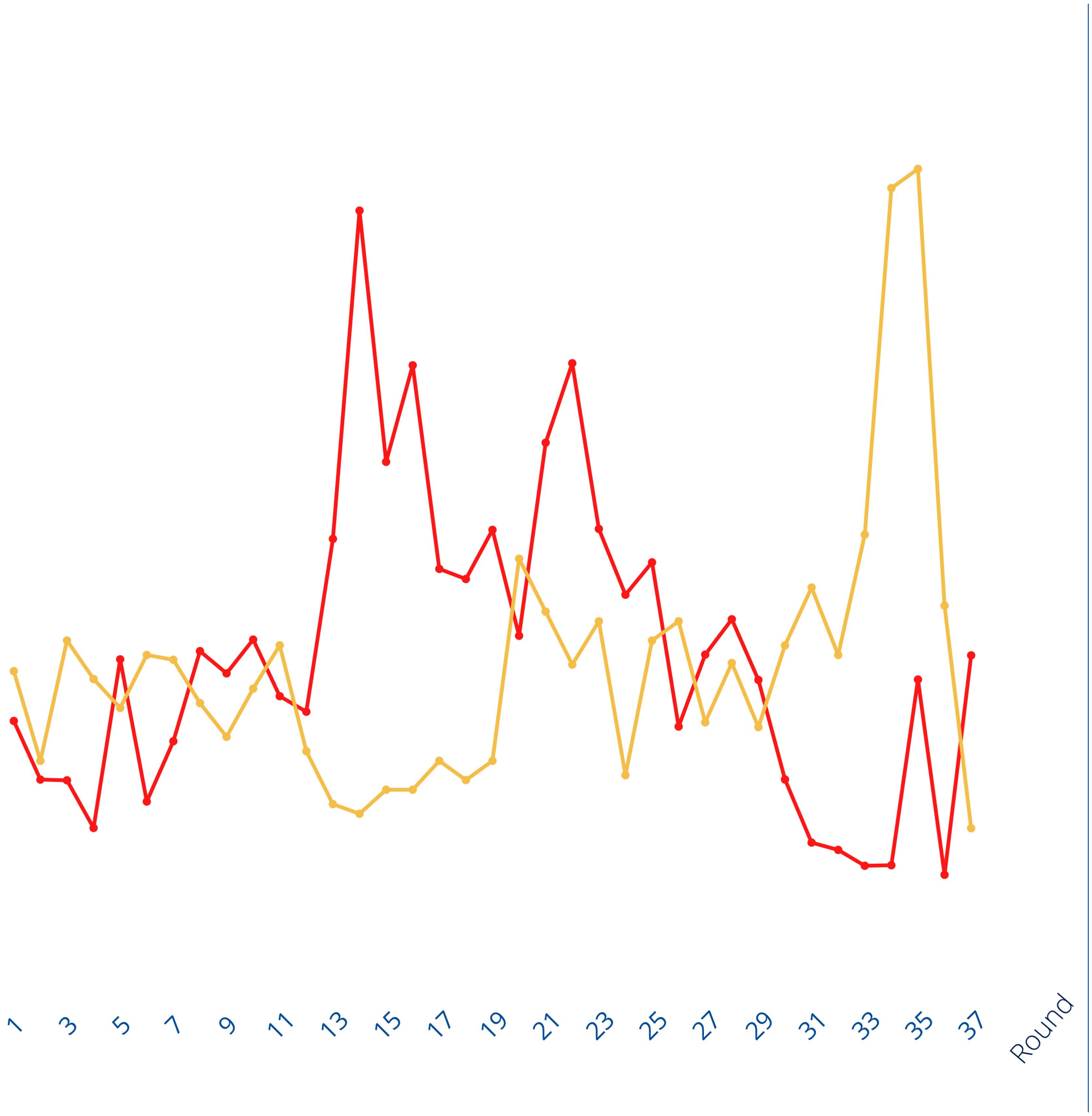


(*interactive chart will follow on dune dashboard)



(*interactive chart will follow on dune dashboard)

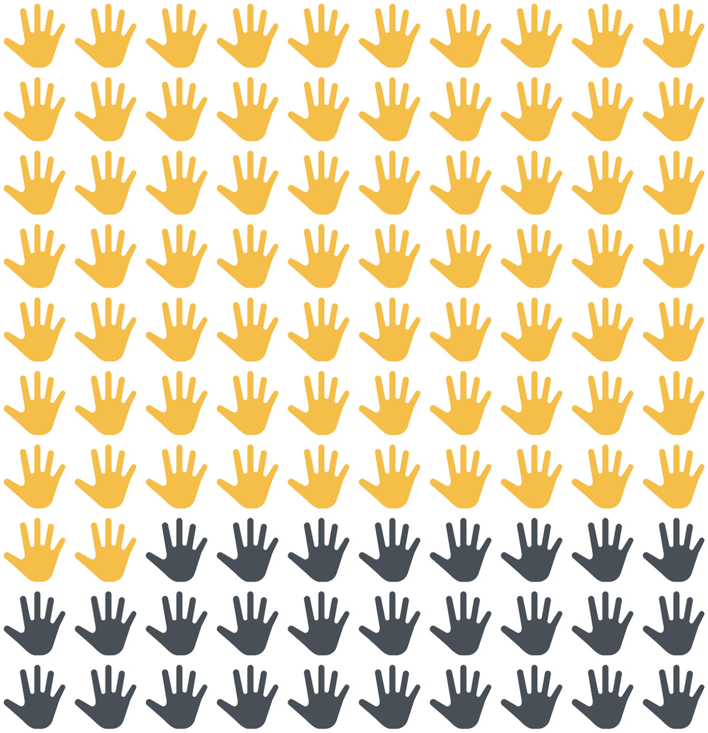
1.28.4 Voter turnout



Voter turnout with **transaction cost***

- *(average amount of ETH spent per transaction)*

1.28.5 How did voter apply QV?



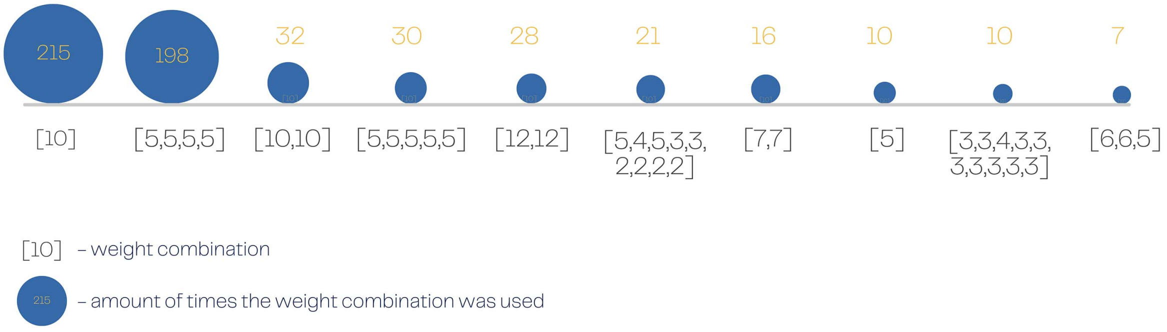
On average, voters used **72%** of their voting power per vote.

100, 125, 97 were three most used amounts of voting power.

On average, voters made **5** choices per vote.

With **10, 4, 1** being the most used number of choices.

Voters used 668 different weight combinations in 1387 votes
Those are the ten most used combinations:



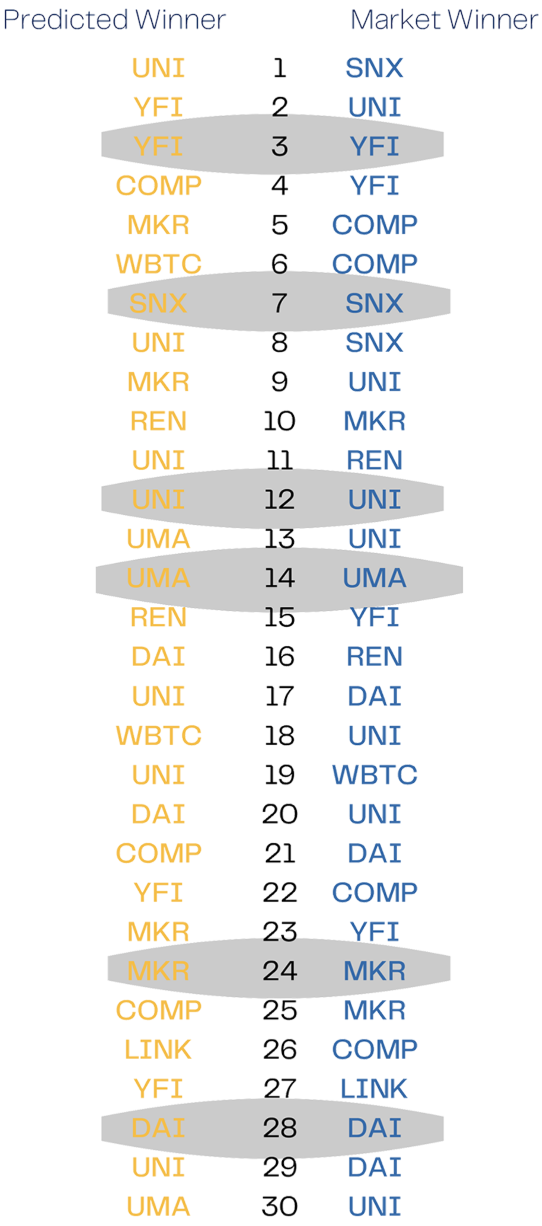
1.28.6 Coin clairvoyants

Top 3 wallets with most Voting Power			
Address	0x34…39EA	0x6e…dDec	0x67…e640
Voting Power	5400	4406	3501
# of voterID	50	5	15
# of rounds participated	3	26	16
% of voting power used	95.9%	45.9%	61.3%
AVG # of choices	4.1	3.7	9.9
3 most weight combination used	[5,5,5,5]	[10,10]	[4,4,5,4,4,4,4,4,4,4]
	[5,5,5,5,5]	[12,12]	[4,4,4,4,4,4,4,4,4,4]
		[10]	[3,3,4,3,3,3,3,3,3,3]

Top 3 Voting Power Earners			
Address	0x6e…dDec	0xDc…CA7D	0xB3…6B7C
Bought Voting Power	500	200	100
Earned Voting Power	3906	1074	469
% Increase	781.2%	537%	375%

How did the most successful voters used the vote markets?			
# of rounds participated	26	25	20
% of voting power used	45.9%	47.5%	45.3%
AVG # of choices	3.7	5.8	9.7
3 most weight combination used	[10,10]	[5,5,5,5]	[EACH VOTE A DIFFERENT COMBINATION]
	[12,12]	[+48 DIFFERENT COMBINATIONS]	
	[10]		

1.28.7 Market predictions



Predicted Winner		Market Winner
UNI	1	SNX
YFI	2	UNI
YFI	3	YFI
COMP	4	YFI
MKR	5	COMP
WBTC	6	COMP
SNX	7	SNX
UNI	8	SNX
MKR	9	UNI
REN	10	MKR
UNI	11	REN
UNI	12	UNI
UMA	13	UNI
UMA	14	UMA
REN	15	YFI
DAI	16	REN
UNI	17	DAI
WBTC	18	UNI
UNI	19	WBTC
DAI	20	UNI
COMP	21	DAI
YFI	22	COMP
MKR	23	YFI
MKR	24	MKR
COMP	25	MKR
LINK	26	COMP
YFI	27	LINK
DAI	28	DAI
UNI	29	DAI
UMA	30	UNI

6 out of **30** coins were predicted successfully. Successful predictions were made in rounds **3,7,12,14,24,28**.

There was no particular coin that was predicted more successfully than others.

1.29 Finding Signal in markets.vote

An analysis of the first year of markets.vote by @ShiftRunStop#8542 >source<

---

markets.vote has been running since *November 2020* on Ethereum and since *April 2021* on Binance Smart Chain.

On each chain, finance.vote ID holders use quadratic voting to choose which token(s) they think will perform best over the coming week. At the end of the market period, 100,000 FVT is distributed amongst all those who chose the winning token, proportional to the number of votes they assigned the winner.

Winning voters will also receive more voting power for forthcoming vote markets. The list of coins to vote on is different on each chain.

On Ethereum the list is dominated by the big players in DeFi as well as the OG cryptocurrency, Bitcoin, and a USD stablecoin:

- Chainlink (LINK)
- Wrapped Bitcoin (WBTC)
- Dai (DAI)
- Uniswap (UNI)
- Maker (MKR)
- Compound (COMP)
- UMA (UMA)
- yearn.finance (YFI)
- Synthetix Network Token (SNX)
- Ren (REN)

The list of coins on Binance Smart Chain is a more general mix of high cap cryptocurrencies and a USD stablecoin:

- Cardano (ADA)
- Bitcoin Cash (BCH)
- Binance Coin (BNB)
- Binance USD (BUSD)
- Polkadot (DOT)
- Ethereum (ETH)

- Filecoin (FIL)
- Litecoin (LTC)
- Tron (TRX)
- Ripple (XRP)

1.29.1 Predicting the Market

The cryptocurrency market is notoriously volatile and unpredictable. It is clearly not possible to perfectly predict the future. But with the combined consideration, market analysis and intuition of hundreds of crypto degens will it be possible to predict the market more profitably than relying on your own strategy?

Beyond the fact that the crypto market is fundamentally difficult to predict, a number of other factors came into play that may have reduced the quality of prediction signal coming out of markets.vote.

Factors with a negative impact on predict market:

- **Mass hedged voting**

  Despite attempts at sybil resistance to stop users minting many IDs and voting multiple times, there is clear evidence that this happened. It happened most on Binance Smart Chain where the parameters affecting the base price and decay rate of minting IDs on that chain failed to provide sufficient sybil resistance. This, coupled with the low cost of voting on BSC meant that many users heavily hedged their votes across all tokens. As there was no penalty for guessing incorrectly, there was nothing to lose from voting many times in order to win a larger share of the FVT award each week. This phenomenon also occurs on Ethereum but to a lesser extent. On the Ethereum chain, the sybil resistance design worked more effectively initially due to a decay rate which kept the price of multiple IDs in quick succession unaffordable. Also, the art attached to Ethereum IDs for a period in early 2021 increased the desirability of the NFTs and kept the price to a level that made it unviable economically to mint multiple IDs purely with the intention of earning rewards on markets.vote. Finally the high price of gas on the Ethereum chain meant that voting en masse was at times excruciatingly expensive and meant it did not make financial sense to vote many times as the rewards may not have covered the gas fees.

- **People look back not forward**

  The data shows that voters continually voted highly for tokens that were doing well that week. Presumably with the confidence that they would continue to do well the following week. While this was sometimes true, history proves that it was not usually the case - tokens in fact tended to have short term spikes in price and were often corrected after a short term rise.

- **Lots of casual voting**

  It’s probably fair to say that the majority of people who voted did so having not done a great deal of research or consideration into what tokens may actually perform well the following week. Besides gas fees (which were often considered on the Ethereum network) there was nothing at stake to lose for incorrect predictions. This resulted in lots of votes being placed with little to no consideration for how well that token may actually perform over the coming week.

1.29.2 Potential Strategies

A number of investment strategies that could have been adopted based on the week's prediction have been chosen for the purpose of this analysis. All of the historic voting data has been extracted from the contract as well as all of the token prices at the start and end of each round's market period. Using this data it has been possible to play out each strategy to see how much money would have been made (or lost) by adopting each.

The strategies chosen for this exercise were as follows. Each strategy begins with $1000.

1. **Baseline**

   Invest $100 into each token after voting for round 1 closes. HODL.

2. **Prediction weighted**

   For those that have absolute trust in their fellow voters. After the voting for each round closes distribute your accrued capital proportionally across all of the tokens according to the number of votes they received in the prediction market.

3. **All in**

   After the voting for each round closes, the entire accrued capital is moved into the token that received the highest number of votes.

4. **Top four**

   After the voting for each round closes, the accrued capital is divided evenly amongst the four tokens that have received the highest number of votes.

5. **All in pessimist**

   For those that believe that the group will consistently fail to pick out the token that will perform best the following week. After the voting for each round closes, the entire accrued capital is moved into the token that received the lowest number of votes.

6. **Bottom four**

   As strategy 5, but after the voting for each round closes, the accrued capital is divided evenly amongst the four tokens that have received the lowest number of votes.

7. **Top; bottom**

   For those that think the group is likely to choose the best token for coming week or the worst, and not much inbetween. After the voting for each round closes, the accrued capital is divided evenly between the token that received the highest number of votes and the token that received the lowest number of votes.

8. **Top 2; Bottom 2**

   As strategy 8 but a bit more hedgy. After the voting for each round closes, the accrued capital is divided evenly amongst the two tokens that have received the highest number of votes as well as the two tokens that have received the lowest number of votes: 25% to each token.

9. **High cap maximalist**

   Invest the full $1000 into the highest market cap coin in the market and hodl (BTC on Ethereum markets.vote; ETH on BSC markets.vote)

10. **BONUS STRATEGY: Crypto Hamster**

    Inspired by this news story from September 2021, this strategy simulates what would have happened if you'd allowed your pet hamster to choose which of the 10 tokens you moved all of your capital into at the start of each round. You can respawn the hamster with a new set of choices by clicking the button below each graph in the analysis for each chain below.

Note: Where applicable, any trading fees that would need to be paid each week when exchanging from one token to another have not been taken into account in this analysis. On the occasions where multiple coins received the same number of votes the capital was evenly divided between them both.

1.29.3 BSC markets.vote data

It's fairly clear from the similarity of the lines for all strategies on the chart above that **no clear signal can be drawn from the historic markets.vote data on the Binance Smart Chain.** This is almost certainly because mass hedged voting by a relatively small number of voters using multiple IDs has introduced too much noise to the data.

It is notable, however, that despite the apparent lack of a clear signal in this data the strategy that would have earned the investor **the most money would have been to go all-in each week on the token** that received the most votes from the markets.vote users.

On the original site there is a button that allows to respawn the BSC Crypto Hamster numerous times. Doing that confirms that following no single strategy based on BSC markets.vote data would have produced a markedly different result than getting your pet hamster to do your investing for you.

1.29.4 ETH markets.vote data

In contrast to the data produced by markets.vote on BSC, on Ethereum there are distinct differences in outcome between the different strategies that could have been employed. One remarkable thing is that markets.vote voters consistently failed to choose the token that would perform best the following week.

Furthermore, they often chose the token that would perform the worst. This is reflected by the fact that after 52 rounds of voting, if after each week an investor had moved their entire capital to the token predicted to perform best they would end up with **only $852 from their initial $1000**.

Even more remarkable is that the strategy that performed best over 52 weeks was for an investor to move their entire capital each week to **the token that had received the fewest votes**. If that strategy was consistently adopted the initial investment of $1000 would have become $10,369 after 52 weeks. This is a 12× better return than investing in the most voted for token each week.

Mashing the "Respawn ETH Crypto Hamster" button quickly highlights that only very rarely would a near brain-dead rodent have made more money than if you had consistently invested in the very tokens that the markets.vote users told you not to. Conversely, only very rarely would little Hammy have done more damage to your initial $1000 investment than if you had consistently moved your investment to the token with the most votes.

Just so you know what you could have won, **if the markets.vote users had perfectly predicted the market each week** and you had consistently moved your entire capital each week to follow their predictions, **your initial investment of $1000 would have become $267.5 million after a year**.

1.29.5 References

- Binance Smart Chain markets.vote historic voting data extracted from the markets.vote contract using bscscan.
- Ethereum markets.vote historic voting data was extracted from the markets.vote contract using etherscan.
- Historic token prices for the start and end of each market period for both blockchains were obtained from cryptowat.ch and their cryptofinance plugin for Google Sheets.
- If you wish to analyse this data fully and perhaps come up with some of your own strategies to play out, here is the full data on Google Sheets.
- Interactive charts powered by ApexCharts.

CHAPTER TWO

INDICES AND TABLES

- genindex
- modindex
- search